Data-enabled physics-informed machine learning for reduced-order modeling digital twin: application to nuclear reactor physics

Helin Gong, Sibo Cheng, Zhang Chen, Qing Li

To cite this version:

Helin Gong, Sibo Cheng, Zhang Chen, Qing Li. Data-enabled physics-informed machine learning for reduced-order modeling digital twin: application to nuclear reactor physics. Nuclear Science and Engineering, In press, 10.1080/00295639.2021.2014752 . hal-03598305

HAL Id: hal-03598305
https://hal.science/hal-03598305v1

Data-enabled physics-informed machine learning for reduced-order modeling digital twin: application to nuclear reactor physics

Helin Gong [1,†], Sibo Cheng [2,3,†], Zhang Chen[1], and Qing Li *[1]

[1]Science and Technology on Reactor System Design Technology Laboratory, Nuclear Power Institute of China, 610041, Chengdu, China.
[2]Data Science Institute, Imperial College London, SW72BU, UK.[†]
[3]EDF R&D Saclay, 91120 Palaiseau, France
[†]Equal contribution to this work

Abstract

This paper proposes an approach that combines reduced-order models with machine learning in order to create physics-informed digital twins to predict high-dimensional output quantities of interest, such as neutron flux and power distributions in nuclear reactor cores. The digital twin is designed to solve forward problems given input parameters, as well as to solve inverse problems given some extra measurements. Offline, we use reduced-order modeling, namely, the proper orthogonal decomposition (POD) to assemble physics-based computational models that are accurate enough for fast predictive digital twin. The machine learning techniques, namely, k-nearest-neighbors (KNN) and decision trees (DT) are used to formulate the input-parameter-dependent coefficients of the reduced basis, whereafter the high-fidelity fields are able to be reconstructed. Online, we use the real time input parameters to rapidly reconstruct the neutron field in the core based on the adapted physics-based digital twin. The effectiveness of the framework is illustrated through a real engineering problem in nuclear reactor physics - reactor core simulation in the life cycle of HPR1000 governed by the two-group neutron diffusion equations affected by input parameters, i.e., burnup, control rod inserting step, power level and temperature of the coolant, which shows potential applications for on-line monitoring purpose.
Keywords. Digital twin; Model order reduction; Machine learning; Proper orthogonal decomposition; Nuclear reactor physics

1 Introduction

A nuclear power plant is a complex physical system to produce electricity. The production of nuclear energy is done under very high safety standards where tight criteria must be satisfied both at design and operation levels. What is essentially required is the accurate knowledge of significant quantities like temperature, neutron flux, power, irradiation etc. The quantities can be global outputs like the maximum or average temperature or the total generated power but the knowledge of more detailed information like temperature, flux and/or power maps in the whole reactor may also be required.

This knowledge can be achieved through measurements, and also through simulation / mathematical / numerical models. Various numerical models have been developed, and give a very good

---

[*] Corresponding author: liqing_npic@163.com

[†] S. Cheng was at EDF R&D, France when this work is performed. He is now at the Data Science institute of Imperial College London

estimation of the fine neutronic behavior of the core [1, 2]. Moreover, measurements, obtained in operational context, allow to improve the quality of the knowledge of the whole core state. These measures could also be used to correct the model bias that are difficult to be taken into account in the real process. The *digital twin* paradigm ensembles *simulation + measurements* are considered together in order to get the best estimation of this complex system state. The digital twin paradigm has received much attention in diverse industrial applications, such as aircraft sustainment procedures and structural health monitoring [3], simulation-based vehicle certification and fleet management [4], mitigating unpredictable, undesirable emergent behavior in complex systems [5, 6].

As one of the key components of the digital twin, the simulation models involved in nuclear engineering often consist of parameter-dependent Partial Differential Equations (PDEs) that are usually expensive to solve [7, 8]. The Reduced basis (RB) methods [9, 10], manage to offer highly advantageous reduction of computational effort without significant loss of precision, thus provide new opportunities for simulations in nuclear engineering domain requiring i) many-queries e.g., optimal control, inverse problems, uncertainty quantification etc., ii) real-time evaluations e.g., online monitoring, parameter estimation, etc.

RB methods are generally functioning in an offline-online paradigm. In the *offline* stage, a set of reduced basis, which represents the principal information of the underlying physical problem, is extracted from a collection of conventional numerical solutions (also called snapshots) for different inputs, usually with a high-fidelity numerical solver (also called full models). The reduced basis can be obtained through a broad range of model reduction processes, such as Proper orthogonal decomposition (POD) [11, 12], Proper Generalized Decomposition (PGD) [13], Information entropy reduction[14], different kinds of weak greedy RB methods [9], or even the empirical interpolation method (EIM) [15] and its generalized version (GEIM) [16] etc. Once the reduced basis are collected, a reduced-order model, which maps the input parameter space into the output solution space (also called input-output map), can be constructed in either an intrusive or a non-intrusive way. In the *online* stage, for a given new parameter configuration, the corresponding reduced basis solution is then derived as a linear combination of the reduced basis, and the coefficients of the reduced basis are computed by running the reduced-order model. We refer to [17, 18, 19] for a complete introduction to RB methods.

Most traditional RB methods are intrusive, requiring full knowledge of the governing equations and the discretization strategy of the full model of the physical system. The intrusive reduced-order model is generally constructed by projecting the full-order model onto the reduced space. The most popular and the simplest choice is Galerkin projection, in which the reduced basis or POD basis is used as the test functions [20, 21, 22]. In most of the real engineering applications, access to the governing equations, discretization frame and solver generally are unavailable when working with proprietary code or commercial software, in which the full model is implemented a priori and is always viewed as a black-box solver by users [23, 24]. Thus the intrusive nature limits the applications of traditional RB methods.

In contrast, non-intrusive RB methods aim to construct reduced order models from reduced basis, by building the input-output map through interpolation, regression or machine-learning-based methods. For example, the work in [25] reconstructs POD coefficients using a cubic spline interpolation. The work in [26, 27] builds the non-intrusive reduced-order model of the Navier-Stokes equations based on Radial Basis Function (RBF) interpolation. The Gaussian process regression [28, 29], self-organizing maps [30] are also widely used to build the map from inputs to reduced basis coefficients. The work in [31, 32] presents a non-intrusive model reduction method, which learns operators for the linear and polynomially nonlinear dynamics via a least-squares problem. The work in [33] builds non-intrusive reduced basis approach (NIRB) based on a two-grid finite element discretization scheme. Recently, different kinds of artificial neural networks [34, 35, 36] are trained to learn the map from inputs to reduced basis coefficients with applications in different domain. Because the reduced basis represents the physical model to some extent, this paradigm is also viewed as physics-informed or physics-based or physics-constrained machine learning for reduced-order modeling in some literature [37, 38, 39, 40, 41, 42].

RB methods also offer new opportunities for the integration of *simulation models* and *measurement data*. This is also a non-intrusive data-driven paradigm in approximation of PDEs, motivated by small-data scenarios, where simulation models are represented by reduced basis. These approaches are also regard as data assimilation [43, 44], where the data is incorporated into a model, see recent works [45, 46, 47, 48, 49] for general description of this paradigm. In this paradigm, RB provides actionable tools to compress prior knowledge about the system coming from the parameterized mathematical model into low-dimensional and more manageable forms, which makes the combination with measurement data more efficient. In the *off-line* phase, RB provides a way to learn the physical system, which allows extracting the principal component information of the system and also providing a guideline to set the amount and locations of the measurement needed; in the *on-line* phase, RB speeds up computations allowing better explorations of the parameter space at an acceptable computational cost.

It is important to note here that a great deal of work has been done in the nuclear engineering community to develop reduced order models alone or in the framework of data assimilation. The method which seems to have been more extensively employed is POD. Examples of applications to neutronics are [50] and [51, 52, 53]. In addition to these contributions, one can find in [54] an evaluation of the temperature reactivity feedbacks in a lead-cooled fast reactor with a POD basis. The works [55, 56, 57] apply the method to nuclear reactor core spatial kinetics and dynamics. In particular, [58] models the motion of control rods by applying reduced bases to parametrized multi-group neutron diffusion equations both in the time-dependent and stationary formulations. Further work [59] gives a technical road map in reduced-order modeling of parameterized multi-group diffusion k-eigenvalue problems, application of proper generalized decomposition to multigroup neutron diffusion eigenvalue calculations can be found in [60]. Comparison of Reduced-Basis techniques for the model order reduction of parametric incompressible fluid flows is shown in [61]. These approaches are referred to as intrusive ROM, in which one has to access the complicated nuclear simulation codes.

For the study of non-intrusive ROM, an early work [62] used polynomials to build the relation of the input parameter space and the output. Recently, there has been extensive research on non-intrusive ROM based on Gaussian Process Regression, examples can be found with application to nuclear component degradation [63, 64], facility deployment decisions[65], state identification in pressurized water reactors [66], and inverse uncertainty quantification [67, 68, 69]. Non-intrusive ROM based on spectral technique finds its application to multi-physics problem in [70]. More recent work [71] presents different methods such as regression and neural network-based training to correlate the input and output in building ROM-Based Surrogate Systems of EBR-II. Non-intrusive ROM using neural networks is studied in [72] for load follow operation, and Support Vector Regression Model is presented in [73] for the prediction of the power peaking factor in a Boron-Free Small Modular Reactor.

In the framework of data assimilation with reduced basis, the works in [45, 74, 47, 75, 76] build a systematical framework in reconstruction the neutronic field by using reduced basis and small-data measurement. Particularly, EIM [15, 77, 78] and GEIM [16, 79, 47], the gappy POD [74] and the parameterized-background data-weak (PBDW) [45, 80] data assimilation formulation can be adapted and integrated into this paradigm, large amounts of real engineering applications have proofed the efficiency of the proposed paradigm. The goal of the above data assimilation paradigm is state estimation. In some circumstances, reactor operators may be more interested in what input parameter caused the related observation or state. Thus input parameter identification (IPD) is another task of interest.

So far, however, there has been little discussion about digital twin using non-intrusive ROM for reactor operation support purpose. On behalf of non-intrusive ROM and observations, our goal in this work is to develop a prototype of a data-enabled physics-informed digital twin for the nuclear reactor physics applications, particularly for on-line monitoring purpose, aiming to predict the operational parameters i.e. IPD, and the caused neutronic field. the digital twin is designed to solve both forward and inverse problems. Specifically, the digital twin can solve forward problems given input parameters, as well as inverse problems given some extra measurements. At the heart of the digital twin is a non-intrusive reduced order model of the reactor core, the non-intrusive nature is realized via machine learning to build the map from inputs to reduced basis coefficients. With this non-intrusive machine learning reduced order model, the digital twin can solve forward problem given input parameters in a second. With this digital twin as a foundation, we build a data-driven digital twin by formulating an inverse problem in which online sensor data from a nuclear reactor core is used to infer which input parameter should comprise the digital twin at the current state. The framework of the machine learning reduced order modeling for data-enabled physics-informed digital twin is sketched in Fig 1.

We demonstrate the proposed methodology and illustrate its benefits of our contributions by a real engineering case study. We create a digital twin of HPR1000 reactor core from nuclear reactor physics aspect, to predict the power evolution induced by four general parameters, i.e., burnup of the nuclear fuel along the lifecycle, control rod movement, variation of temperature and pressure of the coolant. We demonstrate how the reduced order model scales efficiently to the full model of the reactor core, and how we construct a set of solution snapshot, extract a set of reduced basis, thereafter train machine learning models designed to enable the rapid reconstruction of the digital twin to a wide range of effective operation states. We then demonstrate how the reduced model enables the data-driven model adaptation, and how near real-time estimates of the effect of digital twin enables real-time decision-making for operation. We present simulation results for an illustrative of HPR1000, which shows it potential applications for on-line monitoring purpose.

The remainder of the paper is organized as follows. In Section 2, we describe the details of parametrization and the preparation of reduced model of neutronic field. In Section 3, we introduce the methodologies including KNN and DT to build the map from input parameter to the coefficients of the reduce models. In Section 4, we first describe the problem settings of the real industry problem in nuclear reactor domain, then the forward and inverse models are build for field reconstruction and input parameter identification. Numerous numerical results are also brought out in this section to proof the effectiveness of the digital twin. A brief summary is then presented in Section 5.

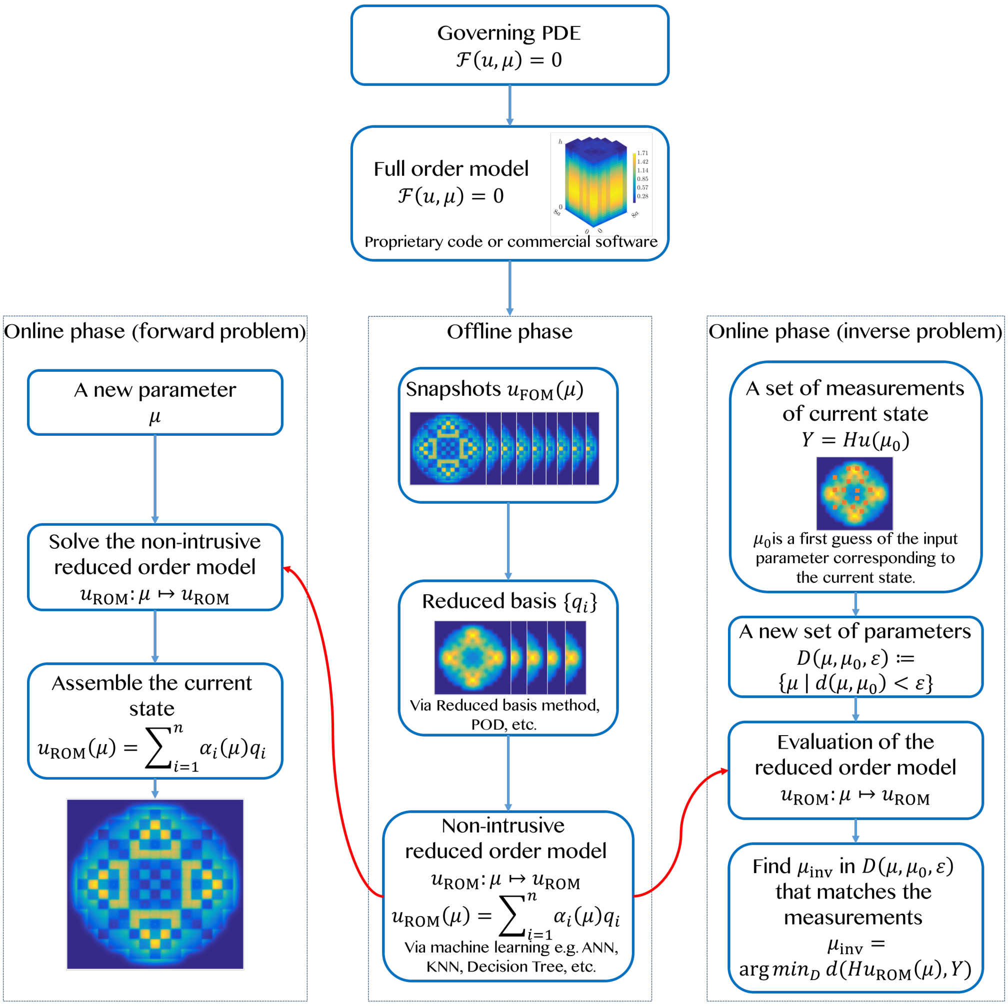


Figure 1: A flow chart of the machine learning reduced order modeling for data-enabled physics-informed digital twin.

2 Parametrization of neutronic fields

We consider systems in nuclear engineering particularly in nuclear reactor physics that respond to inputs with physical fields, for example representing such physical quantities as fast flux, thermal flux, power distribution, etc. This section establishes a physics-informed parametrization of such fields via governing equations and the proper orthogonal decomposition.

2.1 Numerical approximation of physical fields

Consider a reactor core that maps an input parameter onto a physical field i.e. power distribution field, which is our prediction quantity of interest. We notice here that the behavior of the reactor core is characterized by physical laws and governing equations, which are often represented in the form of partial differential equations, i.e., neutron transport equations or its approximations multi-group neutron diffusion equations [2]. We denote the power field as a function: $\Phi : \Omega \times \mathcal{D} \to \mathbb{R}$, with the spatial domain $\Omega \subset \mathbb{R}^d$ which represents the reactor core of dimension d, and input parameter domain $\mathcal{D} \subset \mathbb{R}^p$ of dimension p. Thus, the power field u varies in space, and depends on the input parameter of the system.

In a classical computational setting, numerical models discretize the governing equations, approximating the solution fields in different ways. For example, the finite element method builds the field using an expansion in a finite number of basis functions and the finite difference method builds the field at a set of discrete points in the space-time domain, while for the proprietary code packages such as CORCA-3D [23] and COCAGNE [24], different nodal methods are used which represent the field with node average value followed by pin power reconstruction in each node. Whichever numerical method is chosen, the result is a numerical model that embeds governing equations. For practical nuclear engineering applications, the dimensionality of these numerical models is typically high, e.g., in the range of thousands to millions of unknowns, or even more in three-dimensional time-varying simulations [23, 24, 7]. This means that a large-scale system of equations has to be solved to evaluate the physical model, which presents computational challenge in simulations requiring i) many-queries e.g., optimal control, inverse problems, uncertainty quantification etc., ii) real-time evaluations e.g., online monitoring, parameter estimation, etc.

The interest of this paper is on learning a non-intrusive approximate model of u from training data (solution snapshots) $\mathcal{S} \subset \{u(\boldsymbol{r}, \boldsymbol{\mu}) |\, \boldsymbol{r} \in \Omega, \boldsymbol{\mu} \in \mathcal{D}\}$ in a way that respects the underlying physical constraints of the reactor core, thus leading to the model endowed with physical interpretability and predictive ability. In our digital twin setting, we seek to learn a numerical model from data both from simulations offline and operation data online. We use the reduced order method, particularly machine learning non-intrusive reduced order method, to overcome the computational challenge which proprietary code faces. We then introduce the notion of the reduced order representation and parametrization of the fields in a reactor core. Such a representation can be achieved using reduced basis, particularly the POD basis [11, 74], which are typically computed using sampled training data, i.e., snapshots sampled from $\mathcal{S}$. The parametrization and the non-intrusive nature of the reduced order representation can be achieved using machine learning method, which will be introduced in next section.

2.2 Computing the reduced basis

The standard POD basis is computed via the method of snapshots [11]. Here, for self-consistency of this paper, we recall the most important properties of POD for reduced-order modeling. We consider a collection of N snapshots $\{u(\boldsymbol{\mu}_k)\}_{k=1}^{N} \subset \mathbb{R}^M$ corresponding to the discrete parameter set $\mathcal{D}_N = \{\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_N\} \subseteq \mathcal{D} \subset \mathbb{R}^p$, where M is the discrete spatial dimension and p is the dimension of the parameter space. These snapshots are numerical solutions generated by a numerical model, i.e., the neutronic core code package CORCA-3D in this paper.

The correlation matrix $\mathbf{C}$ is formed by computing the inner product between each pair of snapshots, that is,

$$\mathbf{C}_{i,j} = \frac{1}{N}(u_i, u_j), \quad \forall 1 \le i, j \le N, \tag{2.1}$$

where (u_i, u_j) is the inner product between u_i and u_j. The eigenvalues $\{\lambda_i\}$ and the corresponding eigenvectors $\left\{v^i\right\}$ of $\mathbf{C}$ are computed. The j-th POD basis vector q^j, which is independent of $\boldsymbol{\mu}$, is given by a linear combination of snapshots

$$q^j = \sum_{i=1}^{N} v_i^j \phi^i, \tag{2.2}$$

where v_i^j denotes the i-th element of the j-th eigenvector. The magnitude of the j-th eigenvalue λ_j describes the relative importance of the j-th POD basis vector. Define the snapshot matrix

$\boldsymbol{S} \in \mathbb{R}^{M \times N}$ which contains the snapshots $u(\boldsymbol{\mu}_k)$ as its columns. Assemble the first n POD basis in a matrix $\boldsymbol{V}_n = [q_1, ..., q_n] \in \mathbb{R}^{M \times n}$. Among all orthonormal bases of size n, the POD basis minimizes the least squares error of the reconstruction of snapshot matrix $\boldsymbol{S}$,

$$\min_{\boldsymbol{V}_n} \|\boldsymbol{S} - \boldsymbol{V}_n \boldsymbol{V}_n^T \boldsymbol{S}\|_F^2 = \sum_{k=n+1}^{N} \lambda_k \ . \tag{2.3}$$

Thus the POD basis yields an orthonormal basis that provides an efficient low-dimensional representation of the snapshots. For any $u(\boldsymbol{\mu}) \in \mathcal{S}$, the n-dimensional representation of $u(\boldsymbol{\mu})$ is approximated by

$$u(\boldsymbol{\mu})_n = \sum_{i=1}^{n} \alpha_i(\boldsymbol{\mu}) q_i \ . \tag{2.4}$$

3 Physics-informed machine learning of reduced-order model

Physics-based machine learning algorithms have been widely applied in high-dimensional industrial problems, including aerodynamics [42], air pollution simulation [81], electrical power systems [82, 83] or numerical weather prediction (NWP) [84]. Compared to traditional physics-based simulations, machine learning techniques show a significant strength of efficiency, especially when coupling with model reduction approaches, such as POD [81], domain localization [85] or image auto-encoder [86]. In this section, we describe the problem setup with a brief introduction to each machine learning algorithm used in this study. Here we focus on two types of machine learning algorithms: decision tree (DT) and k-nearest neighbors (KNN), widely adapted in industrial applications for regression or classification. Recently, deep learning (DL) approaches have also been broadly used in physics-based machine learning problems. However, DL methods often require massive amounts of data. Recent work of [42] shows the limitation of DL for field reconstruction with a small number of input variables. In this study, we look for efficient machine learning approaches to build the coefficients map, yielding a compromise of accuracy and computational simplicity. Later we will find that, KNN and DT are more fit for our specified complicated, non-linear, non-smooth and non-convexity real industry problem.

It is important to note here that non-intrusive ROM using Gaussian processes (GP), artificial neural network (ANN) and radial basis function (RBF) are also tested in this work though not shown , but none of these methods presents satisfactory accuracy for our specific engineering problem. The reason is that the model tested is non-smooth and non-continuous in the sense that the moving of control rods is step by step and the burnup calculated by industry code is also step by step. Further more, DT and KNN is more fit for limited data modelling approaches for engineering applications [87], that is exactly the circumstances we are interested in this work. Note also that our digital twin is problem dependent especially for the choice of machine learning method when constructing non-intrusive ROM.

3.1 Setup of the experiment

Our objective is to build data-driven non-intrusive surrogate models to release the computational burden for both forward and inverse simulations. More precisely, we attempt to predict POD modes by observing $\boldsymbol{\mu}$ in the forward model, leading to an efficient field reconstruction through $u(\boldsymbol{\mu})$. In this study, the hyperparameters in machine learning models are chosen according to conducted experiments. The determination of the optimal truncation parameter n^o (i.e. number of modes in POD projection) is crucial for both the machine learning predictions and the posterior field reconstruction [88]. n^o should be determined regarding both the aspects of reconstruction accuracy and computational complexities. As for the inverse modelling, where only partial observations of the entire physical field are available, Bayesian inference (BI) or data assimilation (DA) approaches [77, 89] can be applied. However, these methods require either estimation of the posterior probability density (for BI) or an explicit differentiable forward equation (for DA), which are cumbersome for DT and KNN functions. In this work, we implement a naive approach where an ensemble of samplings $\{\boldsymbol{\mu}\}_i^\rho$ is generated using Latin Hypercube sampling (LHS) around the initial guess $\boldsymbol{\mu}_\mathrm{I}$. The ensemble size n^s is fixed to 100 or 1000 in this study, i.e. $\rho = 1..n^s$. Therefore, the solution of the inverse problem $\boldsymbol{\mu}^*$ can be obtained after applying the forward model to all samplings,

$$\mu^* = \underset{k=1..n^s}{\operatorname{argmin}}(||HV_nF^{\mathrm{ML}}(\boldsymbol{\mu}_k) - Y_\mathrm{o}||_2), \tag{3.5}$$

where $F^{\mathrm{ML}} : \mathbb{R}^4 \to \mathbb{R}^n$ stands for the machine learning forward model which maps the parameter $\boldsymbol{\mu}$ to the POD coefficients space. Y_o and H represent the observation vector and the state-observation transformation function respectively.

As noted before, the naive approach using Latin Hypercube sampling (LHS) is fit for this work because the discrete manifold provided by industry code is piecewise continuous, thus other smarter methods which depend on continuous property of manifold performs not as well as KNN or DT.

3.2 Decision tree regression

Decision tree could be considered as a localization method [90, 91] by building regression models in a tree structure [92]. Briefly speaking, It breaks down a dataset into smaller subsets while developing an associated decision tree at the same time. Two types of nodes could be found in a decision tree: decision nodes and leaf nodes. The former divides the tree structure into further branches while the latter is associated with specific classes/clusters. Recent studies also demonstrate the good performance of DT-based approaches when dealing with correlated input variables [93], which is crucial when analyzing systems with uncertainties [94].

It is important to point out here that, DT is wildly use in nuclear engineering field, e.g., it can be used for the reconstruction of the neutron noise source in the nuclear reactor cores [95], for nuclear power plant performance optimization [96], for the prediction of 3D nuclear reactor's operational parameters from 2D fuel lattice design information [97] and for nuclear reactor transient diagnostics [98] etc.

In this work, we use DT regression to predict $\alpha_i(\boldsymbol{\mu})$ A decision tree arrives at an estimation by splitting the global dataset into several small ones, depending on the depth of the tree. More precisely, the splits take place at each node of the tree regarding a threshold of an input variable decide by the learning algorithm. This algorithm then learn a specific value of the regression problem for each sub-dataset which is determined by the ensemble of nodes [92]. The quality of split is quantified by the mean square error (MSE) ϵ^{MSE}, defined as :

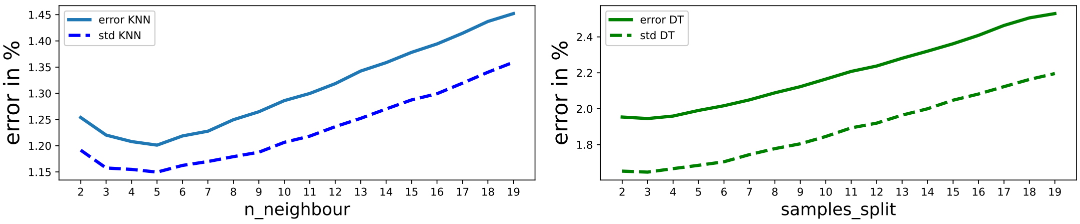
$$\epsilon^{\mathrm{MSE}} = \frac{1}{n_{\mathrm{train}}}\sigma(\alpha_i(\boldsymbol{\mu}) - \alpha_i^{\mathrm{predict}}(\boldsymbol{\mu}))^2 \tag{3.6}$$

where n_{train} denotes the size of the training dataset. The minimum number of split regarding a node is set to 2. The choice of this parameter is determined via experiments where the prediction error is estimated on an independent validation dataset, which is not a part of training or test data, as shown in Fig 2(b). We observe that the predicition error increases against the the minimum number of split while the training time varies little (from $0.7s$ to $0.95s$ on the whole training set).

3.3 k-nearest-neighbors regression

K-nearest neighbors algorithm is widely used for classification and regression problems (e.g [99]), example used in nuclear reactor physics is shown in [100] for control rod position reconstruction, with a similar application scenarios as in our work. The input and output variables in KNN are established over a local set of training samples. As for evaluation, the prediction value of a certain input is calculated as the average of the $k(k \in \mathbb{N}^+)$ neighbours. The Euclidean metric is used in this study. Since the complexity of KNN is proportional to the input dimension [42], from a computational point of view, it is appropriate for coefficients mapping in this study. We refer to [100] for the detail implementation of KNN in this work.

The non-parametric nature of KNN ensures the flexibility of the algorithm when dealing with complex multivariate systems. In this work, the number of neighbors $n_{\text{neighbors}}$ used for updating n^{KNN} is fixed to 5, which is a reasonable choice for relatively small-scale systems. This choice is confirmed by experiments on the validation data shown in Fig 2(a) where the minimum of the prediction error is reached when $n_{\text{neighbors}} = 5$. These experiments are carried out using a validation dataset which includes 3000 samples different from the training and test dataset. The leaf size is maximized by 30, considering a reasonable computer memory consuming. For more details about DT and KNN models, interested readers are referred to [101].



(a) (b)

Figure 2: The averaged prediction error and standard deviation on the validation dataset of 3000 samples for KNN(a) and DT(b), against respectively the number of neighbors and the minimum number of split.

4 Reactor physics application

4.1 Neutronic field simulation

In this part, we introduce the physical model that will be used to test the method introduced above. This part is original introduced in [102] for the study of data assimilation. For the sake of completeness, we repeat here the background of the two examples and give the detail problem settings for the testing purpose. The engineering background it to reconstruct the neutronic field of a 3D realistic nuclear reactor ,i.e., the Pressured Water Reactor (PWR) HPR1000 [103]. HPR1000 reactor core and primary loop is a third generation PWR system designed by Nuclear Power Institute of China (NPIC) based on the experiences of research, design, manufacture, construction and operation of existing pressurized water reactor power plants, and the experience feedback from Fukushima nuclear accident, and considering the preeminent design concept of world-wide advanced nuclear power plants. We refer to [102] for a glance at HPR1000 and the generic neutronic physical model of the reactor. For the sake of completeness, we list here the description of the model with minor modification.

The core is filled with a total of 177 vertical nuclear fuel assemblies, among those 44 are instrumented with self powered neutron detectors (SPNDs) to measure the neutronic activity fields. A horizontal slice of the HPR1000 core and an axial slice of an assembly are represented in Fig. 3. Only one quarter is given because the rest can be inferred by symmetry along the x and y axis. The fuel assemblies in gray represent the assemblies with control rod, and those marked with D present the assemblies with SPNDs.

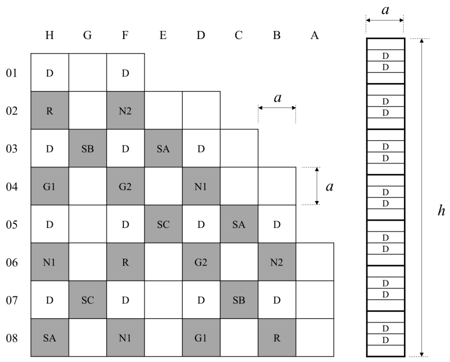


Figure 3: A quarter of the core in radial direction (white square: fuel assembly with SPNDs, gray square: fuel assembly with control rods, D: fuel assembly with neutron detectors).

All the simulations of HPR1000 are based on the advanced node core code package CORCA-3D [23]. CORCA-3D is one of the independent developed codes of NPIC for the nuclear reactor core design. This package can calculate the few-group cross section, solve the 3D diffusion equations, consider the thermal-hydraulic feedback, reconstruct the pin-by-pin power. It has lots of functions such as changing core status calculation, critical searching, control rod value calculation, coefficient calculation and so on.

In the simulation of nuclear reactor core in steady state, the neutron flux $\phi = (\phi_1, \phi_2)$ over the core domain Ω is modeled by two-group neutron diffusion equation with suitable boundary conditions. Index 1 denotes the high energy group and index 2 the thermal energy group. The flux is the solution to the following eigenvalue problem (see [1]). To be precise, the flux ϕ satisfies the following eigenvalue problem: Find $(\lambda, \phi) \in \mathbb{C} \times L^{\infty}(\Omega) \times L^{\infty}(\Omega)$, s.t.

$$\begin{cases} -\nabla\left(D_1 \nabla \phi_1\right) + \left(\Sigma_{a,1} + \Sigma_{1\to 2}\right)\phi_1 = \lambda \chi_1 \left(\nu\Sigma_{f,1}\phi_1 + \nu\Sigma_{f,2}\phi_2\right), \\ -\nabla\left(D_2 \nabla \phi_2\right) + \Sigma_{\mathrm{a},2}\phi_2 - \Sigma_{1\to 2}\phi_1 = \lambda \chi_2 \left(\nu\Sigma_{f,1}\phi_1 + \nu\Sigma_{f,2}\phi_2\right), \end{cases} \tag{4.7}$$

with the zero boundary condition $\phi_1 = \phi_2 = 0$ on $\partial\Omega$. The generated nuclear power is $P = \nu\Sigma_{f,1}\phi_1 + \nu\Sigma_{f,2}\phi_2$. The following parameters are involved in the above equation:

- D_i: the diffusion coefficient of group i with $i \in \{1, 2\}$;
- $\Sigma_{a,i}$: the macroscopic absorption cross section of group i;
- $\Sigma_{1\to 2}$: the macroscopic scattering cross section from group 1 to 2;
- $\Sigma_{f,i}$: the macroscopic fission cross section of group i;
- ν: the average number of neutrons emitted per fission;
- χ_i: the fission spectrum of group i.

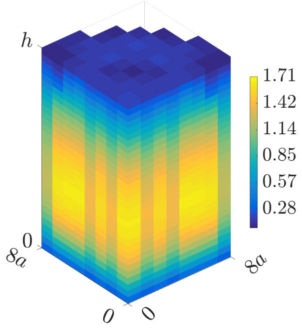


Figure 4: Example of the 3D power distribution over the core in a realistic HPR1000 reactor at CNNC, calculated by CORCA-3D code.

We make some comments on the coefficients and recall well-posedness results of the eigenvalue problem Eq. (4.7). First of all, the first four coefficients (D_i, $\Sigma_{a,i}$, $\Sigma_{s,1\to 2}$ and $\Sigma_{f,i}$) might depend on the spatial variable. In the following, we assume that they are either constant or piecewise constant so that our set of parameters is

$$\mathcal{U} = \{D_1, D_2, \Sigma_{a,1}, \Sigma_{a,2}, \Sigma_{1\to 2}, \nu, \Sigma_{f,1}, \Sigma_{f,2}, \chi_1, \chi_2\}. \tag{4.8}$$

Under some mild conditions on the parameters $\mathcal{U}$, the maximum eigenvalue $\lambda_{\max}$ is real and strictly positive (see [104, Chapter XXI]). The associated eigenfunction ϕ is also real and positive at each point $\boldsymbol{r} \in \Omega$ and it is the flux of interest. In neutronics, it is customary to use the inverse of $\lambda_{\max}$, that is called the multiplication factor $k_{\text{eff}} := \frac{1}{\lambda_{\max}}$. Here, for each parameter setting $\mathcal{U}$, k_{eff} is determined by the solution to the eigenvalue problem 4.7 with CORCA-3D code. Wr give an example of the 3D power distribution in the core of HPR1000 reactor calculated by CORCA-3D in Fig. 4.

4.2 Solution manifold

We adapt the example which was fist presented in [102] and later in [74] to simulate the typical operation history of HPR1000 reactor. During the normal operation of HPR1000, two types control rods are used to control the reactor, the first one is compensation rods, which used for coarse control and/or to remove reactivity in relatively large amounts, in HPR1000, there are four subtypes compensation rods, namely, G1,G2,N1,N2; the second one is regulating rods (R rods), which are used for fine adjustments and to maintain desired power or temperature [103]. We consider the power evolution induced by i) the movement of the control rods, ii) the burnup of the nuclear fuel in the whole core, iii) the variation of core coolant temperature at inlet and iv) the variation of the power level of the reactor core. The evolution of the states (ϕ_1, ϕ_2 and P) are modeled by Eq. (4.7) with specific values of the coefficients as in Eq. (4.8). Thus, the entries of $\mathcal{U}$ vary in space depending on the materials contained in each fuel assembly. In our simulation, their values are limited to depend on four macroscopic parameters which give the stage of the life cycle:

- Bu: the average burnup of the fuel in the whole core. It is a measure of how much energy is extracted from the fuel so it is an increasing function in time. It ranges between 0 (for the beginning of the life cycle) and $Bu_{\max}$ (the end of the life cycle, in this work, we set $Bu_{\max}$=25000 MWd/tU) and its exact evolution depends on the operating history of the reactor.

- St: the inserting steps of control rods, it ranges from 0 to 615. Consider the movement of the compensation rods from all rod clusters out (ARO) to fully inserted (615 steps, considering the overlap steps).
- Pw: the power level of the reactor core. It ranges between 0.3 and 1.
- Tin: core coolant temperature at inlet.

Therefore, the parameter $\mathcal{U}$ depends on the vector of "general" parameters

$$\boldsymbol{\mu} := (Bu, St, Pw, Tin). \tag{4.9}$$

We analyze the power field, although the methods in this article could be applied to fast flux, thermal flux, or any other field variables of interest. The power field can be represented with $u(\boldsymbol{\mu}) = u(\boldsymbol{r}, Bu, St, Pw, Tin)$, where $\boldsymbol{r} \in \Omega$.

To simulate the neutronic process durning the operation period, CORCA-3D code is used in a standard configuration, by solving Eq. (4.7). There are 177 fuel assemblies in HPR1000, each assembly is numerically represented using 28 vertical levels. Thus, the size of the physical field (or vector) u is 4956 (177×28). The training set

$$\boldsymbol{S} := \{u(\boldsymbol{\mu}) | \boldsymbol{\mu} \in D^s\}, \tag{4.10}$$

consists of 18480 solution snapshots with the parameter sampling scheme

$$D^s := Bu^s \otimes St^s \otimes Tin^s \otimes Pw^s, \tag{4.11}$$

where $Bu^s = \{$ 0, 50, 100, 150, 200, 500, 1000, 1500, 2000, 2500$\}$, $St^s = \{0, 1, ..., 615\}$, $Pw^s = RU3(0, 100)$ and $Tin^s = RU3(290, 300)$, the operator $RU3(a, b)$ represents a random uniform sampling in the closed set $[a, b], (a < b)$ for three times.

The locations of the sensors which measure the node power can be found in Fig. 3, and the size of the observation vector Y_{o} is 308 (44×7). In this paper, observations used in the analysis process are not coming from real core measurements, but coming from numerical simulations with CORCA-3D. We further assume each observation $\mathrm{y_i}$ is noise-free.

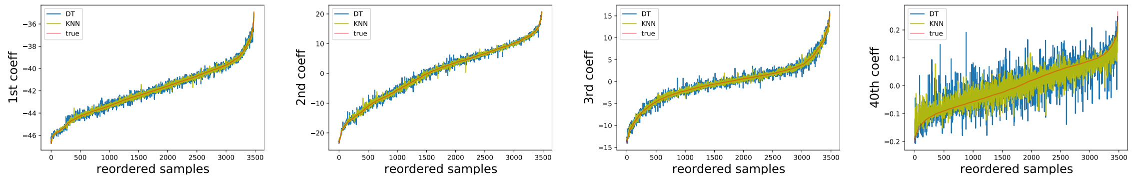
4.3 Forward problem for field prediction

In this section, we will apply the digital twin to model neutronic problem given initial conditions, i.e., burnup, control rod inserting step, power level and coolant temperature at inlet of the reactor core. Our goal it to predict the neutronic field or the power distribution, and compare the results with the output of the neutronic code and the related POD approximation.

To build the forward model of the digital twin, following the methodology described in section 3, we train both DT and KNN models with the same training set of size $n^{\text{training}} = 15000$, randomly chosen from the original 18480 couples of $\{\boldsymbol{\mu}, \alpha(\boldsymbol{\mu})\}$. The rest $n^{\text{test}} = 3480$ couples are used as the test dataset to evaluate the performance of machine learning algorithms. The optimal hyperparameters of KNN and DT models are turned as explained in Section 3. The output of the machine learning algorithms are the coefficients of POD modes of the physical field, estimated in previous study. The methodology is shown in Algorithm 15. The prediction results, together with the true value of $\alpha_1(\boldsymbol{\mu}), \alpha_2(\boldsymbol{\mu}), \alpha_3(\boldsymbol{\mu})$ and $\alpha_{40}(\boldsymbol{\mu})$, are displayed in Fig. 5. The samples are reordered for each sub-figures (a-d) in the increasing order of $\alpha_i(\boldsymbol{\mu})$. For $i = 1, 2, 3$, both DT and KNN manage to provide accurate predictions. The error level arise against the increase of i since we move towards less representative principle components of the physical field.

Algorithm 1 Machine learning forward functions

```
Training
---
Inputs:
Parameters: $\{\boldsymbol{\mu}_j\}_{j=1}^{n^{training}}$
POD coefficients: $\{\alpha_i(\boldsymbol{\mu}_j)\}_{i=1,j=1}^{n,n^{training}}$,   where $n$ is the number of POD models used.
Set: $F^{\mathrm{ML}}$ = KNN ($n^{\mathrm{KNN}}$ neighbors = 5, max leaf size = 30) or DT (quality measure: Gini, min split/node = 2)
Fit: $F^{\mathrm{ML}}$ model with $(\{\boldsymbol{\mu}_j\}, \{\alpha_i(\boldsymbol{\mu}_j)\})$
outputs: $F^{\mathrm{ML}}$
---
Online evaluating
---
Inputs:
The first $n$ POD models: $\{q_i\}_{i=1}^{n}$
For a given parameter: $\boldsymbol{\mu}$
Do: select ML=KNN or DT
Outputs:
Learned coefficients: $\{\alpha_i(\boldsymbol{\mu})\}_{i=1}^{n} = F^{\mathrm{ML}}(\boldsymbol{\mu})$
Learned reconstructed field: $u^{\mathrm{ML}}(\boldsymbol{\mu}) = \sum_{i=1}^{n} \alpha_i(\boldsymbol{\mu}) q_i$
```



(a) $\alpha_1(\boldsymbol{\mu})$ (b) $\alpha_2(\boldsymbol{\mu})$ (c) $\alpha_3(\boldsymbol{\mu})$ (d) $\alpha_{40}(\boldsymbol{\mu})$

Figure 5: Comparison of machine learning predictions against true values in the test dataset reordered in an increasing order for each of the input variables.

We draw the evolution of the averaged reconstruction error $||V_n F^{\mathrm{ML}}(\boldsymbol{\mu}) - u(\boldsymbol{\mu})||_2/||u(\boldsymbol{\mu})||_2$ and the computational time T^{ML} (in a laptop) against the POD mode n, respectively in Fig. 6 (a) and (b). To compare, we also illustrate the error provided by POD method, i.e., the orthogonal projection of the true field on to the n-dimensional reduced POD space. In summary, numerical results are reasonable, POD method provides the best result, both KNN and DT approaches show a competitive performance where the error percentage remains inferior to 3 % when $n > 10$. In terms of reconstruction accuracy, DT is slightly over performed by KNN with less computational time.

In order to determine the optimal truncation parameter n^o, we introduce an indicator, i.e., the performance-computation ratio,

$$E_n = \frac{1}{2}\frac{||V_n F_n^{\mathrm{ML}}(\boldsymbol{\mu}) - u||_2}{||V_{n=150} F_{n=150}^{\mathrm{ML}}(\boldsymbol{\mu}) - u||_2} + \frac{1}{2}\frac{T_n^{\mathrm{ML}}}{T_{n=150}^{\mathrm{ML}}} \tag{4.12}$$

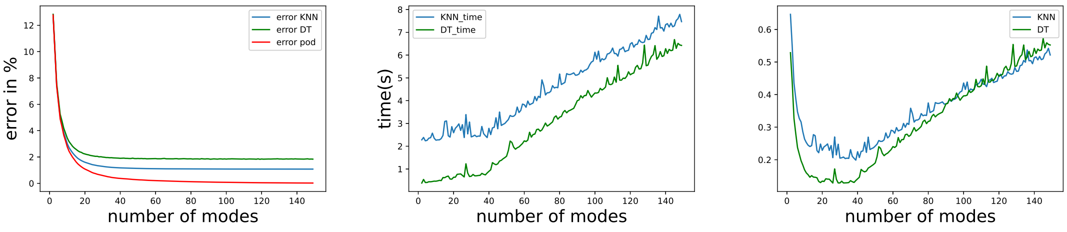
defined as the algebraic sum of the normalized reconstruction error and the normalized computational time. Since runing machine learning algorithms with all 150 POD modes is not computationally expansive in this application, the choice of the indicator E_n is conceptual. For large scale engineering problems, a fine tuning of optimal weight for the computational time and the prediction accuracy is required. The evolution of the averaged value of E_n over the test set, against the truncation number n as shown in Fig. 6 (c), result in the choice of $q_o = 40$ for both DT and KNN. This choice stands for a compromise of the prediction accuracy and the computational efficiency. It is used as a standard choice later in the inverse modelling. We display in Fig. 7, the true and the reconstructed physical field of the seventh floor of the reactor core (see Fig. 3) for three examples $\boldsymbol{\mu}_a, \boldsymbol{\mu}_b, \boldsymbol{\mu}_c$ (see Table 1) by the forward models driven by KNN and DT. The normalized reconstruction error (i.e., $||V_n F_n^{\mathrm{ML}}(\boldsymbol{\mu}) - u(\boldsymbol{\mu})||_2/||u(\boldsymbol{\mu})||_2$) is shown in Table 1. Again, we find that the error distribution of $u^{\mathrm{POD}}, u^{\mathrm{KNN}}$ and u^{DT} with respect to true field u^{true} is consistent with the curves shown in Fig. 6 (a). Besides, all the errors are acceptable (below 5%) in engineering point of view. The averaged online reconstruction time (which doesn't include the training time of DT and KNN) of different approaches is shown in Table 2. A significant strength of machine learning methods in terms of computational efficiency compared to CORCA-3D is notified. We remind that if the reconstruction problem is based on a finer mesh, the computational cost of CORCA-3D can be considerably increased while little impact can be found for the online evaluation of DT and KNN.

	μ^i				reconstruction error		
	μ^1	μ^2	μ^3	μ^4	POD	KNN	DT
$\boldsymbol{\mu}_a$	344	100	79.6	292.0	0.55%	2.1%	2.2%
$\boldsymbol{\mu}_b$	107	150	84.6	298.9	0.59 %	2.4%	2.7%
$\boldsymbol{\mu}_c$	473	50	89.3	293.2	0.65%	1.8%	1.8%

Table 1: Normalized reconstruction error of machine learning methods with 40 POD modes for three samplings of μ^i.

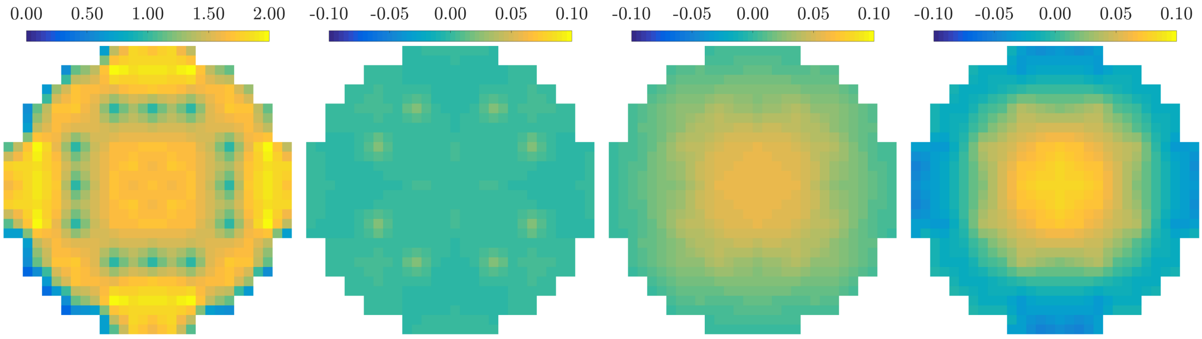
	KNN	DT	CORCA-3D
Evalutation time	$0.028s$	$0.024s$	~ 35s

Table 2: Averaged evaluation time of $\boldsymbol{\mu}_a, \boldsymbol{\mu}_b, \boldsymbol{\mu}_c$ (40 POD modes for KNN and DT) for one time-step using different approaches

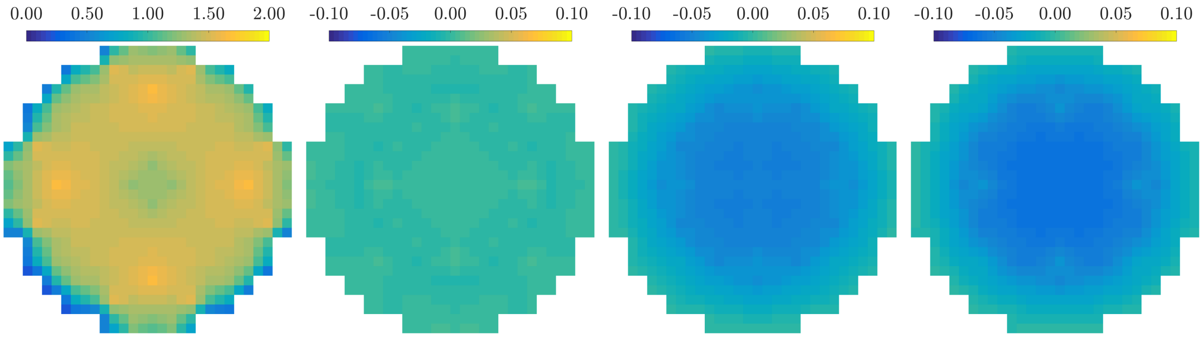


(a) (b) (c)

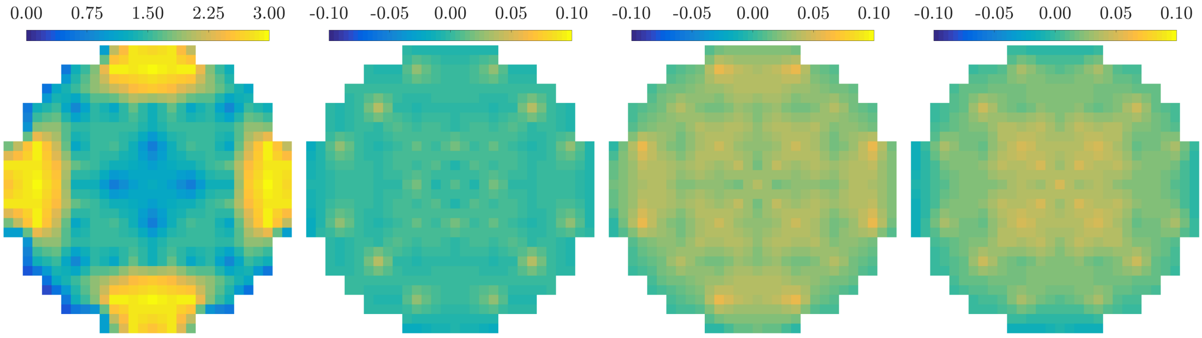
Figure 6: (a): The evolution of error in percentage for KNN and DT, compared to the POD error; (b): The calculation time (in seconds) which includes both training and evaluating for KNN and DT approaches on a laptop CPU; (c): Algebraic sum of the prediction error and the calculation time.



(a) $u(\boldsymbol{\mu}_a)$ (b) error of $u^{\text{POD}}(\boldsymbol{\mu}_a)$ (c) error of $u^{\text{KNN}}(\boldsymbol{\mu}_a)$ (d) error of $u^{\text{DT}}(\boldsymbol{\mu}_a)$



(e) $u(\boldsymbol{\mu}_b)$ (f) error of $u^{\text{POD}}(\boldsymbol{\mu}_b)$ (g) error of $u^{\text{KNN}}(\boldsymbol{\mu}_b)$ (h) error of $u^{\text{DT}}(\boldsymbol{\mu}_b)$



(i) $u(\boldsymbol{\mu}_c)$ (j) error of $u^{\text{POD}}(\boldsymbol{\mu}_c)$ (k) error of $u^{\text{KNN}}(\boldsymbol{\mu}_c)$ (l) error of $u^{\text{DT}}(\boldsymbol{\mu}_c)$

Figure 7: Comparison of machine learning predictions against true values for $\boldsymbol{\mu}_a, \boldsymbol{\mu}_b, \boldsymbol{\mu}_c$.

4.4 Inverse simulation for input parameter identification and field reconstruction

In this section, we test the performance of the inverse model of the digital twin. Specifically, we focus on neutronic field with unknown operational conditions. We assume that we only have partial measurements on the neutronic field inside the computational domain, u.e. reactor core. The objectives are i) to infer the entire neutronic field based on a limited number of scattered observations on power with the aid of the reduced basis, ii) to infer the operational conditions, which are the input parameters of the forward model of the digital twin.

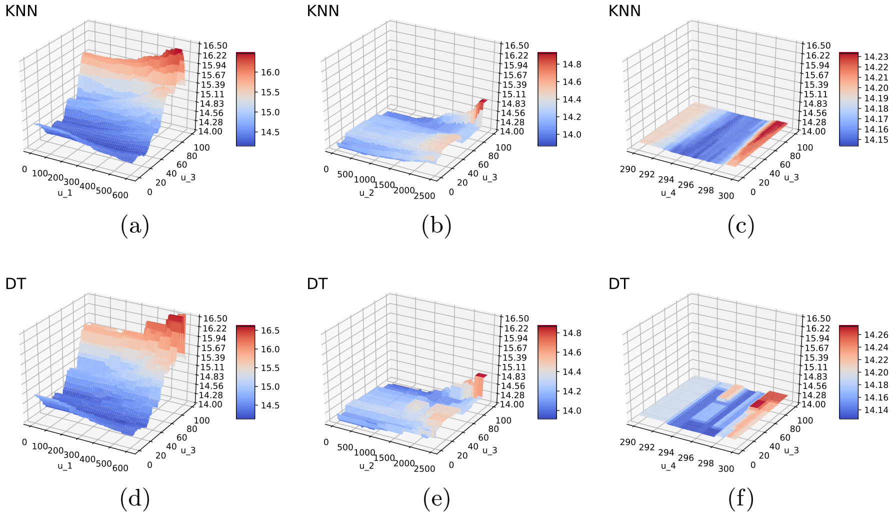
The inverse problem often remains cumbersome in industrial applications, because of, for instance, the large dimension or the non-differentiability of the forward function. We first describe the problem setting of the inverse problem that will be studied in this section.

Inverse problem: Given a set of noise-free observations $Y_o := [y_1, ..., y_m]^T$ of the true field u, which relates to the true parameter $\boldsymbol{\mu}_{\text{true}}$, and a first guess of the parameter $\boldsymbol{\mu}_{\text{initial}}$ which is close to $\boldsymbol{\mu}_{\text{true}}$. The inverse problem is to find an optimal parameter $\boldsymbol{\mu}^*$ that is closer to $\boldsymbol{\mu}_{\text{true}}$ than $\boldsymbol{\mu}_{\text{initial}}$. In mathematic point of view, given a measure $d_{\mathbb{R}^d}(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2)$ and $d_{\mathbb{R}^m}(Y_1, Y_2)$, the inverse problem it to find $\boldsymbol{\mu}^*$ that $d_{\mathbb{R}^d}(\boldsymbol{\mu}^*, \boldsymbol{\mu}_{\text{true}}) \le d_{\mathbb{R}^d}(\boldsymbol{\mu}_{\text{initial}}, \boldsymbol{\mu}_{\text{true}})$, satisfies $\boldsymbol{\mu}^* = \underset{\boldsymbol{\mu}\in\mathbb{R}^d}{\text{argmin}} d_{\mathbb{R}^m}(Y_o(u(\boldsymbol{\mu})), Y_o(u(\boldsymbol{\mu}_{\text{true}})))$.

By recognizing for a given $\boldsymbol{\mu}_{\text{true}}$ and the related observations $Y_o(u(\boldsymbol{\mu}_{\text{true}})))$, a initial guess $\boldsymbol{\mu}_{\text{initial}}$ that satisfies $d_{\mathbb{R}^d}(\boldsymbol{\mu}_{\text{initial}}, \boldsymbol{\mu}_{\text{true}}) < \delta$, where $\delta > 0$ is a small real number. There exists $\epsilon > 0$, such that $d_{\mathbb{R}^m}(Y_o(u(\boldsymbol{\mu}_{\text{initial}})), Y_o(u(\boldsymbol{\mu}_{\text{true}}))) < \epsilon$. Define a ball $B_{\mathbb{R}^d}(\boldsymbol{\mu}_{\text{initial}}, \delta) := \{\boldsymbol{\mu} \in \mathbb{R}^d, d_{\mathbb{R}^d}(\boldsymbol{\mu}, \boldsymbol{\mu}_{\text{initial}}) < \delta\}$ centered at $\boldsymbol{\mu}_{\text{initial}}$, the following problem is defined.

$$\text{Find} \quad \boldsymbol{\mu}^* \text{ s.t. } \quad \boldsymbol{\mu}^* = \underset{\boldsymbol{\mu}\in B_{\mathbb{R}^d}(\boldsymbol{\mu}_{\text{initial}},\delta)}{\text{argmin}} \quad d_{\mathbb{R}^m}(Y_o(u(\boldsymbol{\mu})), Y_o(u(\boldsymbol{\mu}_{\text{true}}))). \tag{4.13}$$

In order to illustrate the sensitivity of the machine learning forward models to the four input parameters, we show in Fig. 8 the evolution of the euclicidean norm of the observation vector (i.e $||HV_nF^{\text{ML}}(\boldsymbol{\mu}_k)||_2$) against a pair of input parameters (respectively $(\mu_1, \mu_3), (\mu_2, \mu_3)$ and (μ_4, μ_3)). More precisely, $[\mu_1, \mu_3, \mu_3, \mu_4] \in \{(0, 615), (0, 2500), (0, 100), (290, 300)\}$ while $\mu_1 \equiv 130$ for (b,c,e,f), $\mu_2 \equiv 1500$ for (a,c,d,f) and $\mu_4 \equiv 300$ for (a,b,d,e). From Fig. 8, we observe that the norm of the observation is more sensitive to input parameters $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_3$. These figures also demonstrate the non-convex nature of both KNN and DT functions, leading to difficiulities in the inverse modelling.



(a) (b) (c)

(d) (e) (f)

Figure 8: The evolution of $||HV_nF^{\text{ML}}(\boldsymbol{\mu}_k)||_2$ against different choices of parameters

4.4.1 Inverse problem for $\{\mu^1, \mu^2, \mu^3, \mu^4\}$

In this work, we introduce an algorithm schema based on the forward models proposed in the previous section and the LHS sampling strategy. More precisely, to represent/approximate $B_{\mathbb{R}^d}(\boldsymbol{\mu}_{\text{initial}}, \delta)$, n^s samplings of $\boldsymbol{\mu}$ are generated uniformly from a local domain (i.e., $[\mu^1_{\text{initial}} \pm 20, \mu^2_{\text{initial}} \pm 100, \mu^3_{\text{initial}} \pm 30, \mu^4_{\text{initial}} \pm 10]$) of the initial guess $\boldsymbol{\mu}_{\text{initial}} = [\mu^1_{\text{initial}}, \mu^2_{\text{initial}}, \mu^3_{\text{initial}}, \mu^4_{\text{initial}}]$. Then run the forward models as mentioned in Section 3.1 n^s times for each sampling and a sub-optimal $\boldsymbol{\mu}^*$ is obtained through Eq. 4.13 or Eq. 3.5. Thereafter, the neutronic field is reconstructed through $u^{\mathrm{ML}}(\boldsymbol{\mu}^*) = V_n F^{\mathrm{ML}}(\boldsymbol{\mu}^*)$. In order to ensure the robustness of our approach, six couples of $(\boldsymbol{\mu}_{\text{true}}, \boldsymbol{\mu}_{\text{initial}})$, listed in Table 3, are used to test the proposed inverse model.

$\boldsymbol{\mu}_{\mathbf{true}}$	μ^1	μ^2	μ^3	μ^4	$\boldsymbol{\mu}_{\mathbf{initial}}$	μ^1	μ^2	μ^3	μ^4
$\boldsymbol{\mu}_{\mathbf{true},1}$	0	0	61.11	291.76	$\boldsymbol{\mu}_{\mathbf{initial},1}$	20	100	61.11	291.76
$\boldsymbol{\mu}_{\mathbf{true},2}$	200	500	73.51	290.54	$\boldsymbol{\mu}_{\mathbf{initial},2}$	210	600	71.51	293.54
$\boldsymbol{\mu}_{\mathbf{true},3}$	400	1000	28.03	298.74	$\boldsymbol{\mu}_{\mathbf{initial},3}$	400	1000	18.00	293.54
$\boldsymbol{\mu}_{\mathbf{true},4}$	500	1000	52.96	291.51	$\boldsymbol{\mu}_{\mathbf{initial},4}$	500	1000	35.00	297.54
$\boldsymbol{\mu}_{\mathbf{true},5}$	130	1500	52.15	291.01	$\boldsymbol{\mu}_{\mathbf{initial},5}$	120	1700	52.15	300.00
$\boldsymbol{\mu}_{\mathbf{true},6}$	130	1500	52.15	291.01	$\boldsymbol{\mu}_{\mathbf{initial},6}$	130	1480	52.15	300.00

Table 3: The values of $\boldsymbol{\mu}_{\mathbf{true},i}$ and $\boldsymbol{\mu}_{\mathbf{initial},i}$ for $i = 1..6$.

The sub-optimal $\boldsymbol{\mu}^*$ obtained using $n^s = 100$ and $n^s = 1000$ samplings are respectively shown in Table 4 and Table 5. We display in Fig. 9 (for KNN inverse model) and Fig. 10 (for DT inverse model), the error distribution of seventh floor of the reactor core (see Fig. 3) for different type of reconstructed field (including $u(\boldsymbol{\mu}_{\text{true},i})$, $u(\boldsymbol{\mu}_{\text{initial},i})$, $u(\boldsymbol{\mu}_{\text{ML},100,i})$ and $u(\boldsymbol{\mu}_{\text{ML},1000,i})$), regarding the true field $u(\boldsymbol{\mu}_{\text{true},i})$, $i = 1, ..., 6$. In brief, both approaches manage to determine a set of parameters closer to the true values, regarding the initial guess. A significant advantage of LHS($n^s = 1000$) compared to LHS($n^s = 100$), can be observed (especially for μ^{1*} and μ^{3*} of $\boldsymbol{\mu}^*$) for both KNN and DT. The reason why μ^{1*} and μ^{3*} match the true parameter μ^1_{true} and μ^3_{true} well is that, the power distribution is more sensitive to he movement of the control rod (μ^1) and the operational power level (μ^3), meanwhile, the burnup of the current state and the temperature of the coolant show less impact on the power distribution, in the given parameter domain.

To show the effectiveness of the field reconstruction with the inverse model, we show in Table 6 the reconstruction error in the field space (i.e., $||u^{\mathrm{ML}}(\boldsymbol{\mu}^*) - u(\boldsymbol{\mu}_{\text{true}})||_2/||u(\boldsymbol{\mu}_{\text{true}})||_2$)) together with POD reconstruction error using 40 POD modes. The reconstruction error in the observation space (i.e., $||HV_nF^{\mathrm{ML}}(\boldsymbol{\mu}^*) - Y_{\mathrm{o}}||_2/||Y_{\mathrm{o}}||_2$) is shown in Table 7. For most cases, the posterior error ranges from 0.5% to 1.5% for $n^s = 1000$ and $n^s = 100$, which has been consequently reduced regarding the initial error. The computational time, mainly due to the evolution of ML models on the sampling ensemble, varies mostly between $2s$ ($n^s = 100$) and $20s$ ($n^s = 1000$) which is considerably smaller than the use of the full physics-based model, the latter is around 30 $\sim$ 40s.

$\boldsymbol{\mu}_{\mathbf{KNN}}$	μ^1	μ^2	μ^3	μ^4	$\boldsymbol{\mu}_{\mathbf{DT}}$	μ^1	μ^2	μ^3	μ^4
$\boldsymbol{\mu}_{\mathbf{KNN},1}$	16.6	11.0	60.81	294.70	$\boldsymbol{\mu}_{\mathbf{DT},1}$	2.6	17.0	59.01	288.46
$\boldsymbol{\mu}_{\mathbf{KNN},2}$	199.0	583.0	68.21	290.36	$\boldsymbol{\mu}_{\mathbf{DT},2}$	196.2	591.0	73.01	296.36
$\boldsymbol{\mu}_{\mathbf{KNN},3}$	405.4	1089.0	32.10	296.00	$\boldsymbol{\mu}_{\mathbf{DT},3}$	405.0	1003.0	32.10	298.76
$\boldsymbol{\mu}_{\mathbf{KNN},4}$	500.6	1000	53.30	291.51	$\boldsymbol{\mu}_{\mathbf{DT},4}$	501.8	1059.0	41.90	294.44
$\boldsymbol{\mu}_{\mathbf{KNN},5}$	135.0	1567.0	48.25	297.50	$\boldsymbol{\mu}_{\mathbf{DT},5}$	132.3	1543.6	52.99	297.64
$\boldsymbol{\mu}_{\mathbf{KNN},6}$	133.8	1457.0	48.25	297.30	$\boldsymbol{\mu}_{\mathbf{DT},6}$	134.6	1535.0	45.85	302.50

Table 4: The estimated values of $\mu_{\mathbf{KNN},i}$ and $\mu_{\mathbf{DT},i}$ for $i = 1..6$ with 100 LHS samplings

$\boldsymbol{\mu}_{\mathbf{KNN}}$	μ^1	μ^2	μ^3	μ^4	$\boldsymbol{\mu}_{\mathbf{DT}}$	μ^1	μ^2	μ^3	μ^4
$\boldsymbol{\mu}_{\mathbf{KNN},1}$	0.0	17.7	58.30	287.40	$\boldsymbol{\mu}_{\mathbf{DT},1}$	13.4	4.5	56.64	285.93
$\boldsymbol{\mu}_{\mathbf{KNN},2}$	198.8	679.1	70.76	288.21	$\boldsymbol{\mu}_{\mathbf{DT},2}$	196.1	545.3	72.86	288.01
$\boldsymbol{\mu}_{\mathbf{KNN},3}$	400.7	998.7	31.65	290.13	$\boldsymbol{\mu}_{\mathbf{DT},3}$	394.4	917.5	28.77	297.34
$\boldsymbol{\mu}_{\mathbf{KNN},4}$	501.6	1035.3	51.07	295.52	$\boldsymbol{\mu}_{\mathbf{DT},4}$	502.7	1046.3	54.61	289.71
$\boldsymbol{\mu}_{\mathbf{KNN},5}$	128.9	1607.9	48.34	295.93	$\boldsymbol{\mu}_{\mathbf{DT},5}$	133.4	1523.7	41.74	305.07
$\boldsymbol{\mu}_{\mathbf{KNN},6}$	133.1	1506.1	49.24	292.63	$\boldsymbol{\mu}_{\mathbf{DT},6}$	129.9	1484.1	51.40	296.37

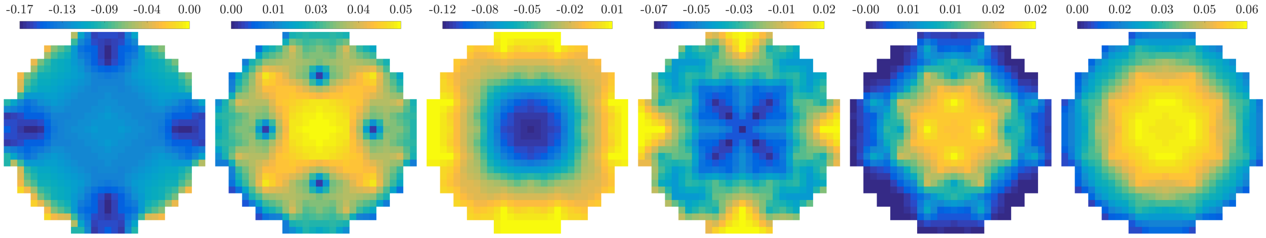
Table 5: The estimated values of $\boldsymbol{\mu}_{\mathbf{KNN},i}$ and $\boldsymbol{\mu}_{\mathbf{DT},i}$ for $i = 1..6$ with 1000 LHS samplings

$\boldsymbol{\mu}$	POD	KNN initial	KNN(100)	KNN(1000)	DT initial	DT(100)	DT(1000)
$\boldsymbol{\mu}_{\mathbf{true},1}$	0.47%	8.41%	1.43%	0.63%	9.98%	0.57%	0.57%
$\boldsymbol{\mu}_{\mathbf{true},2}$	0.64%	2.92%	1.30%	0.93%	2.14%	1.72%	0.64%
$\boldsymbol{\mu}_{\mathbf{true},3}$	0.74%	3.21%	1.70%	1.01%	5.41%	1.05%	0.72%
$\boldsymbol{\mu}_{\mathbf{true},4}$	0.62 %	3.74%	1.13%	0.72%	2.80%	1.52%	1.31%
$\boldsymbol{\mu}_{\mathbf{true},5}$	0.61%	2.73%	1.74%	0.91%	4.50%	1.12%	1.02%
$\boldsymbol{\mu}_{\mathbf{true},6}$	0.61%	2.71%	1.22%	0.85%	4.55%	2.13%	1.33%

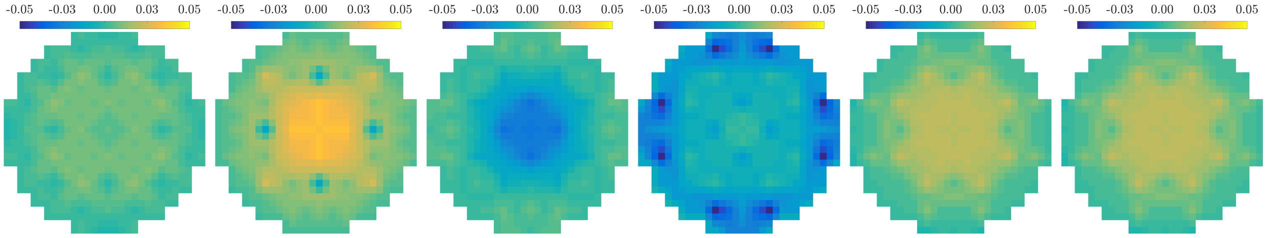
Table 6: Normalized reconstruction error in the field space for KNN and DT inverse models for six samplings of μ^i.

$\boldsymbol{\mu}$	KNN initial	KNN(100)	KNN(1000)	DT initial	DT(100)	DT(1000)
$\boldsymbol{\mu}_{\mathbf{true},1}$	8.52%	1.71%	0.33%	10.01%	0.31%	0.25%
$\boldsymbol{\mu}_{\mathbf{true},2}$	2.72%	1.31%	0.90%	1.83%	0.32%	0.32%
$\boldsymbol{\mu}_{\mathbf{true},3}$	3.11%	0.83%	0.72%	5.52%	0.42%	0.31%
$\boldsymbol{\mu}_{\mathbf{true},4}$	3.63 %	1.23%	0.73%	2.71%	0.81%	0.84%
$\boldsymbol{\mu}_{\mathbf{true},5}$	4.13%	1.71%	0.91%	2.51%	1.10%	1.01%
$\boldsymbol{\mu}_{\mathbf{true},6}$	2.64%	1.12%	0.71%	4.10%	0.83%	0.82%

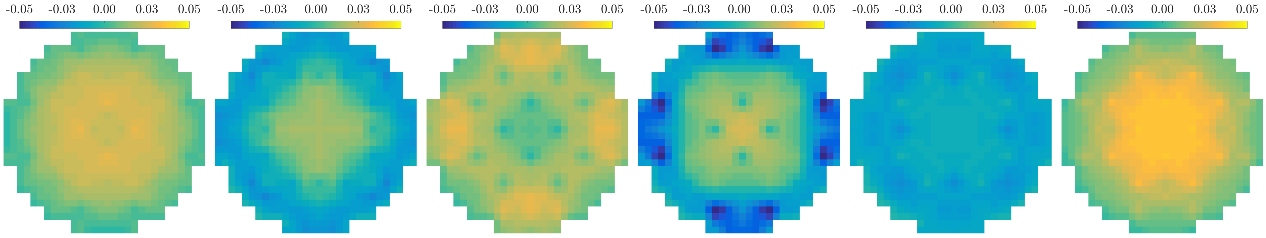
Table 7: Normalized error in the observation space for KNN and DT inverse models for six samplings of μ^i.



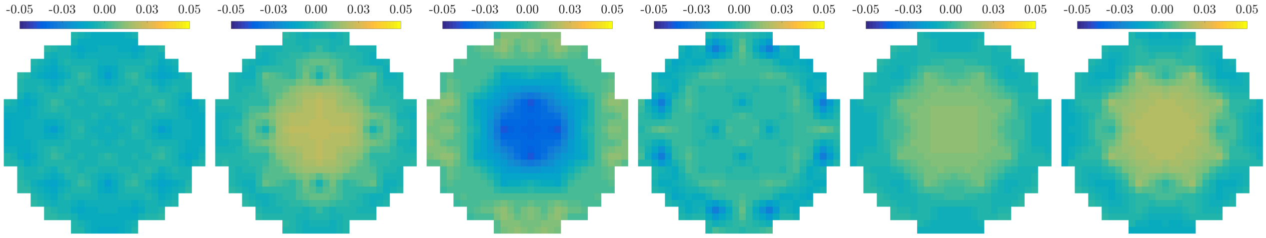
(a) err(b) err(c) err(d) err(e) err(f) err
$u^{\mathrm{KNN}}(\boldsymbol{\mu}_{\mathrm{initial},1})$ $u^{\mathrm{KNN}}(\boldsymbol{\mu}_{\mathrm{initial},2})$ $u^{\mathrm{KNN}}(\boldsymbol{\mu}_{\mathrm{initial},3})$ $u^{\mathrm{KNN}}(\boldsymbol{\mu}_{\mathrm{initial},4})$ $u^{\mathrm{KNN}}(\boldsymbol{\mu}_{\mathrm{initial},5})$ $u^{\mathrm{KNN}}(\boldsymbol{\mu}_{\mathrm{initial},6})$



(g) err(h) err(i) err(j) err(k) err(l) err
$u^{\mathrm{KNN}}(\boldsymbol{\mu}_{\mathrm{true},1})$ $u^{\mathrm{KNN}}(\boldsymbol{\mu}_{\mathrm{true},2})$ $u^{\mathrm{KNN}}(\boldsymbol{\mu}_{\mathrm{true},3})$ $u^{\mathrm{KNN}}(\boldsymbol{\mu}_{\mathrm{true},4})$ $u^{\mathrm{KNN}}(\boldsymbol{\mu}_{\mathrm{true},5})$ $u^{\mathrm{KNN}}(\boldsymbol{\mu}_{\mathrm{true},6})$

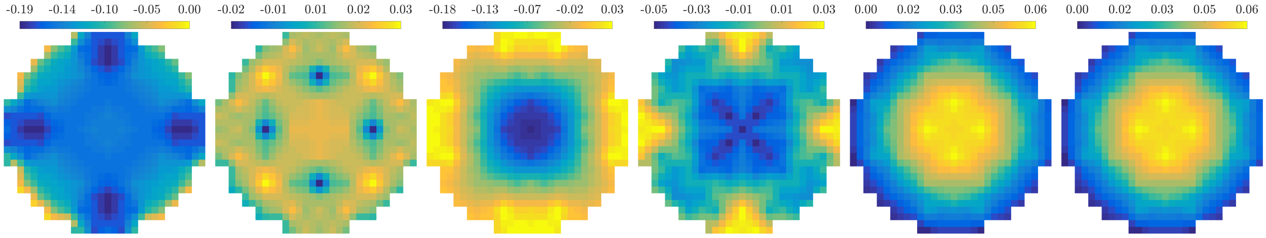


(m) err(n) err(o) err(p) err(q) err(r) err $u^{\mathrm{KNN}}(\boldsymbol{\mu}_{100,6})$
$u^{\mathrm{KNN}}(\boldsymbol{\mu}_{100,1})$ $u^{\mathrm{KNN}}(\boldsymbol{\mu}_{100,2})$ $u^{\mathrm{KNN}}(\boldsymbol{\mu}_{100,3})$ $u^{\mathrm{KNN}}(\boldsymbol{\mu}_{100,4})$ $u^{\mathrm{KNN}}(\boldsymbol{\mu}_{100,5})$

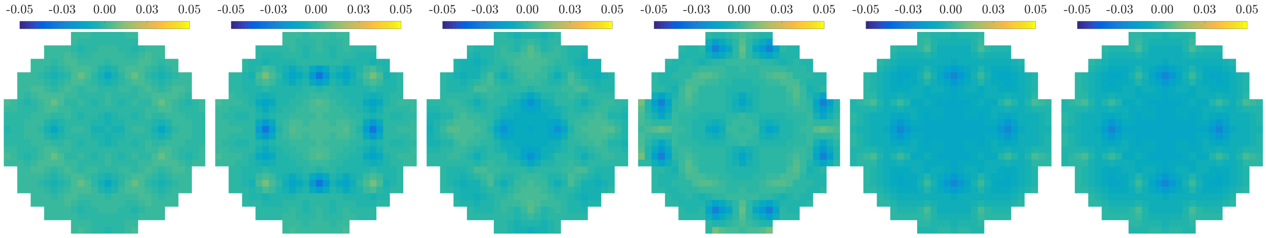


(s) err(t) err(u) err(v) err(w) err(x) err
$u^{\mathrm{KNN}}(\boldsymbol{\mu}_{1000,1})$ $u^{\mathrm{KNN}}(\boldsymbol{\mu}_{1000,2})$ $u^{\mathrm{KNN}}(\boldsymbol{\mu}_{1000,3})$ $u^{\mathrm{KNN}}(\boldsymbol{\mu}_{1000,4})$ $u^{\mathrm{KNN}}(\boldsymbol{\mu}_{1000,5})$ $u^{\mathrm{KNN}}(\boldsymbol{\mu}_{1000,6})$

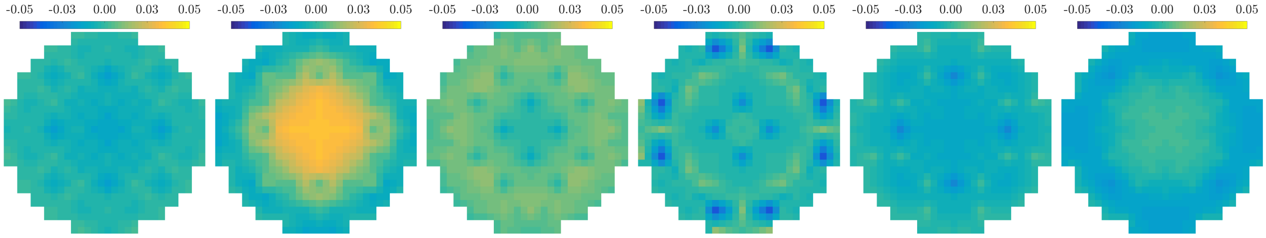
Figure 9: Comparison of KNN machine learning predictions against true values for reordered samples in the test set.



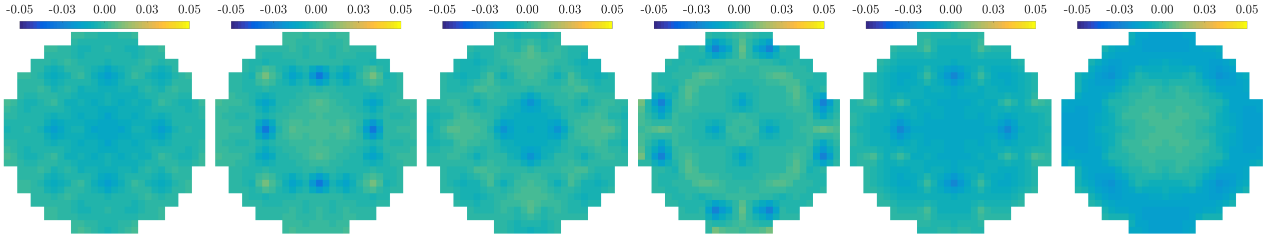
(a) err(b) err(c) err(d) err(e) err(f) err
$u^{\mathrm{DT}}(\boldsymbol{\mu}_{\text{initial},1})$ $u^{\mathrm{DT}}(\boldsymbol{\mu}_{\text{initial},2})$ $u^{\mathrm{DT}}(\boldsymbol{\mu}_{\text{initial},3})$ $u^{\mathrm{DT}}(\boldsymbol{\mu}_{\text{initial},4})$ $u^{\mathrm{DT}}(\boldsymbol{\mu}_{\text{initial},5})$ $u^{\mathrm{DT}}(\boldsymbol{\mu}_{\text{initial},6})$



(g) err $u^{\mathrm{DT}}(\boldsymbol{\mu}_{\text{true},1})$ (h) err $u^{\mathrm{DT}}(\boldsymbol{\mu}_{\text{true},2})$ (i) err $u^{\mathrm{DT}}(\boldsymbol{\mu}_{\text{true},3})$ (j) err $u^{\mathrm{DT}}(\boldsymbol{\mu}_{\text{true},4})$ (k) err $u^{\mathrm{DT}}(\boldsymbol{\mu}_{\text{true},5})$ (l) err $u^{\mathrm{DT}}(\boldsymbol{\mu}_{\text{true},6})$



(m) err $u^{\mathrm{DT}}(\boldsymbol{\mu}_{100,1})$ (n) err $u^{\mathrm{DT}}(\boldsymbol{\mu}_{100,2})$ (o) err $u^{\mathrm{DT}}(\boldsymbol{\mu}_{100,3})$ (p) err $u^{\mathrm{DT}}(\boldsymbol{\mu}_{100,4})$ (q) err $u^{\mathrm{DT}}(\boldsymbol{\mu}_{100,5})$ (r) err $u^{\mathrm{DT}}(\boldsymbol{\mu}_{100,6})$



(s) err $u^{\mathrm{DT}}(\boldsymbol{\mu}_{1000,1})$ (t) err $u^{\mathrm{DT}}(\boldsymbol{\mu}_{1000,2})$ (u) err $u^{\mathrm{DT}}(\boldsymbol{\mu}_{1000,3})$ (v) err $u^{\mathrm{DT}}(\boldsymbol{\mu}_{1000,4})$ (w) err(x) err $u^{\mathrm{DT}}(\boldsymbol{\mu}_{1000,6})$
$u^{\mathrm{DT}}(\boldsymbol{\mu}_{1000,5})$

Figure 10: Comparison of DT machine learning predictions against true values for reordered samples in the test set.

Despite the improvement, we notice that the $\boldsymbol{\mu}$ value on the output (obtained from Eq. 3.5) exhibits some random behavior. In fact, both KNN and DT forward functions can be highly non-convex, thus avoiding local minimums when applying the inverse strategies can be hurdle. For this reason, instead of using one estimated vector $\boldsymbol{\mu}^*$, we envisage averaging an ensemble of parameters $\mathcal{U} = \{\boldsymbol{\mu}_1^*, \boldsymbol{\mu}_2^*, \boldsymbol{\mu}_3^*, \boldsymbol{\mu}_4^*, \boldsymbol{\mu}_5^*\}$. More precisely,

$$\boldsymbol{\mu}_1^* = \boldsymbol{\mu}^* \quad \text{and} \quad \boldsymbol{\mu}_i^* = \underset{k=\{1..n^s\}/\{0..i-1\}}{\operatorname{argmin}} (||HV_n F^{\mathrm{ML}}(\boldsymbol{\mu}_k) - Y_{\mathrm{o}}||_2), \quad \text{for} \quad i > 1. \tag{4.14}$$

This process is shown in details in Algorithm 15. The averaged vector $\bar{\boldsymbol{\mu}} = (\sum_{i=1}^{5} \boldsymbol{\mu}_i^*)/5$ using 1000 LHS samples is shown in Table 8. Compared to previous results presented in Table 5, little difference is observed concerning the output of KNN. On the other hand, the average DT output of the inverse modelling $\bar{\boldsymbol{\mu}}$ owns a smaller estimation error in comparison with $\boldsymbol{\mu}^*$, especially for parameter μ^2 of $\boldsymbol{\mu}$.

Algorithm 2 Inverse modelling based on machine learning forward functions

```
Inputs:
Observations: $Y_o$
Initial guess: $\boldsymbol{\mu}_{\text{initial}}$
Forward function: $F^{\text{ML}}$
Transformation operator: $H$
Sampling number: $n^s$

$\mathcal{U}$ = LHS (number = $n^s$, initial = $\boldsymbol{\mu}_{\text{initial}}$)
$\boldsymbol{\mu}_1^* = \underset{k=1..n^s}{\text{argmin}}(||HV_n F^{\text{ML}}(\boldsymbol{\mu}^k) - Y_{\text{o}}||_2)$
for $i$ from 2 to 5: do
    $\boldsymbol{\mu}_i^* = \underset{k=\{1..n^s\}/\{0..i-1\}}{\text{argmin}}(||HV_n F^{\text{ML}}(\boldsymbol{\mu}^k) - Y_{\text{o}}||_2)$
end for
$\bar{\boldsymbol{\mu}} = (\sum_{i=1}^{5} \boldsymbol{\mu}_i^*)/5$

Outputs:
Parameter: $\boldsymbol{\mu}^*$
Reconstructed field: $V_n F^{\text{ML}}(\boldsymbol{\mu}^*)$
```

$\boldsymbol{\mu}_{\bar{\mathbf{KNN}}}$	μ^1	μ^2	μ^3	μ^4	$\boldsymbol{\mu}_{\bar{\mathbf{DT}}}$	μ^1	μ^2	μ^3	μ^4
$\boldsymbol{\mu}_{\bar{\mathbf{KNN}},1}$	6.2	12.2	59.51	289.79	$\boldsymbol{\mu}_{\bar{\mathbf{DT}},1}$	12.8	18.5	58.57	290.56
$\boldsymbol{\mu}_{\bar{\mathbf{KNN}},2}$	197.8	626.5	71.17	289.80	$\boldsymbol{\mu}_{\bar{\mathbf{DT}},2}$	197.6	580.0	73.66	287.61
$\boldsymbol{\mu}_{\bar{\mathbf{KNN}},3}$	399.8	978.1	31.33	295.27	$\boldsymbol{\mu}_{\bar{\mathbf{DT}},3}$	399.4	1020.7	27.33	300.68
$\boldsymbol{\mu}_{\bar{\mathbf{KNN}},4}$	500.4	960.1	48.18	292.34	$\boldsymbol{\mu}_{\bar{\mathbf{DT}},4}$	502.1	1023.2	51.36	298.68
$\boldsymbol{\mu}_{\bar{\mathbf{KNN}},5}$	134.9	1694.9	47.57	301.23	$\boldsymbol{\mu}_{\bar{\mathbf{DT}},5}$	129.2	1678.7	48.99	291.28
$\boldsymbol{\mu}_{\bar{\mathbf{KNN}},6}$	132.9	1465.5	46.40	300.32	$\boldsymbol{\mu}_{\bar{\mathbf{DT}},6}$	129.9	1487.8	54.53	291.01

Table 8: The averaged values of $\boldsymbol{\mu}_{\mathbf{KNN},i}$ and $\boldsymbol{\mu}_{\mathbf{DT},i}$ for $i = 1..6$ with 1000 LHS samplings

The averaging of $\boldsymbol{\mu}_i^*$ is helpful to reduce the randomness of the samplings, thus enhancing the robustness of the inverse approach as shown in Table 8. However, for highly non-convex functions, when the sampling number is insufficient to cover the domain of definition, $\boldsymbol{\mu}_i^*$ can be included in the neighborhoods of different local minimums as illustrated in Fig 11. We propose, in this paper, a two-steps advanced sampling strategy to improve the accuracy of the inverse modelling where the first step consists of discovering 5 optimal samplings $\boldsymbol{\mu}_i^*$ regarding the reconstruction in the observation space and 50 more samplings are generated around each $\boldsymbol{\mu}_i^*$ with a finer range $[\mu_i^{1*} \pm 2, \mu_i^{2*} \pm 10, \mu_i^{3*} \pm 3, \mu_i^{4*} \pm 1]$. The output of the algorithm $\boldsymbol{\mu}_{\hat{\mathbf{KNN}},i}$ or $\mu_{\hat{\mathbf{DT}},i}$ is thus determined as the set of parameters, among all 250 samplings of the second step, for which the reconstruction error is the smallest in the observation space. The procedure of this two-steps sampling strategy is illustrated in Fig 11 and in Algorithm 19 where the initial 750 samplings are symbolized by blue points and the second-step samplings (250 in total) around $\boldsymbol{\mu}_i^*$ are represented by red points.

Therefore, this inverse resolution with two-steps sampling owns the same complexity as the ones presented in Table 5 and Table 8. The output of this two-steps sampling strategy $\boldsymbol{\mu}_{\mathbf{K\hat{N}N},i}$ is shown in Table 9. We observe a more robust inverse estimation, especially for the DT method. This result is consistent with the high non-convexity of the KNN regression function [105].

Algorithm 3 Two-steps inverse modelling

```
Inputs:
Observations: $Y_o$
Initial guess: $\boldsymbol{\mu}_{\text{initial}}$
Forward function: $F^{\text{ML}}$
Transformation operator: $H$
First stage sampling number: $n^{s,1}$
Second stage sampling number: $n^{s,2}$

$\mathcal{U}^1$ = LHS (number = $n^{s,1}$, initial = $\boldsymbol{\mu}_{\text{initial}}$)
$\boldsymbol{\mu}_1^* = \underset{k=1..n^s}{\text{argmin}}(||HV_nF^{\text{ML}}(\boldsymbol{\mu}^k) - Y_{\text{o}}||_2)$
for $i$ from 2 to 5: do
    $\boldsymbol{\mu}_i^* = \underset{k=\{1..n^s\}/\{0..i-1\}}{\text{argmin}} (||HV_nF^{\text{ML}}(\boldsymbol{\mu}_k) - Y_{\text{o}}||_2)$
end for
for $i$ from 1 to 5: do
    $\mathcal{U}_i^2$ = LHS (number = $n^{s,2}$, initial = $\boldsymbol{\mu}_i^*$)
end for
$\boldsymbol{\mu}^* = \underset{\boldsymbol{\mu} \in \bigcup \mathcal{U}_i^2}{\text{argmin}}(||HV_nF^{\text{ML}}(\boldsymbol{\mu}) - Y_{\text{o}}||_2)$

Outputs:
Parameter: $\boldsymbol{\mu}^*$
Reconstructed field: $V_nF^{\text{ML}}(\boldsymbol{\mu}^*)$
```

$\boldsymbol{\mu}_{\mathbf{K\hat{N}N}}$	μ^1	μ^2	μ^3	μ^4	$\boldsymbol{\mu}_{\mathbf{\hat{D}T}}$	μ^1	μ^2	μ^3	μ^4
$\boldsymbol{\mu}_{\mathbf{K\hat{N}N},1}$	2.52	1.40	58.97	285.55	$\boldsymbol{\mu}_{\mathbf{\hat{D}T},1}$	1.65	16.40	53.43	303.03
$\boldsymbol{\mu}_{\mathbf{K\hat{N}N},2}$	200.92	524.06	67.69	287.77	$\boldsymbol{\mu}_{\mathbf{\hat{D}T},2}$	197.49	551.46	76.27	286.10
$\boldsymbol{\mu}_{\mathbf{K\hat{N}N},3}$	398.01	1025.67	27.14	303.18	$\boldsymbol{\mu}_{\mathbf{\hat{D}T},3}$	403.95	928.40	31.68	285.50
$\boldsymbol{\mu}_{\mathbf{K\hat{N}N},4}$	500.60	924.60	55.54	288.28	$\boldsymbol{\mu}_{\mathbf{\hat{D}T},4}$	501.09	969.60	50.24	295.46
$\boldsymbol{\mu}_{\mathbf{K\hat{N}N},5}$	133.37	1587.40	50.01	291.50	$\boldsymbol{\mu}_{\mathbf{\hat{D}T},5}$	132.53	1527.20	41.63	299.58
$\boldsymbol{\mu}_{\mathbf{K\hat{N}N},6}$	129.85	1567.93	50.29	296.65	$\boldsymbol{\mu}_{\mathbf{\hat{D}T},6}$	130.00	1480.00	52.15	300.00

Table 9: The output of the two-steps sampling strategy $\boldsymbol{\mu}_{\mathbf{K\hat{N}N}}$ and $\boldsymbol{\mu}_{\mathbf{\hat{D}T}}$ with 1000 LHS samplings in total

4.4.2 Inverse problem for $\{\mu^1, \mu^3\}$ with a small number of samplings

In order to better illustrate the performance of different inverse approaches proposed in this article, we compare three more study cases as shown in table. 10. Since the observations are most sensitive to μ^1 and μ^3 as illustrated in Fig. 8, μ^2 and μ^4 are supposed to be known and kept invariant in

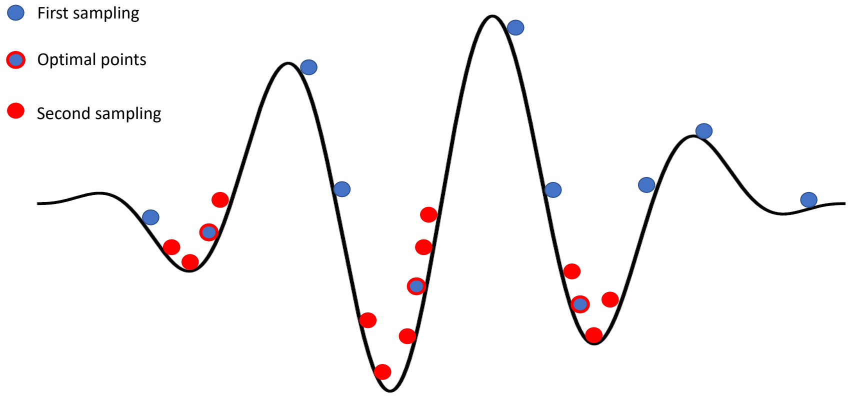


Figure 11: Illustration of advanced two-steps sampling strategy

these tests, i.e, $(\mu^2_{\text{initial}}, \mu^4_{\text{initial}}) \equiv (\mu^2_{\text{true}}, \mu^4_{\text{true}})$. For each of these methods, including the standard (Eq. 4.13), the averaging (Eq. 4.14) and the two-step (Algorithm 3) approaches, 150 LHS samples, within the range $[\mu^{1*}_i \pm 100, \mu^{3*}_i \pm 30]$, have been generated to solve the inverse problem. For the two-step method, 100 samples are used to find $\boldsymbol{\mu}^*_i, i = 1..5$, followed by 10 samples around each $\boldsymbol{\mu}^*_i$ within $[\mu^{1*}_i \pm 10, \mu^{3*}_i \pm 3]$. The ouput of these algorithms is shown in Table. 12 where a significant advantage of averaging and two-steps algorithms, compared to the standard approach, can be observed in general. The reconstruction error is shown in Table. 11. We display the distribution of generated samples in Fig. 12. For the comparison of standard and averaging approaches, we observe in (a),(b),(e),(f) that for both KNN and DT methods, the green points which are the output of the averaging algorithms, are in general closer to the true value of (μ^1, μ^3) compared to the purple points (output of the standard methods). On the other hand, for the two-steps method, the green points (output) are also generally closer to the true values compared to the ensemble of blue points. These results confirm the strength of averaging and two-steps methods in the inverse modelling, especially with a small budget of samplings. The ensemble of these numerical results (including the ones in section 4.4.1) demonsrtates also the robustness of the proposed approaches for problems of different dimensions.

$\boldsymbol{\mu}_{\textbf{true}}$	μ^1	μ^2	μ^3	μ^4	$\boldsymbol{\mu}'_{\textbf{initial}}$	μ^1	μ^2	μ^3	μ^4
$\boldsymbol{\mu}_{\textbf{true},4}$	500	1000	52.96	291.51	$\boldsymbol{\mu}'_{\textbf{initial},4}$	560	1000	30.00	291.54
$\boldsymbol{\mu}_{\textbf{true},5}$	130	1500	52.15	291.01	$\boldsymbol{\mu}'_{\textbf{initial},5}$	80	1500	72.15	291.01

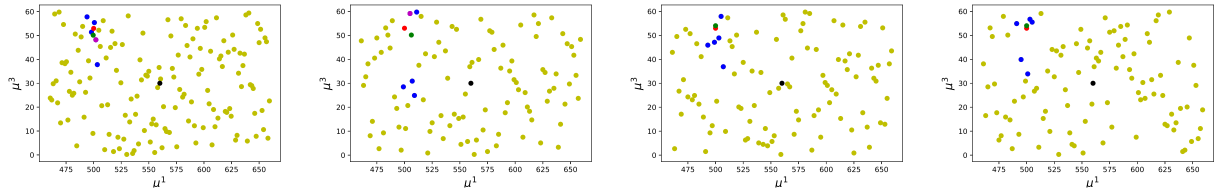
Table 10: The values of $\boldsymbol{\mu}_{\textbf{true},i}$ and $\boldsymbol{\mu}'_{\textbf{initial},i}$ for $i = 4, 5$.

$\boldsymbol{\mu}$	KNN (initial)	KNN (standard)	KNN (averaging)	KNN (two-steps)	DT (initial)	DT (standard)	DT (averaging)	DT (two-steps)
$\boldsymbol{\mu}_{\mathbf{true},4}$	24.4 %	0.9%	0.7%	0.9%	24.3%	3.8%	2.5%	4.7%
$\boldsymbol{\mu}_{\mathbf{true},5}$	8.0%	1.8%	1.3%	1.6%	8.3%	5.9%	3.0%	5.6%

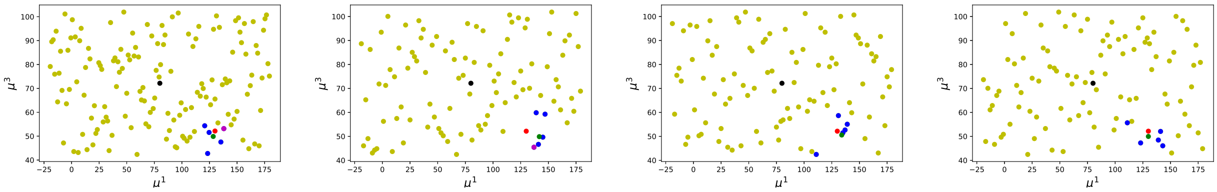
Table 11: Normalized reconstruction error for KNN and DT inverse models using different approaches.

$\boldsymbol{\mu}_{\mathbf{KNN}}$	μ^1	μ^2	μ^3	μ^4	$\boldsymbol{\mu}_{\mathbf{DT}}$	μ^1	μ^2	μ^3	μ^4
$\boldsymbol{\mu}_{\mathbf{KNN},4}$(standard)	502.0	1000	48.23	291.51	$\boldsymbol{\mu}_{\mathbf{DT},4}$(standard)	505.0	1000	59.10	291.51
$\boldsymbol{\mu}_{\mathbf{KNN},4}$(averaging)	499.6	1000	50.12	291.51	$\boldsymbol{\mu}_{\mathbf{DT},4}$(averaging)	506.2	1000	49.62	291.51
$\boldsymbol{\mu}_{\mathbf{KNN},4}$(two-steps)	500.1	1000	48.60	291.51	$\boldsymbol{\mu}_{\mathbf{DT},4}$(two-steps)	500.0	1000	54.00	291.51
$\boldsymbol{\mu}_{\mathbf{KNN},5}$(standard)	138.1	1500	53.15	291.01	$\boldsymbol{\mu}_{\mathbf{DT},5}$(standard)	137.2	1500	45.45	291.01
$\boldsymbol{\mu}_{\mathbf{KNN},5}$(averaging)	128.4	1500	49.87	291.01	$\boldsymbol{\mu}_{\mathbf{DT},5}$(averaging)	141.8	1500	52.17	291.01
$\boldsymbol{\mu}_{\mathbf{KNN},5}$(two-steps)	134.0	1500	50.55	291.01	$\boldsymbol{\mu}_{\mathbf{DT},5}$(two-steps)	130.0	1500	49.95	291.01

Table 12: The estimated values of $\mu_{\mathbf{KNN},i}$ and $\mu_{\mathbf{DT},i}$ for $i = 4, 5$ with 150 LHS samplings in total of three different strategies.



(a) $\boldsymbol{\mu}_{\mathrm{KNN},4}$ (b) $\boldsymbol{\mu}_{\mathrm{DT},4}$ (c) $\boldsymbol{\mu}_{\mathrm{KNN},4}$ (2-steps) (d) $\boldsymbol{\mu}_{\mathrm{DT},4}$ (2-steps)



(e) $\boldsymbol{\mu}_{\mathrm{KNN},5}$ (f) $\boldsymbol{\mu}_{\mathrm{DT},5}$ (g) $\boldsymbol{\mu}_{\mathrm{KNN},5}$ (2-steps) (h) $\boldsymbol{\mu}_{\mathrm{DT},5}$ (2-steps)

Figure 12: Distribution of generated samplings through different approaches. The black points are the initial guesses of μ^1 and μ^3 while the red points stand for the true value. The blue points represent the five optimal samplings, evaluated in the observation space. The green points represent the output of the algorithms. In (a),(b),(e),(f), the purple points stand for the optimal sampling without averaging.

5 Conclusions

In this paper, a data enabled physics-informed digital twin is proposed to predict the neutronic field and the operational parameters of a nuclear reactor core. The digital twin is designed to solve forward problems given input parameters, as well as to solve inverse problems given some extra measurements. Our main contribution is to build a non-intrusive reduced order model with machine learning methods such as KNN and DT, which is fit for limited data modelling approaches for our nuclear engineering applications. Though other machine learning methods such ANN, GP are also tested in this work, we didn't present any related results because of non-satisfactory compared to KNN and DT.

The data enabled physics-informed digital twin is tested through a real engineering problem in nuclear reactor physics - reactor core simulation in the life cycle of HPR1000. Numerical results proved the accuracy of the forward model, though a little worse than POD method its self, it is still acceptable from the engineering point view. The inverse model for the digital twin is totally new in nuclear reactor physics domain. Benefits from the rapid response of the machine learning reduced order forward model, which is far less than solving the forward problem using a neutronic simulation code directly, the digital twin is able to evaluate the forward model many times by sampling the parameter around the initial guess, the optimal parameter which match the observations better can be selected. Numerical results also proved that, with the advanced sampling strategy, the determination of the optimal parameter and the related output neutronic field is more robust.

This work is valuable because of its simple theory, easy implementable and satisfactory accuracy and time cost for the prediction of input parameters and output physical fields. Thus it successfully supports the implementation of real engineering scale digital twin of reactor physics at NPIC.

The framework of the proposed data enabled physics-informed digital twin is general, which can be also adapted to other industry domain. The next work is to study the robustness of the digital twin with respect to the noisy parameter and noisy observations, and an industrial scale code shall be developed for real applications of the proposed digital twin. Future work can be considered to improve the inverse modelling of the proposed digital twin, for instance, using data assimilation or Bayesian framework. At the current stage, the main difficulty is the non-differentiability of the machine learning forward functions.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant No. 11905216). This work benefited from helpful supervision with Yvon Maday, professor at Laboratoire Jacques-Louis Lions, Sorbonne Universités, UPMC Univ Paris 06, 4, Place Jussieu, 75005 Paris, France. This work also benefited from helpful discussions with Chuanju Xu, School of the Mathematical Modeling and High Performance Scientific Computing, Xiamen University, 361005 Xiamen, China.The authors are grateful to two anonymous reviewers for the useful remarks on the manuscript.

Code availability

Code for the implementation of CNN and DT forward prediction with hyperparameters tuning and inverse modelling is available at
https://github.com/scheng1992/ML-for-predicting-POD-coefficients

References

[1] A. Hebert. *Applied Reactor Physics.* Presses inter Polytechnique, 2009.

[2] S. Marguet. *La physique des réacteurs nucléaires.* Lavoisier, 2011.

[3] M.G. Kapteyn, D.J. Knezevic, D.B.P. Huynh, M. Tran, and K.E. Willcox. Data-driven physics-based digital twins via a library of component-based reduced-order models. *International Journal for Numerical Methods in Engineering*, pages 1–18, 2020.

[4] J. Kraft and S. Kuntzagk. Engine Fleet-Management: The Use of Digital Twins From a MRO Perspective. *ASME Turbo Expo 2017: Turbomachinery Technical Conference and Exposition, American Society of Mechanical Engineers*, 2017.

[5] Francisco Chinesta, Elias Cueto, Emmanuelle Abisset-Chavanne, Jean Louis Duval, and Fouad El Khaldi. Virtual, Digital and Hybrid Twins: A New Paradigm in Data-Based Engineering and Engineered Data. *Archives of Computational Methods in Engineering*, pages 1–30, 2018.

[6] Michael G. Kapteyn, David J. Knezevic, and Karen Willcox. *Toward predictive digital twins via component-based reduced-order models and interpretable machine learning.* 2020.

[7] Paul J Turinsky and Douglas B Kothe. Modeling and simulation challenges pursued by the Consortium for Advanced Simulation of Light Water Reactors (CASL). *Journal of Computational Physics*, 313:367–376, 2016.

[8] F Franceschini, AT Godfrey, and JC Gehin. AP1000 PWR Reactor Physics Analysis with VERA-CS and KENO-VI–Part I: Zero Power Physics Tests. In *Proceedings of PHYSOR 2014: The Role of Reactor Physics Toward a Sustainable Future*, 2014.

[9] Y. Maday. Reduced basis method for the rapid and reliable solution of partial differential equations. In *in International Congress of Mathematicians. Vol. III, 1255–1270, Eur. Math. Soc., Zürich.* Citeseer, 2006.

[10] A. Quarteroni, G. Rozza, and A. Manzoni. Certified reduced basis approximation for parametrized partial differential equations and applications. *Journal of Mathematics in Industry*, 1(1):1–49, 2011.

[11] Lawrence Sirovich. Turbulence and the dynamics of coherent structures. II. Symmetries and transformations. *Quarterly of Applied mathematics*, 45(3):573–582, 1987.

[12] K. Kunisch and S. Volkwein. Galerkin proper orthogonal decomposition methods for parabolic problems. *Numerische mathematik*, 90(1):117–148, 2001.

[13] P. Ladevèze and L. Chamoin. On the verification of model reduction methods based on the proper generalized decomposition. *Computer Methods in Applied Mechanics and Engineering*, 200(23):2032–2047, 2011.

[14] Sibo Cheng, Didier Lucor, and Jean-Philippe Argaud. Observation data compression for variational assimilation of dynamical systems. *Journal of Computational Science*, page 101405, 2021.

[15] M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera. An ‘empirical interpolation’ method: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathematique*, 339(9):667–672, 2004.

[16] Y. Maday and O. Mula. A generalized empirical interpolation method: application of reduced basis techniques to data assimilation. In *Analysis and numerics of partial differential equations*, pages 221–235. Springer, 2013.

[17] P. Benner, M. Ohlberger, A. Cohen, and K. Willcox. *Model Reduction and Approximation: Theory and Algorithms*, volume 15. SIAM, 2017.

[18] J. S. Hesthaven, G. Rozza, B. Stamm, et al. *Certified reduced basis methods for parametrized partial differential equations*. Springer, 2016.

[19] G. Rozza, D. B. P. Huynh, and A. T. Patera. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. *Archives of Computational Methods in Engineering*, 15(3):229, 2008.

[20] C. W. Rowley, T. Colonius, and R. M. Murray. Model reduction for compressible flows using pod and Galerkin projection. *Physica D: Nonlinear Phenomena*, 189(1-2):115–129, 2004.

[21] Qian Wang, Nicolò Ripamonti, and Jan S.Hesthaven. Recurrent neural network closure of parametric POD-Galerkin reduced-order models based on the Mori-Zwanzig formalism. 410:109402, 2020.

[22] A. Iollo, S. Lanteri, and J.-A. Désidéri. Stability Properties of POD-Galerkin Approximations for the Compressible Navier-Stokes Equations. *Theoretical & Computational Fluid Dynamics*, 13(6):377–396, 2000.

[23] P. An, Y. Ma, P. Xiao, F. Guo, W. Lu, and X. Chai. Development and validation of reactor nuclear design code CORCA-3D. *Nuclear Engineering and Technology*, 51:1721–1728, 2019.

[24] A. Calloo, D. Couyras, F. Févotte, and M. Guillo. Cocagne: EDF new neutronic core code for ANDROMEDE calculation chain. In *Proceedings of International Conference on Mathematics & Computational Methods Applied to Nuclear Science & Engineering (M&C), Jeju, Korea*, 2017.

[25] H Ly and H Tran. Modeling and control of physical processes using proper orthog- onal decomposition. *J Math Comput Model*, 33:223–236, 2001.

[26] D. Xiao, F. Fang, C.C. Pain, and I.M. Navon. A parameterized non-intrusive reduced order model and error analysis for general time-dependent nonlinear partial differential equations and its applications. *Computer Methods in Applied Mechanics and Engineering*, 317:868–889, 2017.

[27] D Xiao, F Fang, C Pain, and G Hu. Non-intrusive reduced-order modelling of the Navier–Stokes equations based on RBF interpolation. *International Journal for Numerical Methods in Fluids*, 79(11):580–595, 2015.

[28] C. Audouze, F. De Vuyst, and P. B. Nair. Reduced-order modeling of parameterized PDEs using time-space-parameter principal component analysis. *International Journal for Numerical Methods in Engineering*, 80(8), 2009.

[29] Christophe Audouze, Florian De Vuyst, and Prasanth B. Nair. Nonintrusive reduced-order modeling of parametrized time-dependent partial differential equations. *Numerical Methods for Partial Differential Equations*, 29(5):1587–1628, 2013.

[30] Laura Mainini and Karen E. Willcox. Surrogate modeling approach to support real-time structural assessment and decision making. *AIAA J*, 53(6):1612–1626, 2015.

[31] Benjamin Peherstorfer and Karen Willcox. Data-driven operator inference for nonintrusive projection-based model reduction. *Computer Methods in Applied Mechanics and Engineering*, 306:196–215, 2016.

[32] Peter Benner, Pawan Goyal, Boris Kramer, Benjamin Peherstorfer, and Karen Willcox. Operator inference for non-intrusive model reduction of systems with non-polynomial nonlinear terms, 2020.

[33] R. Chakir, Y. Maday, and P. Parnaudeau. A non-intrusive reduced basis approach for parametrized heat transfer problems. *Journal of Computational Physics*, 376:617 – 633, 2019.

[34] J.S. Hesthaven and S. Ubbiali. Non-intrusive reduced order modeling of nonlinear problems using neural networks. *Journal of Computational Physics*, 363:55 – 78, 2018.

[35] Hugo FS Lui and William R Wolf. Construction of reduced-order models for fluid flows using deep feedforward neural networks. *Journal of Fluid Mechanics*, 872:963–994, 2019.

[36] Qian Wang, Jan S. Hesthaven, and Deep Ray. Non-intrusive reduced order modeling of unsteady flows using artificial neural networks with application to a combustion problem. 384:289 – 307, 2019.

[37] Chen Wenqian, Qian Wangand Jan S. Hesthaven, and Chuhua Zhang. Physics-informed machine learning for reduced-order modeling of nonlinear problems. *Preprint submitted to Journal of Computational Physics*, 2020.

[38] Didier Lucor, Atul Agrawal, and Anne Sergent. Physics-aware deep neural networks for surrogate modeling of turbulent natural convection, 2021.

[39] Shaowu Pan and Karthik Duraisamy. Physics-Informed Probabilistic Learning of Linear Embeddings of Non-linear Dynamics With Guaranteed Stability. *arXiv preprint arXiv:1906.03663*, 2019.

[40] Luning Sun, Han Gao, Shaowu Pan, and Jian-Xun Wang. Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data. *Computer Methods in Applied Mechanics and Engineering*, page 112732, 2019.

[41] 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.

[42] Renee Swischuk, Boris Kramer, Cheng Huang, and Karen Willcox. Learning physics-based reduced-order models for a single-injector combustion process. *arXiv preprint arXiv:1908.03620*, 2019.

[43] Kody Law, Andrew Stuart, and Konstantinos Zygalakis. *Data assimilation: a mathematical introduction*, volume 62. Springer, 2015.

[44] Mark Asch, Marc Bocquet, and Maëlle Nodet. *Data assimilation: methods, algorithms, and applications*. SIAM, 2016.

[45] Gong Helin. *Data assimilation with reduced basis and noisy measurement: Applications to nuclear reactor cores.* PhD thesis, Sorbonne University, 2018.

[46] Y. Maday, A. T. Patera, J. D. Penn, and M. 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(5):933–965, 2015.

[47] 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(2018):354–370, 2018.

[48] Tommaso Taddei and Anthony T Patera. A localization strategy for data assimilation; application to state estimation and parameter estimation. *SIAM Journal on Scientific Computing*, 40(2):B611–B636, 2018.

[49] Albert Cohen, Wolfgang Dahmen, and Ron DeVore. State Estimation – The Role of Reduced Models, 2020.

[50] F. Wols, D. Lathouwers, and W. Uyttenhove. Transient analyses of accelerator driven systems using modal expansion techniques. Master's thesis, Delft University of Technology, 2010.

[51] A. G. Buchan, C. C. Pain, F. Fang, and I. M. Navon. A POD reduced-order model for eigenvalue problems with application to reactor physics. *International Journal for Numerical Methods in Engineering*, 95(12):1011–1032, 2013.

[52] A. G. Buchan, A. A. Calloo, M. G Goffin, S. Dargaville, F. Fang, C. Pain, and I. M. Navon. A POD reduced order model for resolving angular direction in neutron/photon transport problems. *Journal of Computational Physics*, 296:138–157, 2015.

[53] C. E. Heaney, A. G. Buchan, C. C. Pain, and S. Jewer. A reduced order model for criticality problems in reactor physics varying control rod settings. In *Proceedings of the 24th UK Conference of the Association for Computational Mechanics in Engineering, Cardiff University, Cardiff*, 2016.

[54] S. Lorenzi, A. Cammi, and L. Luzzi. Spatial neutronics modelling to evaluate the temperature reactivity feedbacks in a lead-cooled fast reactor. *Proceedings of ICAPP 2015, Nice, France, May 03-06*, 2015.

[55] A. Sartori, D. Baroli, A. Cammi, D. Chiesa, L. Luzzi, R. Ponciroli, E. Previtali, M. E. Ricotti, G. Rozza, and M. Sisti. Comparison of a Modal Method and a Proper Orthogonal Decomposition approach for multi-group time-dependent reactor spatial kinetics. *Annals of Nuclear Energy*, 71:217–229, 2014.

[56] A. Sartori. *Reduced order methods: applications to nuclear reactor core spatial dynamics.* PhD thesis, Italy, 2015.

[57] A. Sartori, A. Cammi, L. Luzzi, M. E. Ricotti, G. Rozza, et al. Reduced Order Methods: Applications to Nuclear Reactor Core Spatial Dynamics. In *ICAPP 2015*, pages 3318–3327. Omnipress, 2015.

[58] A. Cammi, L. Luzzi, and G. Rozza. A Reduced Basis Approach for Modeling the Movement of Nuclear Reactor Control Rods. *Journal of Nuclear Engineering and Radiation Science APRIL*, 2:021019–1, 2016.

[59] Péter German and Jean C Ragusa. Reduced-order modeling of parameterized multi-group diffusion k-eigenvalue problems. *Annals of Nuclear Energy*, 134:144–157, 2019.

[60] Zachary M Prince and Jean C Ragusa. Application of proper generalized decomposition to multigroup neutron diffusion eigenvalue calculations. *Progress in Nuclear Energy*, 121:103232, 2020.

[61] Péter German, Mauricio Tano, Jean C Ragusa, and Carlo Fiorina. Comparison of Reduced-Basis techniques for the model order reduction of parametric incompressible fluid flows. *Progress in Nuclear Energy*, 130:103551, 2020.

[62] JK Vaurio and C Mueller. Probabilistic analysis of Liquid-Metal Fast Breeder Reactor accident consequences with response surface techniques. *Nuclear Science and Engineering*, 65(2):401–413, 1978.

[63] Piero Baraldi, Francesca Mangili, and Enrico Zio. A prognostics approach to nuclear component degradation modeling based on Gaussian Process Regression. *Progress in Nuclear Energy*, 78:141–154, 2015.

[64] Antonio Figueroa and Malte Göttsche. Gaussian processes for surrogate modeling of discharged fuel nuclide compositions. *Annals of Nuclear Energy*, 156:108085, 2021.

[65] Anthony Michael Scopatz. Facility Deployment Decisions Through Warp Optimization of Regressed Gaussian Processes. *Nuclear Science and Engineering*, 186(1):83–97, 2017.

[66] Andreas Ikonomopoulos, Miltiadis Alamaniotis, Stylianos Chatzidakis, and Lefteri H Tsoukalas. Gaussian processes for state identification in pressurized water reactors. *Nuclear Technology*, 182(1):1–12, 2013.

[67] Xu Wu, Tomasz Kozlowski, Hadi Meidani, and Koroush Shirvan. Inverse uncertainty quantification using the modular Bayesian approach based on Gaussian process, Part 1: Theory. *Nuclear Engineering and Design*, 335:339–355, 2018.

[68] Xu Wu, Tomasz Kozlowski, Hadi Meidani, and Koroush Shirvan. Inverse uncertainty quantification using the modular Bayesian approach based on Gaussian Process, Part 2: Application to TRACE. *Nuclear Engineering and Design*, 335:417–431, 2018.

[69] Chen Wang, Xu Wu, and Tomasz Kozlowski. Gaussian process–based inverse uncertainty quantification for trace physical model parameters using steady-state psbt benchmark. *Nuclear Science and Engineering*, 193(1-2):100–114, 2019.

[70] L Gilli, D Lathouwers, JL Kloosterman, and THJJ Van Der Hagen. Development of an adaptive nonintrusive spectral technique for uncertainty quantification and application to a multiphysics problem. *Nuclear Science and Engineering*, 175(2):172–187, 2013.

[71] Yeni Li, Hany S Abdel-Khalik, Acacia J Brunett, Elise Jennings, Travis Mui, and Rui Hu. ROM-Based Surrogate Systems Modeling of EBR-II. *Nuclear Science and Engineering*, 195(5):520–537, 2021.

[72] Seung-Hwan Seong, Heui-Youn Park, Dong-Hoon Kim, Yong-Suk Suh, Seop Hur, In-Soo Koo, Un-Chul Lee, Jin-Wook Jang, and Yong-Chul Shin. Development of fast-running simulation methodology using neural networks for load follow operation. *Nuclear science and engineering*, 141(1):66–77, 2002.

[73] Priscila Palma Sanchez and Adimir dos Santos. Prediction of the Power Peaking Factor in a Boron-Free Small Modular Reactor Based on a Support Vector Regression Model and Control Rod Bank Positions. *Nuclear Science and Engineering*, 195(5):555–562, 2021.

[74] Helin Gong, Yingrui Yu, and Qing Li. Reactor power distribution detection and estimation via a stabilized gappy proper orthogonal decomposition method. *Nuclear Engineering and Design*, 370:110833, 2020.

[75] H. Gong, Q. Li, Y.-R. Yu, J.-P. Argaud, B. Bouriquet, Y. Maday, and O. Mula. A new data-driven approach for reconstruction with noisy data and physical constraints: application to nuclear reactor physics. In *ICAPP 2019*, pages 3318–3327. SFEN, 2019.

[76] Helin Gong, Zhang Chen, Yvon Maday, and Qing Li. Optimal and fast field reconstruction with reduced basis and limited observations: application to reactor core online monitoring. *Nuclear Engineering and Design, accepted*, 2020.

[77] H. Gong, J.-P. Argaud, B. Bouriquet, and Y. Maday. The Empirical Interpolation Method applied to the neutron diffusion equations with parameter dependence. In *Proceedings of PHYSOR*, 2016.

[78] H. Gong, J.-P. Argaud, B. Bouriquet, Y. Maday, and O. Mula. Monitoring flux and power in nuclear reactors with data assimilation and reduced models. In *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2017), Jeju, Korea*, 2017.

[79] J.-P. Argaud, B. Bouriquet, H. Gong, Y. Maday, and O. Mula. Stabilization of (G) EIM in presence of measurement noise: application to nuclear reactor physics. In *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2016*, pages 133–145. Springer, 2017.

[80] H. Gong, Y. Maday, O. Mula, and T. Taddei. PBDW method for state estimation: error analysis for noisy data and nonlinear formulation. *arXiv preprint arXiv:1906.00810*, 2019.

[81] César Quilodrán Casas, Rossella Arcucci, Pin Wu, Christopher Pain, and Yi-Ke Guo. A reduced order deep data assimilation model. *Physica D: Nonlinear Phenomena*, 412:132615, 2020.

[82] Cong Feng, Mucun Sun, Morteza Dabbaghjamanesh, and Yuanzhi Liu. *Advanced machine learning applications to modern power systems*, pages 209–257. 01 2021.

[83] M. Dabbaghjamanesh, A. Moeini, N. D. Hatziargyriou, and J. Zhang. Deep learning-based real-time switching of hybrid ac/dc transmission networks. *IEEE Transactions on Smart Grid*, pages 1–1, 2020.

[84] Chen Wang, Pierre Tandeo, Alexis Mouche, Justin E. Stopa, Victor Gressani, Nicolas Longepe, Douglas Vandemark, Ralph C. Foster, and Bertrand Chapron. Classification of the global sentinel-1 sar vignettes for ocean surface process studies. *Remote Sensing of Environment*, 234:111457, 2019.

[85] Sibo Cheng, Jean-Philippe Argaud, Bertrand Iooss, Angélique Ponçot, and Didier Lucor. A graph clustering approach to localization for adaptive covariance tuning in data assimilation based on state-observation mapping, *submitted.* 2020.

[86] Julian Mack, Rossella Arcucci, Miguel Molina-Solana, and Yi-Ke Guo. Attention-based convolutional autoencoders for 3d-variational data assimilation. *Computer Methods in Applied Mechanics and Engineering*, 372:113291, 2020.

[87] Hamid Khayyam, Gelayol Golkarnarenji, and Reza N Jazar. Limited data modelling approaches for engineering applications. In *Nonlinear Approaches in Engineering Applications*, pages 345–379. Springer, 2018.

[88] R. Arcucci, Laetitia Mottet, Chris Pain, and Yi-Ke Guo. Optimal reduced space for variational data assimilation. *Journal of Computational Physics*, 379:51–69, 2018.

[89] Sibo Cheng, Jean-Philippe Argaud, Bertrand Iooss, Didier Lucor, and Angélique Ponçot. Error covariance tuning in variational data assimilation: application to an operating hydrological model. *Stochastic Environmental Research and Risk Assessment*, 2020.

[90] Yan-Yan Song and Ying Lu. Decision tree methods: applications for classification and prediction. *Shanghai archives of psychiatry*, 27:130–5, 04 2015.

[91] Luigi Vanfretti and V.S. Narasimham Arava. Decision tree-based classification of multiple operating conditions for power system voltage stability assessment. *International Journal of Electrical Power I& Energy Systems*, 123:106251, 2020.

[92] Mohammad Reza Dehbozorgi, Mohammad Rastegar, and Morteza Dabbaghjamanesh. Decision tree-based classifiers for root-cause detection of equipment-related distribution power system outages. *IET Generation, Transmission & Distribution*, 14(24):5809–5815, 2020.

[93] A. Poterie, J.-F. Dupuy, V. Monbet, and L. Rouvière. Classification tree algorithm for grouped variables. *Computational Statistics*, 34(4):1613–1648, Dec 2019.

[94] S. Cheng, J.-P. Argaud, B. Iooss, D. Lucor, and A. Ponçot. Background error covariance iterative updating with invariant observation measures for data assimilation. *Stochastic Environmental Research and Risk Assessment*, 33(11):2033–2051, 2019.

[95] Seyed Abolfazl Hosseini and Iman Esmaili Paeen Afrakoti. On a various soft computing algorithms for reconstruction of the neutron noise source in the nuclear reactor cores. *Annals of Nuclear Energy*, 114:19–31, 2018.

[96] S Narasimhan and V Rajendran. Decision Trees for Nuclear Power Plant Performance Optimization.

[97] José Manuel Cadenas, Juan José Ortiz-Servin, David A Pelta, José-Luis Montes-Tadeo, Alejandro Castillo, and Raúl Perusquía. Prediction of 3D nuclear reactor's operational parameters from 2D fuel lattice design information: A data mining approach. *Progress in Nuclear Energy*, 91:97–106, 2016.

[98] Pedro Mena, RA Borrelli, and Leslie Kerby. Nuclear Reactor Transient Diagnostics Using Classification and AutoML. *Nuclear Technology*, pages 1–14, 2021.

[99] M. Rahmatian, B. Vahidi, A.J. Ghanizadeh, G.B. Gharehpetian, and H.A. Alehosseini. Insulation failure detection in transformer winding using cross-correlation technique with ann and k-nn regression method during impulse test. *International Journal of Electrical Power I& Energy Systems*, 53:209–218, 2013.

[100] Xingjie Peng, Yun Cai, Qing Li, and Kan Wang. Control rod position reconstruction based on K-Nearest Neighbor Method. *Annals of Nuclear Energy*, 102:231–235, 2017.

[101] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*, volume 1 of *Springer Series in Statistics*. Springer New York Inc., New York, USA, 2001.

[102] Helin Gong, Yingrui Yu, Qing Li, and Chaoyu Quan. An inverse-distance-based fitting term for 3D-Var data assimilation in nuclear core simulation. *Annals of Nuclear Energy*, 141:107346, 2020.

[103] X. Li, Q. Liu, Q. Li, L. Chen, X. Liu, S. Wang, Y. Xie, and Z. Chen. 177 Core Nuclear Design for HPR1000. *Nuclear Power Engineering*, 40 S1:8–12, 2019.

[104] R. Dautray and J.-L. Lions. *Mathematical Analysis and Numerical Methods for Science and Technology: Volume 6 Evolution Problems II*. Springer Science & Business Media, 2012.

[105] Pascal Vincent and Y. Bengio. K-local hyperplane and convex distance nearest neighbor algorithms. *Advances in Neural Information Processing Systems*, 04 2002.