On the accuracy and efficiency of the reactor operation digital twin for parameter identification and state estimation

Lizhan Hong,[1] Helin Gong,[1,*] Hongjun Ji,[1] Jialiang Lu,[1] Han Li,[1] and Qing Li[2]

[1]*Paris Elite Institute of Technology, Shanghai Jiao Tong University, Shanghai 200240, China*
[2]*Science and Technology on Reactor System Design Technology Laboratory, Nuclear Power Institute of China, 610041, Chengdu, China.*

Accurate and efficient online parameter identification and state estimation are crucial to leveraging Digital Twin simulations for optimizing the operation of near-carbon-free nuclear energy systems. In previous works, we have developed the idea of Reactor Operation Digital Twin (RODT). However, non-differentiabilities and discontinuities arise when employing some machine learning based surrogate forward models, challenging traditional gradient-based inverse methods and their variants. This study investigates deterministic and metaheuristic algorithms and further develops hybrid algorithms to address such issues. An efficient modular RODT software framework is presented incorporating these methods into its post-evaluation module for comprehensive comparison. Methods are rigorously assessed based on convergence profiles, stability with respect to noise, and computational performance. Numerical results show that the hybrid KNNLHS algorithm excels in real-time online applications, balancing accuracy and efficiency with a prediction error rate of just 1% and processing times under 0.1 seconds. Contrastingly, algorithms like FSA, DE, and ADE, though slightly slower (approximately 1 second), demonstrate higher accuracy with a 0.3% relative L_2 error.This work advances RODT methodologies to harness machine learning and systems modelling for improved reactor monitoring, systematic diagnosis of off-normal events, lifetime management strategies, etc. The developed modular software and novel optimization methods presented offer pathways to realize the full potential of RODT for transforming energy engineering practices.

Keywords: Parameter Identification; State Estimation; Reactor Operation Digital Twin; Reduced Order Model; Inverse Problem

I. INTRODUCTION

A. The Concept of Digital Twins (DTs)

The concept of DTs has gained significant traction in recent years, particularly in the realm of engineering and industrial disciplines [1]. A DT refers to a virtual model that closely mirrors a real-world physical product, system, or process, serving as an effectively indistinguishable digital counterpart for practical purposes such as simulation, integration, testing, monitoring, and maintenance. DTs have been recognized as the fundamental premise for Product Lifecycle Management, encompassing the entire lifecycle of the physical entity they represent, including creation, construction, operation/support, and disposal [2–6].

The value and fidelity of a DT representation depend on the specific use cases, considering its granularity and complexity tailored accordingly. Notably, a DT can be conceptualized and utilized even before the physical counterpart exists, allowing for comprehensive modelling and simulation of the intended entity's lifecycle.

The origins of the DT concept can be traced back to NASA, which pioneered its practical definition in 2010 as part of efforts to enhance the simulation of spacecraft using physical-model representations [2, 6]. Since then, the development of DTs has progressed in tandem with advances in product design and engineering. Traditional manual drafting techniques have given way to computer-aided drafting and designs, and to model-based system engineering approaches, which establish a direct link between the DT and signals from its physical counterpart.

In recent years, DT technology has garnered substantial attention across various industrial sectors. Notably, its application in the chemical process industry demonstrates its versatility [7]. Additionally, DTs are increasingly utilized in simulation-based vehicle certification and fleet management [8], as well as in addressing the challenges of unpredictable and undesirable behaviours in complex systems [9, 10]. A specific area of growth is observed in the nuclear plant industry, where digital twin architectures have been developed for the management and monitoring of nuclear plants, reflecting the technology's expanding scope and potential [11, 12].

B. Previous Work on Reactor Operation Digital Twins (RODTs)

Significant research has been conducted to explore the potential and applications of RODT in the domain of nuclear energy. RODTs represent a specific instantiation of the DT concept, focusing on the numerical representation of nuclear reactors for real-time prediction, optimization, monitoring, control, and decision-making during their operational stage [13].

* Corresponding author, gonghelin@sjtu.edu.cn

The previous work [13] introduced the first prototype of the RODT specifically tailored for the prediction of neutron flux and power fields in the operational stage of the HPR1000 reactor core [14]. The prototype demonstrated the feasibility of using the RODT for online monitoring and prediction of key reactor parameters, thereby enhancing operational understanding and performance assessment.

In the study [15], the forward model of the RODT is realized through the application of a non-intrusive reduced-order model constructed using an SVD autoencoder, learning the non-linearity of the field distribution. Machine learning techniques, specifically leveraging K-Nearest Neighbour (KNN) and Decision Tree, are employed to establish the forward mapping. Additionally, the inverse model adopts a generalized latent assimilation method for accurate estimation of model parameters.

While demonstrating promise, opportunities remained to improve the efficacy of the RODT for practical deployment. Specifically, the inverse solver relied upon a computationally expensive methodology, limiting its potential integration within an online framework. Therefore, the aim of the work [16] was to enhance key capabilities of the RODT. This included proposing an advanced differential evolution algorithm to upgrade the inverse solver for improved efficiency and accuracy of parameter estimation. Additionally, uncertainty quantification was conducted to validate the RODT's performance considering noisy observational data. Validation numerical experiments were performed across representative domains for an HPR1000 pressurized water reactor core, demonstrating the potential for engineering applications.

The key steps for constructing a RODT include: (i) training a Reduced Order Model (ROM) using Model Order Reduction (MOR) [17–20] techniques as well as an efficient forward model; (ii) adaptation and implementation of inverse models to infer the input parameters and the resulted field distribution.

C. Introduction to Reduced Order Models (ROMs)

Many mathematical models encountered in real-life processes present computational challenges when employed in numerical simulations due to their inherent complexity and large dimensionality. In order to address these challenges, MOR techniques have been developed to reduce the computational complexity of such problems, particularly in the context of simulations involving large-scale dynamical systems and control systems. By reducing the dimensionality or degrees of freedom associated with a model, a ROM is computed as an approximation to the original model.

The preparation of a ROM model involves two distinct phases: the offline phase and the online phase, which are integral to the creation of a simplified representation of a high-fidelity full-order numerical model, enabling efficient real-time simulations and analysis. The offline phase focuses on determining the inherent low-dimensional structure of the underlying full-order model and deriving reduced basis functions from high-fidelity snapshots of the system. Intrusive MOR techniques [21] are employed when a comprehensive understanding of the governing equations and numerical strategies employed in the full-order model is available, requiring access to the detailed numerical framework of the full-order model, which may involve proprietary commercial codes. By projecting the full-order model onto the reduced space, intrusive ROMs provide a concise representation with reduced computational complexity.

However, in practical engineering scenarios [22–24], detailed information regarding the full-order model is often unattainable, and the solvers or codes implementing these models are treated as black-box entities. This limitation poses challenges to the implementation of traditional intrusive MOR methods. To address this, non-intrusive MOR techniques have been developed. Non-intrusive MOR approaches aim to construct ROMs without relying on detailed knowledge of the full-order model's numerical framework. Instead, these methods establish an input-output mapping between the input parameters and the reduced basis through data-driven techniques such as interpolation, regression, or machine learning, approximating the behaviour of the full-order model based on a reduced set of parameters and basis functions.

ROMs prepared through the offline and online phases play a critical role in the development of RODTs. RODTs necessitate the integration of high-fidelity models and solvers with real-time capabilities, with ROM-based emulators providing a promising solution [21–23, 25]. Leveraging ROM-based forward solvers, RODTs can perform both one-to-one forward simulations and one-to-many simulations for inverse problems, encompassing parameter identification [26–29], data assimilation [24, 30–35], sensitivity/uncertainty quantification [36–40], and optimization/control [37, 41–47].

D. Forward Model Construction

In line with the framework RODT, as investigated in the previous work [13, 15], we select KNN [48] to construct the forward model. The choice of KNN is also justified by its numerous advantages, which render it a suitable candidate for the forward model in this context.

First, it is a non-parametric algorithm, meaning it does not assume any specific functional form for the relationship between inputs and outputs. Second, KNN is relatively simple to implement and efficient to compute, especially for low-dimensional input problems. The algorithm's complexity mainly depends on the number of

training samples, making it suitable for problems with a moderate dataset size [49]. In recent research [50], an investigation was conducted to explore the possibility of utilizing different values of K for each dataset, taking into account the information provided by the correlation matrix.

E. Inverse Model Construction

Generally, the inverse problem can be divided into two categories: (i) the first category involves solving the forward system with the knowledge of an existing dataset of input and output; (ii) the second category involves solving the system's input with the knowledge of the forward system and the output [51]. In our case, the first category corresponds to the construction of the forward model, where we build a non-intrusive model as a surrogate to the full-order system; while in the second category, we primarily focus on two specific scenarios:

- **Parameter Identification.** Solving the parameter vector based on the information of observation vectors: In this scenario, we aim to estimate the unknown parameters of the system based on the available observation vectors. The observation vectors contain information about the system's behaviour or response under conditions which are characterized by the parameter vector;
- **State Estimation.** Solving the field distribution based on the information of observation vectors: In this scenario, the objective is to determine the unknown field distribution of the nuclear system based on the available observation vectors.

F. Contribution of this Work

The previous work [13, 15, 16] have demonstrated the feasibility of employing the RODT framework in engineering applications. However, there is a need for a systematic and in-depth investigation of the core components of RODT, namely parameter identification and state estimation. Additionally, the development of mature and modular software is crucial to facilitate the application of RODT in practical scenarios. This motivates us to find an efficient and proper surrogate inverse model to provide an accurate prediction of parameters and states. This also motivates our research of hybrid optimization approaches that can effectively balance coarse-grid and fine-grid search (Coarse-Fine-Grid Search) methods, which will be discussed in the following sections.

In conclusion, our contributions to this research include:

- Investigation and implementation of novel gradient-free optimization algorithms specifically tailored to handle the challenges posed by the discontinuous surrogate forward mapping constructed with the KNN forward model of our RODT, and an overall comparison in convergence profile, stability and accuracy-time performance is also conducted;
- Exploration and evaluation of the proposed Coarse-Fine-Grid Search methods within various global optimization approaches, aiming to strike a balance between accuracy and computational efficiency in parameter identification and state estimation;
- Development and deployment of an efficient modular RODT software integrated with the mentioned algorithms above into the inverse problem solver, which is capable of being applied to other sustainable energy systems.

These contributions focus on advancing the state-of-the-art in RODT by addressing the unique challenges encountered in solving inverse problems in nuclear engineering and by proposing novel methodologies for improved parameter identification and state estimation. Furthermore, our work addresses the critical challenge of achieving real-time, high-precision simulations in nuclear energy DT applications, significantly enhancing the capabilities of DTs in the nuclear energy sector, and facilitating improved monitoring, control, and optimization of nuclear energy systems.

II. METHODOLOGY

As mentioned above, the modelling process of a RODT can be divided into the offline stage and the online stage:

- **Offline phase**: training a non-intrusive ROM, which includes (i) preparation of the full order snapshots; (ii) preparation of the reduced basis; (iii) preparation of the coefficient; (iiii) preparation of the surrogate forward model to predict the coefficient with the information of the parameter;
- **Online phase**: parameter identification and state estimation, which includes (i) building an inverse model to predict the parameter based on the information of the observation data (clean or with observation noise); (ii) building an inverse model to predict the coefficient based on the information of the observation data (clean or with observation noise).

The concrete methodology can be illustrated in the following flow chart.

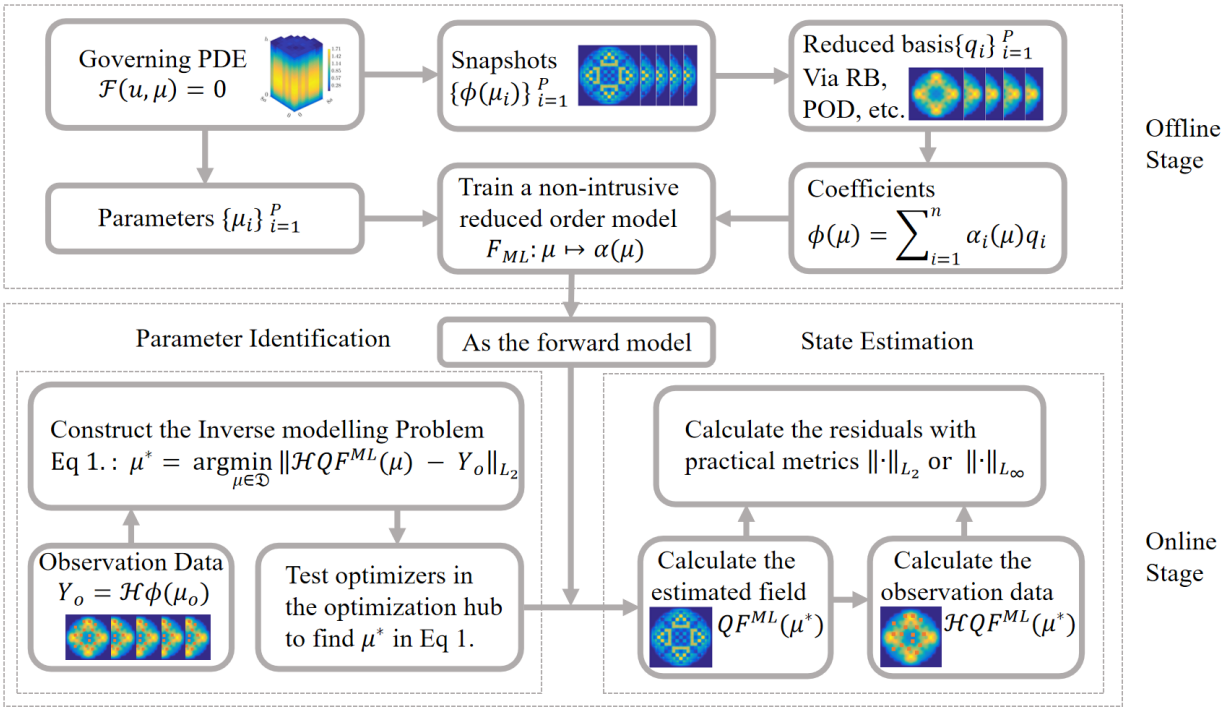


Fig. 1: The working flow of the RODT schema.

A. Offline phase: Training a Non-Intrusive ROM

The variation of the power field Φ in a nuclear core is characterized by physical laws, which are generally described implicitly by a governing equation [52], such as the neutron transport or diffusion equations, in the form of Equation 2.1.

$$\begin{cases} \mathcal{F}(\Phi(r,\boldsymbol{\mu}),\boldsymbol{\mu})=0 \\ r\in\Omega\subset\mathbb{R}^d \\ \boldsymbol{\mu}\in\mathcal{D}\subset\mathbb{R}^p \end{cases}, \tag{2.1}$$

where $\Omega\subset\mathbb{R}^d$ represents the spatial domain of dimension d, with $d\geq 1$, and $\phi(\boldsymbol{\mu})=\Phi(r,\boldsymbol{\mu})$ is defined in a Hilbert space which is equipped with an inner product $\langle\cdot,\cdot\rangle$ and the induced norm $\|\cdot\|=\sqrt{\langle\cdot,\cdot\rangle}$. $\mathcal{D}\subset\mathbb{R}^p$ presents the p-dimensional feasible parameter domain which covers the operation process of the reactor core. In this work, Equation 2.1 represents two-group neutron diffusion equations [53].

1. *Preparation of the Full-Order Snapshots*

In this study, the numerical solution of Equation 2.1 is obtained using the proprietary CORCA-3D code package, which was developed at the Nuclear Power Institute of China (NPIC) [54]. The CORCA-3D code is considered a black-box solver, meaning it can be substituted with other proprietary codes if desired.

To construct the non-intrusive ROM, it is necessary to gather a set of full-order solution snapshots. We sample $\boldsymbol{\mu}_i$ in the parameter domain $\mathcal{D}$ to get a discrete set $\boldsymbol{D}=\{\boldsymbol{\mu}_i\in\mathbb{R}^p\mid\boldsymbol{\mu}_i\in\mathcal{D},i=1,\ldots,P\}$ which is representative of the whole set $\mathcal{D}$. With the help of CORCA-3D, we can also get a solution snapshot set $\boldsymbol{M}=\{\phi(\boldsymbol{\mu})\in\mathbb{R}^N\mid\boldsymbol{\mu}\in\boldsymbol{D}\}$.

As mentioned above, knowledge of this manifold is implicitly dependent on the knowledge of the parameters $\boldsymbol{\mu}$, and for better constructing a non-intrusive model in-between, we endeavour to find a low-rank representation for the solution snapshot set $\boldsymbol{M}$. Moreover, reduced basis methods (RB) [55–57]

indicate that in the state Φ is sufficiently regular, it can be approximated by an n-dimensional reduced basis $\{q_i\}$ as follows:

$$\phi(\boldsymbol{\mu})\cong\phi_n(\boldsymbol{\mu})=\sum_{i=1}^{n}\alpha_i(\boldsymbol{\mu})q_i. \tag{2.2}$$

For the sake of simplicity, here we denote the approximation of the solution manifold as follows:

$$\phi_n(\boldsymbol{\mu})=\boldsymbol{Q}\alpha(\boldsymbol{\mu})\,, \tag{2.3}$$

where $\boldsymbol{Q}\in\mathbb{R}^{N\times n}$ represents the assembled basis $N\times n$-dimensional matrix $\{q_i,i=1,...,n\}$, and $\alpha(\boldsymbol{\mu})\in\mathbb{R}^n$

denotes the assembled coefficient n-dimensional vectors $\{\alpha_i, i = 1, ..., n\}$.

2. Preparation of the Reduced Basis

For the sake of simplicity and efficiency in time, we finally choose the Singular Value Decomposition (SVD) method for generating the reduced basis. The concrete steps can be illustrated as follows:

Correlation Matrix. The correlation matrix $\mathbf{C}$ is computed by taking the inner product between each pair of snapshots, as expressed by the equation:

$$\mathbf{C}_{i,j} = \frac{1}{P}\langle\phi(\boldsymbol{\mu_i}), \phi(\boldsymbol{\mu_j})\rangle, \quad \forall 1 \leq i, j \leq P. \tag{2.4}$$

Here, $\langle\phi(\boldsymbol{\mu_i}), \phi(\boldsymbol{\mu_j})\rangle$ represents the inner product between the i-th and j-th snapshots. The eigenvalues λ_i and the corresponding eigenvectors v^i of the correlation matrix $\mathbf{C}$ are then computed.

Proper Orthogonal Decomposition (POD). The j-th POD basis vector q^j, which is independent of the parameter $\boldsymbol{\mu}$, is obtained as a linear combination of the snapshots:

$$q^j = \sum_{i=1}^{N} v_i^j \phi^i. \tag{2.5}$$

In this equation, v_i^j represents the i-th element of the j-th eigenvector, and ϕ^i corresponds to the i-th snapshot. The magnitude of the j-th eigenvalue λ_j provides information about the relative importance of the j-th POD basis vector

Collection of the best basis minimizing the Frobenious error. We consider the solution snapshot set denoted as $M = \{\phi(\boldsymbol{\mu_k})\}_{k=1}^{P} \subset \mathbb{R}^N$. Applied to the computation, we define the snapshot matrix $\boldsymbol{S} \in \mathbb{R}^{N\times P}$ which contains the snapshots $\phi(\boldsymbol{\mu}_k)$ as its columns. Assemble the first n POD basis in a matrix $\boldsymbol{Q}_n = [q_1, \ldots, q_n] \in \mathbb{R}^{N\times n}$. Among all orthonormal bases of size n, the POD basis minimizes the Frobenious norm least squares error (defined in the form of $\| \bullet \|_{\mathrm{F}} = \sqrt{\sum_i^m \sum_j^n |\bullet_{ij}|^2}$) of the reconstruction of snapshot matrix $\boldsymbol{S}$, which further corresponds to the Echart-Young theorem [58]:

$$\min_{\boldsymbol{Q}_n} \left\|\boldsymbol{S} - \boldsymbol{Q}_n\boldsymbol{Q}_n^T\boldsymbol{S}\right\|_F^2 = \sum_{k=n+1}^{N} \lambda_k, \tag{2.6}$$

where λ_k represents the k-th singular value of matrix S. In summary, the POD basis constitutes a set of orthonormal vectors that offer the most effective n-dimensional representation of the given snapshots.

3. Preparation of the Coefficient

Following the procedure mentioned above, the coefficient is constructed as the projection of the snapshot matrix S onto the vector space spanned by the POD basis $\mathrm{Span}\{q_1, q_2, \ldots, q_n\}$

For any $\phi(\boldsymbol{\mu})$ belonging to the set $\mathcal{S}$, the n-dimensional approximation of $\phi(\boldsymbol{\mu})$ is given by Equation 2.2. In this manner, the coefficient can be computed with the orthogonal projection as follows:

$$\alpha(\boldsymbol{\mu}) = \sum_{i=1}^{n} \frac{\langle\phi(\boldsymbol{\mu}), q_i\rangle}{\langle q_i, q_i\rangle} = \sum_{i=1}^{n} q_i^{\mathrm{T}}\phi(\boldsymbol{\mu}) \,, \tag{2.7}$$

where, the $\langle q_i, q_i\rangle$ equals 1, due to the orthogonormality of the POD basis.

Moreover, in matrix form, the coefficient matrix $\boldsymbol{A}_n = [\alpha_1(\boldsymbol{\mu}), ..., \alpha_n(\boldsymbol{\mu})]^{\mathrm{T}} \in \mathbb{R}^{n\times P}$ follow the relationship as follows:

$$\boldsymbol{S} \cong \boldsymbol{S}_n = \boldsymbol{Q}_n\boldsymbol{A}_n = \boldsymbol{Q}_n\boldsymbol{Q}_n^T\boldsymbol{S} \,. \tag{2.8}$$

4. Training a Forward Model

The goal of the forward model is to build an efficient surrogate mapping model between the parameter and the coefficient as follows:

$$\alpha(\boldsymbol{\mu}) \cong F^{ML}(\boldsymbol{\mu}), \tag{2.9}$$

where F^{ML} denotes the surrogate forward mapping constructed by the machine learning approach, for instance, model-dependant machine learning methods or model-free optimization searching methods.

And the response of a ROM model denoted as $F^{ML}(\boldsymbol{\mu})$, corresponds to the coefficient data predicted. The image of space $\boldsymbol{D}$ under the forward map $F^{ML}(\boldsymbol{\mu})$ represents the set of responses for all possible states. The difference $Y_o - \mathcal{H}\boldsymbol{Q}F^{\mathrm{ML}}(\boldsymbol{\mu})$, referred to as observation data misfits or residuals associated with the parameter $\boldsymbol{\mu}$.

We design the cost function as Equation 2.12 based on the residual. The forward model in the search of the optimal approximation for the operator ϕ in the functional set $\{\boldsymbol{Q}F^{ML}, F^{ML} \in \mathcal{E}(\boldsymbol{D}, \mathbb{R}^n)\}$, where $\mathcal{E}(\boldsymbol{D}, \mathbb{R}^n)$ denotes the set of functions mapping from $\boldsymbol{D}$ to $\mathbb{R}^n$, with n the reduced dimension in the MOR process.

Previous research [13] has evaluated various forward model approaches, and among them, the KNN algorithm has demonstrated exceptional accuracy and efficiency. In this study, we specifically select the KNN algorithm [48] as the forward model to exploit its full potential. To enhance the performance of the KNN algorithm, we concentrate on two crucial parameters that significantly influence its efficacy.

The first parameter relates to the voting distance in KNN, wherein we investigate the impact of utilizing either the Euclidean distance or the Manhattan distance as the metric for determining the nearest parameter choice in the training set. Notably, these two distances can be expressed as the Minkowski norm with an exponent value of $p = 1, 2$, corresponding to the Euclidean distance and the Manhattan distance, respectively. The Minkowski distance, a measure of the discrepancy between the testing parameter and the training parameter, can be mathematically represented as:

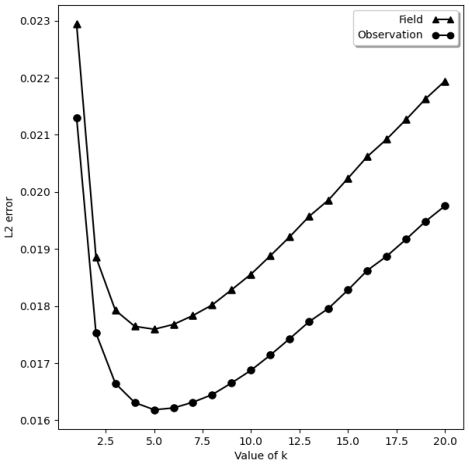
$$\left\|\boldsymbol{\mu}^{test} - \boldsymbol{\mu}^{train}\right\|_p = \left(\sum_{i=1}^{dim(\boldsymbol{\mu})} |\boldsymbol{\mu}_i^{test} - \boldsymbol{\mu}_i^{train}|^p \right)^{\frac{1}{p}}, \tag{2.10}$$

where $dim(\boldsymbol{\mu})$ denotes the dimension of the parameter $\boldsymbol{\mu}$, and we finally choose the Euclidean metric.

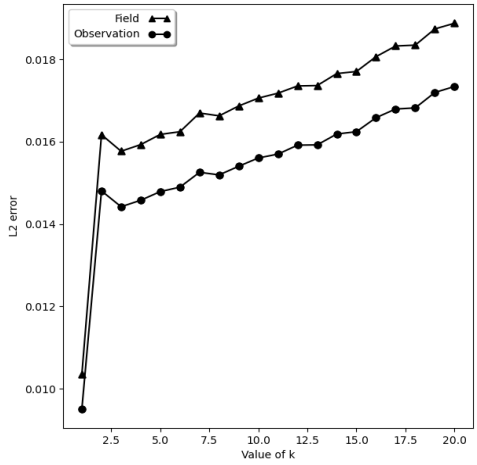
Another vital parameter is the value of K, referring to the number of voted candidates. The optimal choice of K can be determined by plotting the relative L_2 error of the predicted observation vector and the true vector, which is demonstrated as follows.

From Figure 2a, we can deduce that when the K equals 5, our forward model using KNN performs with the best accuracy of prediction, and the variation of the error is relatively smooth, which corresponds to the continuous variation of the value of the field at the corresponding position of the field manifold snapshot.

Whereas, the stair-shaped observation data predictions in Figure 3 demonstrate that the KNN forward surrogate model has a step-like progression as neighbours are adjusted between discrete datasets, thus not continuous, and not differentiable. In this manner, the well-developed state-of-the-art algorithms based on gradient information for solving optimization problems and inverse problems could not work well in our case. In this demand, we investigate other novel approaches, which will be discussed in detail in the following sections.



(a) KNN Forward model



(b) KNN Inverse model

Fig. 2: Finding the optimal choice of K for KNN model (Forward on the left, Inverse on the right).

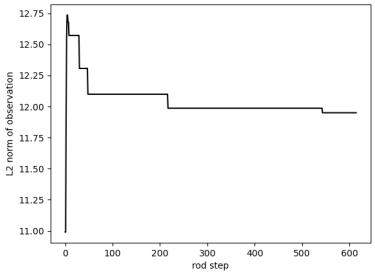
B. Online phase: Parameter Identification and State Estimation

The inverse problem in nuclear engineering refers to the task of determining the model parameters that give rise to the observed data, as opposed to the forward problem which involves predicting data based on given model parameters. The objective is to find the model parameters $\boldsymbol{\mu}^*$ that approximately satisfy Equation 2.12.

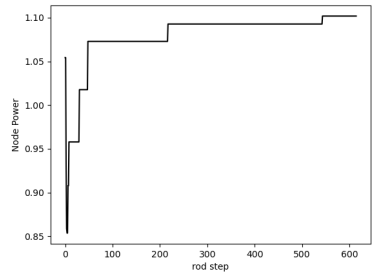
The nature of the inverse problem depends on whether the operator ϕ is linear or nonlinear. In most cases, particularly in solving systems governed by the neutron flux diffusion function, the inverse problem is nonlinear due to the nonlinearity of the forward map. Thus we try to develop nonlinear algorithms, gradient-free algorithms or algorithms based on the initial guess given by fast linear algorithms such as KNN.

1. *Setup of Inverse Model and Optimization*

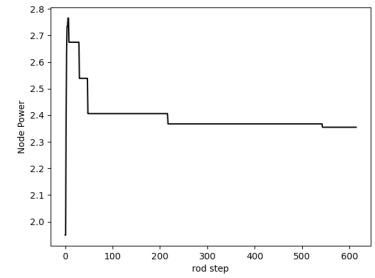
In the context of optimal control theory, the governing equations that describe the behaviour of a physical system are commonly referred to as state equations. However, in many practical scenarios, the interest lies not only in the physical state itself but also in its impact on certain objects or quantities. Furthermore, it is often the case that only a limited amount of data can be obtained



(a) L_2 norm of predicted observation



(b) 1th vertical level, the 43th assembly



(c) 18th vertical level, the 11th assembly

Fig. 3: The non-differentiable stair-shaped observation data prediction via the KNN surrogate forward model (tested when the first parameter S_t varies from 0 to 615 steps, the other parameters take the midpoint values of the parameter intervals respectively, 1250 for B_u; 50 for P_w; 295 for T_{in}).

from the physical state. To address these considerations, an additional operator denoted as $\mathcal{H}$, known as the observation operator, is introduced. This operator maps the state of the physical system, denoted as Φ, to the desired observations denoted as $Y_o \in \mathbb{R}^m$, where m denotes the total number of observations. Given a field Φ, the observation vector Y_o is given by the equation below and approximated with an error:

$$Y_o = \mathcal{H}\Phi = \mathcal{H}\boldsymbol{Q}F^{\mathrm{ML}}(\boldsymbol{\mu}) + e. \tag{2.11}$$

Here, $e \in \mathbb{R}^m$ represents the observation noise vector, which captures the presence of observational errors. The quality of a given set of model parameters $\boldsymbol{\mu}$ is evaluated using the L_2 distance between the simulated observation and the practical observation, which is quantified by the following cost function:

$$J(\boldsymbol{\mu}) := \left\|\mathcal{H}\boldsymbol{Q}F^{\mathrm{ML}}(\boldsymbol{\mu}) - Y_o\right\|_{L_2}^2. \tag{2.12}$$

The objective of the inverse problem is to minimize the observation error, and it can be formulated as follows:

$$\boldsymbol{\mu}^* = \underset{\boldsymbol{\mu}\in\mathcal{D}}{\operatorname{argmin}} J(\boldsymbol{\mu}) := \underset{\boldsymbol{\mu}\in\mathcal{D}}{\operatorname{argmin}} \left(\left\|\mathcal{H}\boldsymbol{Q}F^{\mathrm{ML}}(\boldsymbol{\mu}) - Y_o\right\|_{L_2}^2\right). \tag{2.13}$$

It is important to note that the cost function described above is possibly nonlinear and non-differentiable concerning the discontinuous function F^{ML}. Thus, we develop the following algorithms to solve this optimization problem.

III. COMPARISON OF DIFFERENT APPROACHES FOR SOLVING INVERSE MODELS

To solve inverse problems, machine learning and optimization methods can be employed to search for optimal parameters or field distributions that best fit observed data [59]. Additionally, as mentioned in Section II A 4, when we are employing the KNN algorithm to construct the surrogate forward model, the surrogate forward mapping is non-differentiable and discontinuous. This poses a challenge, as traditional approaches used for solving continuous optimization problems and inverse problems [60] are not applicable in this context. This necessitates an exploration of novel inverse problem-solving methods that are suitable for our particular problem. Hence, this paper investigates various deterministic optimization algorithms, metaheuristic algorithms and their hybrids to address this challenge.

A. Exhaustive Direct Search with Latin Hypercube Sampling (LHS)

To evaluate and compare different approaches for solving inverse models, it is essential to establish a benchmark for measuring their performance. In this section, we propose using Exhaustive Direct Search (EDS) with Latin Hypercube Sampling (shortened as LHS) as a benchmark method for the inverse problem. It combines the accuracy of EDS methods, of which it has been widely used since the 1960s [61], with the sampling efficiency of LHS.

LHS [62] is a sampling technique that ensures a comprehensive exploration of the solution space while maintaining desirable properties. Each variable's range is divided into equally probable intervals, and sample points

are strategically placed to satisfy the requirements of a Latin hypercube, where each sample is the only one in its corresponding axis-aligned hyperplane. The advantage of LHS lies in its independence from the number of dimensions, allowing for efficient sampling without the need for an increasing number of samples. Furthermore, LHS facilitates sequential sampling, enabling the tracking of already selected samples.

B. Exhaustive Direct Search with 2 Steps Latin Hypercube Sampling (LHS2STEPS)

In an effort to enhance our benchmark algorithm and mitigate the uncertainties introduced by the randomness of the sampling process, the work by [13] introduces a two-step sampling strategy. This strategy involves further sampling the neighborhoods of the n_1 best candidates generated during the initial stage of the LHS algorithm. For detailed pseudo-code, please refer to 1, excluding the part related to ALHS.

C. Exhaustive Direct Search with Assembled 2 Steps Latin Hypercube Sampling (ALHS)

To improve our benchmark algorithm, and further reduce the uncertainty brought by the randomness of the sampling process, we introduce a taking mean mechanism by the end of the classical LHS algorithm, to calculate the mean of the n_1 top-rated candidates in the 5 clones, which is spread around the n_2 top-rated candidates in the first sampling process. This idea can be further illustrated in Algorithm 1.

D. An Efficient Deterministic Machine Learning Algorithm: KNN

In addition to the benchmark LHS method, another efficient deterministic algorithm that can be considered for solving inverse models is the KNN algorithm. KNN is a non-parametric classification and regression algorithm that can be adapted for solving our inverse problems.

The KNN algorithm operates based on the principle of similarity. Given a new input data point, KNN finds the K nearest neighbours in the training dataset based on Euclidean distance and then generates the output value by averaging the output values of the K nearest neighbours. The selection of the optimal value for K can be determined by graphing the relative L_2 error between the predicted observation vector and the true vector, the details can be found in Figure 2b. The analysis of Figure 2b reveals that selecting K=1 yields the optimal outcome in terms of minimizing the L_2 prediction error, thereby emphasizing its advantageous characteristic of accuracy.

E. Metaheuristics Algorithms with Physical Constraints

In recent years, the application of metaheuristic algorithms with physical constraints [63–65] has gained significant attention in the field of inverse modelling. These algorithms offer powerful gradient-free optimization techniques that can effectively handle complex inverse problems while satisfying the physical constraints imposed by the system dynamics.

In this manner, we have implemented and adapted practical algorithms and further constructed the related solvers integrated into the developed RODT framework. These solvers are designed to address the inverse problem presented by Equation 2.12.

1. *Differential Evolution Algorithm (DE)*

The DE algorithm [66] offers a promising approach to tackle our inverse modelling problem. By leveraging its ability to handle non-linear and non-convex objective functions, DE can effectively explore the parameter space. Furthermore, DE provides mechanisms to incorporate physical constraints, ensuring that the solutions generated adhere to the system dynamics. The concrete procedure of the DE algorithm can be found in the previous work [13], in this work, we use the best mutation strategy recommended in the work [67] for the mutation step in DE, the mutation donor $V_{i,G}$ could be given by Equation 3.14.

$$V_{i,G} = \mu_{r_5,G} + F \cdot (\mu_{r_1,G} - \mu_{r_2,G} + \mu_{r_3,G} - \mu_{r_4,G}), \tag{3.14}$$

where the $r_{k,G}$, with $k \in \{1, 2, 3, 4, 5\}$ is randomly chosen from $\{1, 2, ...NP\}$, NP is the number of population, μ indicate the parameter and F and G denote the scaling factor and the number of generation respectively.

2. *Particle Swarm Optimization Algorithm (PSO)*

PSO [68] is a population-based optimization technique inspired by the collective behaviour of bird flocking or fish schooling. It involves particles representing potential solutions, which update their positions based on personal and global best positions. The algorithm iterates for a set number of times, adjusting particle velocities and positions. The best solution found, represented by the global best position with the lowest objective function value, is returned.

3. *Fast Simulated Annealing Algorithm (FSA)*

The SA [69] algorithm, initially proposed for solving the famous Travelling Salesman Problem (TSP) [70], is

Algorithm 1 Exhaustive Direct Search with Assembled 2 Steps Latin Hypercube Sampling

Inputs:
Observations: Y_o
Initial guess: $\boldsymbol{\mu}_{\text{initial}}$
Forward function: F^{ML}
Transformation operator: H
First stage sampling number: $n^{s,1}$
Second stage sampling number: $n^{s,2}$
$\mathcal{U}^1 = \text{LHS}\left(\text{ number } = n^{s,1},\ \text{initial } = \boldsymbol{\mu}_{\text{initial}}\right)$
$\boldsymbol{\mu}_1^* = \underset{k=1..n^s}{\text{argmin}} \left(\left\| HV_n F^{\text{ML}}\left(\boldsymbol{\mu}^k\right) - Y_{\text{o}} \right\|_2\right)$
for i from 2 to n_1 **do** ▷ $n_1 = 5$ in this article.
  $\mathcal{U}^1 = \text{LHS}\left(\text{ number } = n^{s,1},\ \text{initial } = \boldsymbol{\mu}_{\text{initial}}\right)$
  $\boldsymbol{\mu}_1^* = \underset{k=1..n^s}{\text{argmin}} \left(\left\| HV_n F^{\text{ML}}\left(\boldsymbol{\mu}^k\right) - Y_{\text{o}} \right\|_2\right)$
end for
for j from 1 to n_1 **do**
  $\mathcal{U}_j^2 = \text{LHS}\left(\text{ number} = n^{s,2}, \text{initial} = \boldsymbol{\mu}_i^*\right)$
end for
for k from 1 to n_2 **do** ▷ Start of the main part of ALHS, $n_2 = 10$ in this article.
  $\boldsymbol{\mu}_k^* = \underset{\boldsymbol{\mu} \in \bigcup \mathcal{U}_j^2}{\text{argmin}} \left(\left\| HV_n F^{\text{ML}}(\boldsymbol{\mu}) - Y_{\text{o}} \right\|_2\right)$
end for ▷ End of the main part of ALHS.
$\overline{\boldsymbol{\mu}} = \left(\sum_{k=1}^{10} \boldsymbol{\mu}_k^*\right)/10$
Outputs:
Parameter: $\overline{\boldsymbol{\mu}}$
Reconstructed field: $V_n F^{\text{ML}}\left(\boldsymbol{\mu}^*\right)$

a popular approach for solving optimization problems in various fields [71], including nuclear engineering [72].

SA draws inspiration from metallurgical annealing, where a metal is heated and gradually cooled to achieve a more ordered configuration. Similarly, SA starts from a high-energy state (initial solution) and progressively lowers the temperature until it converges to a state of minimal energy (optimal solution).

FSA [73] is an adaptation of Classical Simulated Annealing(CSA) [69] that improves computational efficiency and convergence speed. CSA utilizes a local sampling approach and controls fluctuation variance through an artificial cooling temperature, denoted as $T_c(t)$. FSA modifies the cooling schedule to accelerate convergence by using a cooling temperature $T_f(t)$ that decreases reciprocally to time t, instead of logarithmically.

4. *Cuckoo Search Algorithm (CS)*

In the CS algorithm [74], nests are used to represent potential solutions, while eggs within the nests symbolize candidate solutions. Through an iterative process, the algorithm generates new solutions by employing Lévy flights and subsequently evaluates their fitness. If a newly generated solution proves to be superior to the existing one within a given nest, it replaces the incumbent solution. To maintain population diversity and improve the overall quality of solutions, the population is sorted based on fitness, and a predetermined number of solutions are replaced with new ones generated via random walks. The algorithm terminates upon reaching a predefined maximum number of generations, and the best solution, characterized by the lowest objective function value, is returned as the final outcome.

CS utilizes a balanced combination of a local random walk and a global explorative random walk, controlled by a switching parameter p_a. The local random walk is defined as follows:

$$\mu_i^{t+1} = \mu_i^t + \alpha \otimes H\left(p_a - \epsilon\right) \otimes \left(\mu_j^t - \mu_k^t\right). \quad (3.15)$$

In the above equation, μ_j^t and μ_k^t represent two different solutions randomly selected through permutation. $H(\bullet)$ denotes the Heaviside function, $\otimes$ represents point-to-point multiplication, ϵ is a random number drawn from a uniform distribution, and s represents the step size. The global random walk, on the other hand, is conducted using Lévy flights:

$$\mu_i^{t+1} = \mu_i^t + \alpha L(s, \lambda), \quad (3.16)$$

where, $L(s, \lambda)$ is defined as:

$$L(s, \lambda) = \frac{\lambda \Gamma(\lambda) \sin(\pi\lambda/2)}{\pi} \frac{1}{s^{1+\lambda}}, \quad (s \gg s_0 > 0). \quad (3.17)$$

5. *Artificial Neural Network (NN)*

The NN algorithm [75], inspired by the behaviour of the human brain, simulates interconnected artificial neurons to process information, learn from data, and estimate unknown parameters or states. They consist of interconnected artificial neurons organized into layers, which receive input signals, process them through weighted connections, and produce output signals using activation functions. NNs can learn from input-output examples by adjusting connection weights during training, enabling them to capture underlying patterns and correlations, making them suited for solving physical problems [76].

The combination of the Adam and L-BFGS optimizers in an alternating manner has demonstrated efficacy in training NNs for complex problem solutions [77]. This approach leverages the strengths of the two optimizers to enhance training and improve overall network performance.

However, due to NN's continuous property, it commonly cannot handle discrete problems well, especially when the test data is noised, and this is further discussed in Section V.

F. Hybrids of Optimization Methods for Inverse Problem: Balancing Coarse-Fine-Grid Search

To address the challenges posed by the discontinuity of KNN surrogate forward mapping and noisy observation, we propose hybrid algorithms that combine the strengths of deterministic machine learning methods for coarse-grid search and metaheuristic algorithms for fine-grid search. This methodology can be also found in fields encompassing feature extraction of aerial data [78], automatic free-form optics design [79] and hyperparameter optimization [80]. The objective of the Coarse-Fine-Grid search is to strike a balance between global exploration and local exploitation [81] to achieve accurate, efficient and less sensitive parameter estimation. The hybrid optimization hub designed accordingly can be further illustrated as the following schema.

1. *Cuckoo Search with Differential Evolution (CSDE)*

In the later stages of the standard CS process, a notable issue arises in the form of information waste and slow convergence. These issues stem from the execution of independent evaluations and a lack of sufficient mechanisms for information sharing within the population. Consequently, valuable information fails to propagate effectively, resulting in suboptimal convergence rates and a reduction in search efficiency.

From an alternative perspective, an exploration is conducted into the underlying biological nature of mutation within the standard DE algorithm. In this context, a novel mathematical elucidation of mutation is sought. Notably, researchers in the field of natural sciences have observed that mutations can be understood as a sequence of transformations transpiring within the DNA molecule. Furthermore, these transformations, pertaining to modifications, exhibit a statistical distribution that adheres to the Levy distribution. This finding suggests an analogy between mutations and Levy flights [82]. Drawing inspiration from this observation, it becomes conceivable to integrate the Levy flight process into the generation of the DE population.

In this concern, we develop the CSDE algorithm, which is in the framework of the CS algorithm, integrating the DE mutation operation into its production of the population for each generation. The first step of CSDE is the initialization: Define the objective function $f(\bullet)$. Set population size N, problem dimension D, discovery probability Pa, step size α, scaling factor F, crossover probability CR, maximum iteration times T, and search domain range $[\mu_{lb}, \mu_{ub}]$. The following steps could be represented in Algorithm 2.

2. *Advanced CSDE (ACSDE) and Advanced DE (ADE)*

We utilize the SVC algorithm for classifying the test parameter and denote the centre of the classified hypercube as the initial machine learning guess μ_{SVC}, and then use the CSDE and DE algorithms to further explore the parameter space with the initial population spread around the centre μ_{SVC}. In the SVC process, according to the previous research [15] on the importance rate of the input parameters, the last parameter temperature Tin contributes slightly, and we also considered that the Tin varies in a relatively small boundary, in this manner, we only administer the SVC process to the first three parameters. Then we choose two representative metrics to evaluate the performance of the SVC model:

Confusion Matrix. The confusion matrix [83] provides a comprehensive overview of the binary classification results by displaying the counts of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions. It allows us to assess the model's accuracy by the differentiation between classes and identify potential sources of misclassification. We can infer from Figure 5 that our SVC classifying model predicts the first two parameters with relatively high accuracy while exhibiting confusion concerning the third parameter.

F1-score. The F1-score [84], defined in Equation 3.18, is a popular performance metric for the binary classifier model. In Equation 3.18, TP, FN, FP are the number of true positives, false negatives and false pos-

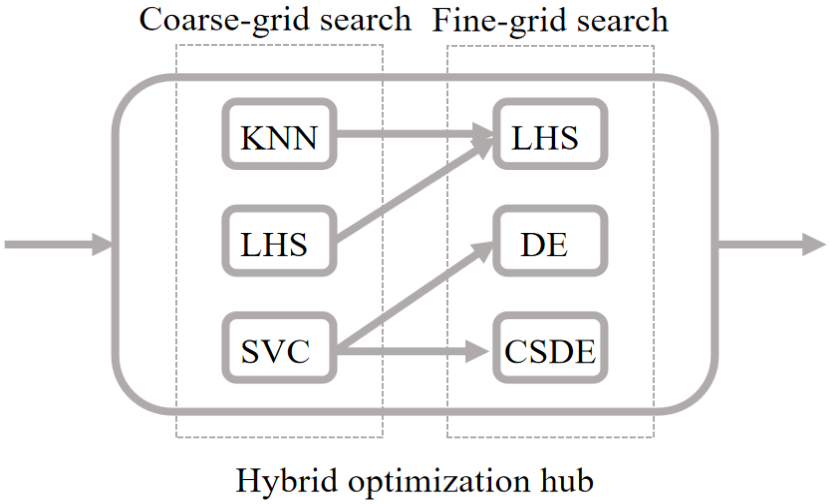


Fig. 4: The hybrid optimization schema of Coarse-Fine-Grid Search.

itives classified by the model respectively.

$$F1 = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \tag{3.18}$$

The predictive model employed in this study for each parameter is a multi-class SVC utilizing the "one-vs-others" strategy. This approach involves training separate binary classifiers to distinguish each candidate interval for a given parameter. Consequently, the evaluation of the performance necessitates the adoption of the Macro F1-score as a metric. The Macro F1-score 3.19 is computed as the average of the F1-scores obtained for each interval, reflecting the overall effectiveness of the classifiers in capturing the intra-parameter variations:

$$\mathrm{F1}_{\mathrm{macro}} := \frac{1}{N} \sum_{i=0}^{N} \mathrm{F1}_i, \tag{3.19}$$

where i is the interval index and N is the number of intervals.

Based on the Confusion Matrix shown in Figure 5 and the $\mathrm{F1}_{\mathrm{macro}}$ shown below, we designed the grid size of the fine-grid search process as follows.

3. *K-Nearest Neighbour plus Exhaustive Direct Search with Latin Hypercube Sampling (KNNLHS)*

Integrating KNN with LHS offers a robust methodology for addressing inverse problems. To enhance the comprehensiveness of our research, we have standardized the grid size for the Coarse-Fine-Grid search. This standardization is in line with the specifications detailed in Table 1.

In conclusion, the KNNLHS method stands out as a robust and efficient approach relative to similar strategies. This technique combines the exploratory capabilities of KNN in conducting coarse-grid searches with the efficiency and diversity of LHS in sampling local solution spaces. As a result, KNNLHS facilitates precise and efficient parameter estimation, adeptly navigating complex objective functions and managing noisy data sets. The effectiveness of this approach and its comparative advantages will be further elucidated in Section V.

IV. APPLICATION TO NUCLEAR REACTOR

A. Setup of Reactor Physics Operation Problems

Setup of background information of HPR1000. The objective of constructing an RODT is to accurately predict the power distribution within the HPR1000 reactor core during its operation. The core consists of 177 vertical nuclear fuel assemblies, and among them, 44 assemblies are equipped with Self-Powered Neutron Detectors (SPNDs) to measure the neutronic activity and power fields. Figure 6 provides a visualization of a horizontal slice of the HPR1000 core and an axial slice of an SPND-equipped assembly, with only one-quarter of the core displayed due to symmetry along the x and y axes. The grey fuel assemblies represent those containing control rods, while the assemblies marked with D indicate the presence of SPNDs. For more detailed information regarding the HPR1000 reactor and the generic neutronic physical model, please refer to [14]. Further descriptions of the model for data assimilation and the initial work on RODT can be found in [35] and [13], respectively.

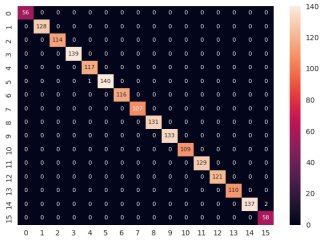
During the normal operation of the HPR1000 reactor,

Algorithm 2 Cuckoo Search Algorithm with Differential Evolution

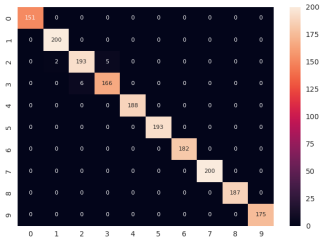
Step 1: Population initialization
Initialize the population P and the initial position using a random distribution within the search domain range $[\mu_{lb}, \mu_{ub}]$
while $G \leq G_{max}$ **do**
  for each individual μ_i in population P_G **do**
    Step 2: Population updating
    Generate random indices $r1, r2, r3$ (different from i)
    for each dimension j in problem dimension $dim(\mu)$ **do**
      if random$(0, 1) < CR$ **then**
        $\mu_{i,j}^G = \mu_{i,j}^G + \alpha \otimes (\mu_{r2,j}^G - \mu_{r3,j}^G) \otimes L(s, \lambda)$
      else
        $\mu_{i,j}^G = \mu_{i,j}^G$
      end if
    end for
    Step 3 Mutation Step
    Generate random indices $r3, r4, r5$ (different from i)
    Generate a donor vector $V_i^G \in \mathbb{R}^{dim(\mu)}$:
    $V_i^G = \mu_{r_3}^G + F \cdot (\mu_{r_4}^G - \mu_{r_5}^G)$
    Step 4 Crossover Step
    Generate a trial vector $U_i^G \in \mathbb{R}^{dim(\mu)}$:
    for $j = 1$ to $dim(\mu)$ **do**
      if $random(0, 1) \leq Cr$ **then**
        $U_{i,j}^G = V_{i,j}^G$
      else
        $U_{i,j}^G = \mu_{i,j}^G$
      end if
    end for
    Step 5 Selection Step
    Evaluate the trial vector U_i^G
    if $f(U_i^G) < f(\mu_i^G)$ **then**
      $\mu_i^{G+1} = U_i^G$
    else
      $\mu_i^{G+1} = \mu_i^G$
    end if
    Step 6 Abandon nests
    Sort the population P_G based on the fitness values in ascending order.
    Determine the number of solutions to be replaced, $n_{\text{replace}} = \text{rand}(p_a \cdot NP)$.
    for $i = 1$ to n_{replace} **do**
      Generate a new solution X in the search space
      with random walk given in Equation 3.15.
      Evaluate the objective function f for X.
      if $f(X) < f(\text{worst solution in } P_G)$ **then**
        Replace the worst solution in P_G with X.
      end if
    end for
  end for
  $G = G + 1$
end while

TABLE 1: Grid size design for the Coarse-Fine-Grid search based on the SVC prediction evaluation.

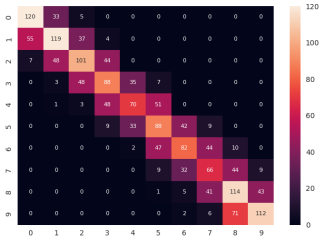
ParameterNames	S_t	B_u	P_w
$\text{F1}_{\textbf{macro}}$	9.98E-01	9.93E-01	5.18E-01
Size coarse-grid(SVC)	41 steps	250 MWd/tU	8% FP
Size fine-grid(DE/CSDE)	41 steps	250 MWd/tU	20% FP
Relative Size coarse-grid(SVC)	6.67%	10.00%	10.00%
Relative Size fine-grid(DE/CSDE)	6.67%	10.00%	25.00%
Parameter Boundary	[0, 615]	[0, 2500]	[20%, 100%]



(a) S_t



(b) B_u



(c) P_w

Fig. 5: The Confusion Matrix Plot that evaluates the efficacy of SVC prediction in the first 3 parameters (S_t, B_u, P_w respectively).

two types of control rods are utilized for regulation. The first type, known as compensation rods, is responsible for coarse control and substantial reactivity reduction. These compensation rods consist of four subtypes: G1, G2, N1, and N2. The second type called regulating rods (R rods), is employed for fine adjustments to maintain the desired power or temperature [85]. The power evolution within the reactor is influenced by various factors, including the movement of control rods, the burnup of nuclear fuel, variations in the power level of the reactor core, and fluctuations in the inlet coolant temperature. This evolution is mathematically modelled using two-group diffusion equations [13] and numerically solved using the CORCA-3D code package [86]. Developed at NPIC, the CORCA-3D code is capable of solving the 3D few-group diffusion equations, considering thermal-hydraulic feedback, and performing pin-by-pin power reconstruction.

Setup of parameter domain. In this study, we consider CORCA-3D as an opaque solver where we input the parameter vector $\boldsymbol{\mu}$ and obtain the output Φ. In particular, the power field Φ is constrained to be dependent on a set of "general" parameters that indicate the stage of the reactor's life cycle:

$$\mu := (S_t, B_u, P_w, T_{in})\,. \tag{4.20}$$

- S_t: The control rod insertion steps, ranging from 0 to 615, represent the movement of the compensation rods from all rod clusters being out (ARO) to fully inserted. This range takes into account the overlap steps.
- B_u: The average burnup of the fuel in the entire core indicates the amount of energy extracted from the fuel and increases over time. Its value ranges between 0 (at the beginning of the fuel's life cycle) and $B_{u,\max}$, which is set to 2500 MWd/tU (the end of the fuel's life cycle). The specific evolution of B_u depends on the reactor's operational history.
- P_w: The power level of the reactor core ranges from 0.3 to 1 FP (Full Power).
- T_{in}: The temperature of the core coolant at the inlet falls within the range of 290 to 300°C.

In this manner, Φ can be represented by $\phi(\boldsymbol{\mu}) = \phi(S_t, B_u, P_w, T_{in})$ implicitly thanks to CORCA-3D. There are 177 fuel assemblies in HPR1000, each assembly is numerically represented using 28 vertical levels. Thus, Φ is a vector of dimension $N = 4956 (= 177 \times 28)$. The discrete solution set $\boldsymbol{M} = \left\{\phi(\boldsymbol{\mu}) \in \mathbb{R}^N \mid \boldsymbol{\mu} \in \boldsymbol{D}\right\}$ consists of $P = 18480$ solution snapshots with the parameter configuration $D := B_u^s \otimes S_t^s \otimes P_w^s \otimes T_{in}^s$, where $S_t^s = \{0, 1, \ldots, 615\}$, $B_u^s = \{0, 50, 100, 150, 200, 500, 1000, 1500, 2000, 2500\}$, $P_w^s = RU3(30, 100)$, and $T_{in}^s = RU3(290, 300)$. The operator $RU3(a, b)$ represents three independent and identically uniformly distributed samplings in the closed set $[a, b]$ where $a < b$. In this article, we select 90% of the 18480 snapshots for training the forward and inverse models, and the left 10% snapshots are used for test purpose.

Setup of observation data and modelling the observation noise. In the context of RODT, the observations denoted as Y_o are utilized to infer the input parameter vector $\boldsymbol{\mu}$ and subsequently determine the resulting power field Φ. The arrangement of these observations, organized node by node, is illustrated in Figure 6. Each observation value represents the average value over the corresponding node where SPNDs are located. It is important to clarify that the observations used in the analysis are not obtained from direct engineering measurements but are derived from numerical simulations of Φ using CORCA-3D.

The noise associated with the observations can be modelled either as a Gaussian distribution or a uniform

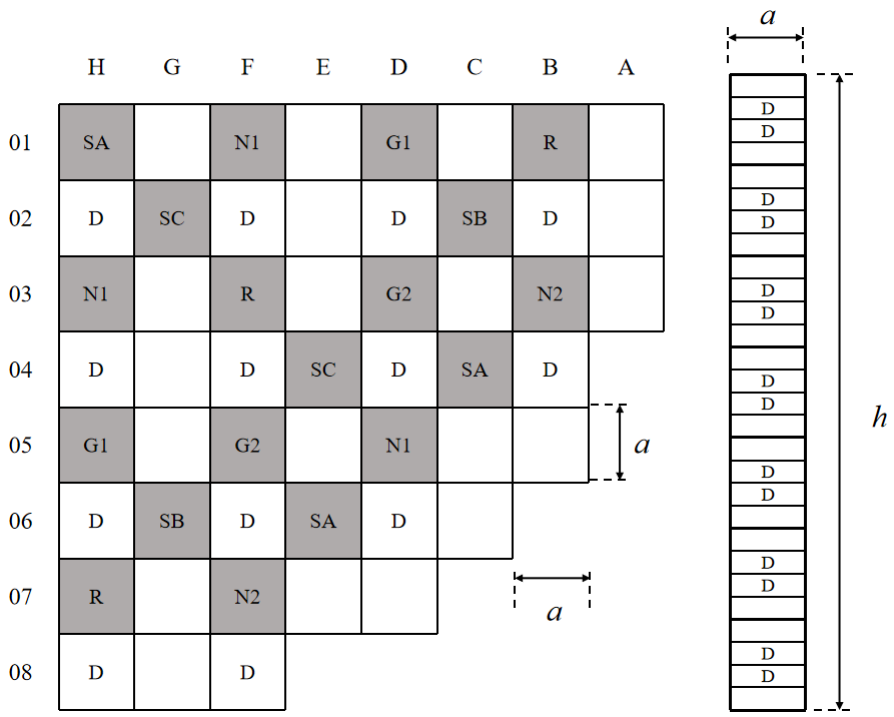


Fig. 6: A quarter of the core in the radial direction (white square: fuel assembly with SPNDs, grey square: fuel assembly with control rods, D: fuel assembly with neutron detectors) [13].

distribution, depending on the specific requirements. Specifically, the noise term can be mathematically represented as follows:

$$e = Y_o \cdot N(0, \sigma, Y_o) \text{ or } Y_o \cdot U(-\delta, \delta, Y_o)$$

In this context, the vector $N(0, \sigma, Y_o)$ denotes a collection of random variables that follow a normal distribution. These random variables have the same dimensions as Y_o and are characterized by a mean of 0 and a standard deviation of σ. Similarly, the vector $U(-\delta, \delta, Y_o)$ represents a set of random variables that are uniformly distributed within the interval $[-\delta, \delta]$. These variables also have the same dimensions as Y_o.

Considering the prevalence of Gaussian noise in real-world nuclear reactors, our subsequent analysis primarily focuses on evaluating the performance of the algorithm specifically under Gaussian noise conditions.

Software implementation We implemented our software integrated into the RODT platform with the modules as follows:

- rodtPro: This module implements data cleaning and preprocessing techniques based on domain expertise.
- rodtROM: This module constructs ROM through techniques like POD, SVD and RB.
- rodtML: This module focuses on surrogate modelling and inverse model optimization. It also includes hyperparameter optimization techniques.
- rodtPost: This is a post-evaluation module that defines convergence metrics and conducts comparative analyses using evaluation methodologies derived from peer-reviewed literature.
- rodtUI: This module is responsible for visualizing the results, providing an intrinsic review of the methods, showcasing the optimized hyperparameters, and designing user-specific visualizations.

B. Practice Metrics for Inverse Model Evaluation

To evaluate the performance of the considered algorithms above, we design a series of numerical experiments using nuclear reactor data produced by the intrusive program with added noise. We carefully select metrics relating to the accuracy, and finally define the normalized reconstruct field L_2 error (E_2) and L_∞ error (E_∞) to evaluate the performance of the comparative algorithms.

The reconstructed field L_2 error is defined in Equation 4.21 to evaluate the average estimation of the field, where $\| \bullet \|_2 := \left(\sum_i (\bullet_i)^2 \right)^{1/2}$ denotes the L_2-norm.

$$E_2 := \left\| \boldsymbol{Q} F^{\mathrm{ML}}(\boldsymbol{\mu}^*) - \Phi_{\text{true}} \right\|_2 / \| \Phi_{\text{true}} \|_2 \quad (4.21)$$

The reconstructed field L_∞ error is defined in Equation 4.22 to evaluate the worst estimation of the field, where $\| \bullet \|_\infty := \max_i |\bullet_i|$ denotes the L_∞-norm.

$$E_\infty := \left\| \boldsymbol{Q} F^{\mathrm{ML}}\left(\boldsymbol{\mu}^*\right) - \Phi_{\text{true}} \right\|_\infty / \left\| \Phi_{\text{true}} \right\|_\infty \quad (4.22)$$

We should also note that the reconstructed observation L_2 error and L_∞ error can be generated in the same manner.

V. NUMERICAL RESULTS

In this section, we conduct experiments to evaluate the performance of parameter identification and state estimation for the algorithms we proposed. The evaluation is carried out on a test dataset consisting of 40 specimens from HPR1000, as described in Section IV. All the proposed inverse modelling approaches are utilized in the experiments, and it is important to mention that the cost function employed remains consistent with Equation 2.12. It should be noted that we solely focused on testing the CSDE method and did not include the CS method. Furthermore, we introduce the proposed accuracy metric illustrated in Section IV B for assessing the performance.

A. Accuracy and Stability Analysis of the Parameter Identification Phase

In this section, we conducted tests to evaluate the accuracy of parameter identification. Initially, we assessed the effectiveness of the proposed hybrid methodology, Coarse-Fine-Grid Search, as discussed in Section III F, using the KNNLHS method as an illustrative example. To visualize this progress, we generated scatter plots depicting the predicted values of the parameters S_t and P_w, which were identified as the two dominant parameters in previous research [15], see Figure 7 for six cases with different input parameter settings.

Figure 7 provides visualizations of the local search space in inverse algorithms. The figures in the first row depict scenarios with varying P_w values, ranging from relatively low to moderate to relatively high. Correspondingly, the figures in the second row illustrate cases with varying S_t values, again encompassing relatively low, moderate, and relatively high values. These figures effectively showcase the considerable improvements achieved through the KNNLHS method, particularly in the simultaneous optimization of both P_w and S_t. The visualizations underscore the method's effectiveness in yielding enhanced optimization outcomes.

In general, the parameter prediction deviation is demonstrated in Figures 8 over the test set. Due to space limitations, we only chose KNN, KNNLHS, FSA for illustration, and tested on the clean and noised ($\sigma = 5\%$) observation data. Hybrid with the LHS algorithm, the KNN prediction for parameter identification becomes more accurate, and the prediction deviation for FSA algorithm is concentrated on a relatively low interval, thus proving its excellent accuracy. In general, the accuracy of burnup, the power level and control rod step are all acceptable by the engineering point of view.

B. Comparison of the Convergence Rate among Metaheuristic Algorithms

The convergence rate is a crucial criterion for evaluating algorithms which are integrated with the generation design, thus in this section, we compared the different metaheuristic algorithms and their hybrids. Their convergence performance in 50 iterations (generation) with the clean, noised ($\sigma = 1\%$ and $\sigma = 5\%$) observation data are tested in Figure 9.

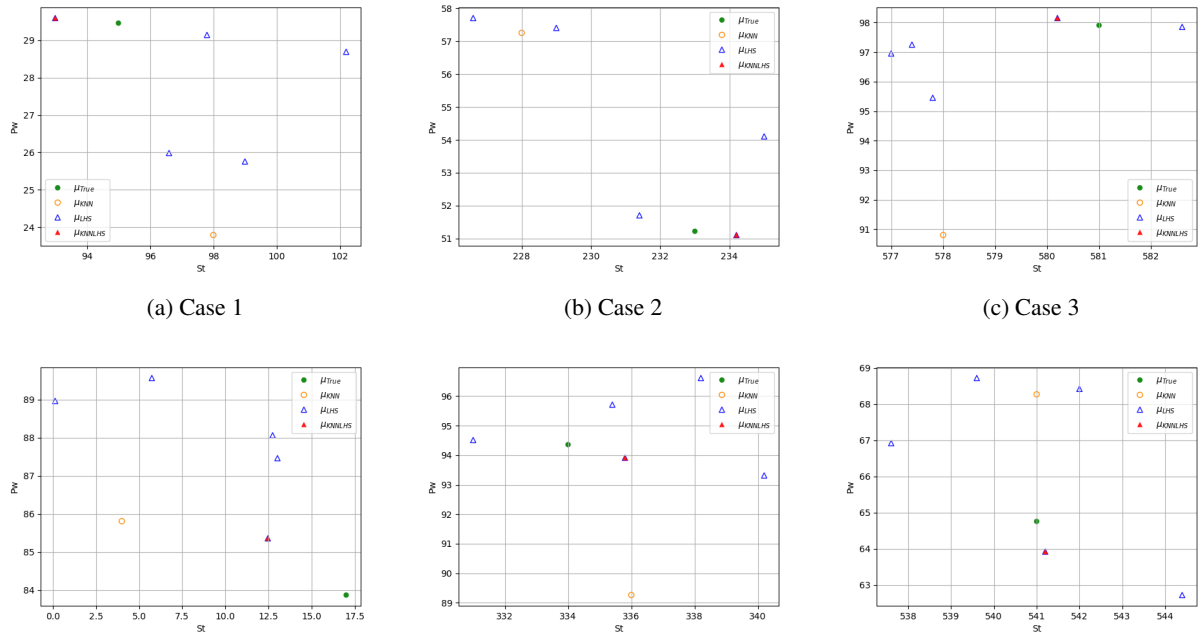
When the observation data is clean, the FSA algorithm exhibits the fastest convergence rate, followed by the ADE algorithm. Moreover, when integrated within an SVC coarse-grid search, both the DE and CSDE algorithms demonstrate the ability to escape local optima in the early stages while incurring a relatively low cost function, thereby surpassing their standalone performance.

Among the two population-based algorithms, DE outperforms CSDE. This can be attributed to two factors: (i) the adoption of the best mutation method for DE, where we use the optimal mutation strategy for DE, while the classical strategy; (ii) the relatively straightforward nature of our problem, wherein the inverse problem lacks multiple optima. Consequently, the CS step in each generation of CSDE may lead to unnecessary exploration of parameter space that is distant from the optimal solution.

In the presence of noise-contaminated observation data, FSA continues to exhibit superior convergence capabilities. However, its advantage diminishes as the level of Gaussian noise reaches 5%, eventually leading to a decline in advantage. Notably, the ADE and ACSDE algorithms fail to deliver satisfactory results under noisy conditions. This is primarily due to the erroneous predictions of SVC, which further leads to the inadequate initial guesses provided to the population-based algorithms.

C. Evaluation of State Estimation Phase by Accuracy

In this section, we evaluate various methods in the state estimation phase by accuracy. First, we compared the accuracy and stability under two metrics, namely the relative error in L_2 norm defined by Equation 4.21 and in $L\infty$ norm defined by Equation 4.22 for the field Φ, which is presented in Table 2. The 'Mean' values in the tables represent the average accuracy across 40 tested specimens, while the 'STD' values indicate the standard



(a) Case 1 (b) Case 2 (c) Case 3

(d) Case 4 (e) Case 5 (f) Case 6

Fig. 7: Distribution of generated candidate parameters $\boldsymbol{\mu}^1 = S_t$ and $\boldsymbol{\mu}^3 = P_w$ with hybrid algorithms. The green solid points are the true values while the yellow hollow points stand for the candidate parameter generated by the mere KNN algorithm. The blue hollow triangles represent the five optimal samplings by LHS around the initial guess given by the KNN, evaluated in the observation space. The red solid points represent the output of the algorithms, which is the best among the five optimal samplings.

deviation of errors over the same set of specimens. Notably, the optimal values are indicated in bold.

It is observed that, generally, the integration of the LHS algorithm enhances the accuracy of the KNN model. This improvement can be attributed to the LHS sampling approach, which mitigates the discontinuity issue associated with KNN predictions by sampling around the initial guess.

Furthermore, the NN method experiences a decline in accuracy when confronted with noise-contaminated observation backgrounds. This can be attributed to its reliance on gradients, rendering it less effective in our discrete case.

To visualize the performance of the inverse algorithms directly, we randomly select one specimen from the test dataset. Figures (11; 12) depict the relative L_2 error of the field prediction on the 11th vertical level. Readers are referred to Figure 10a for the true field value.

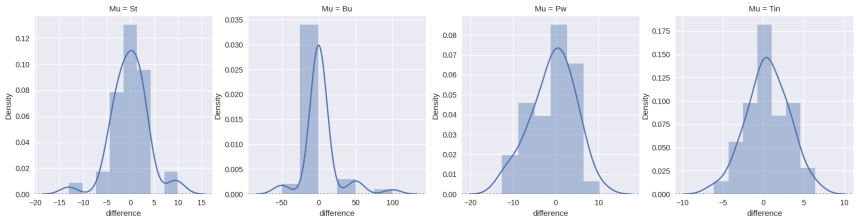
To fully demonstrate the reconstruction accuracy of our approach, we also examined cases where the reconstruction was relatively poor, especially around control rod interfaces. The reconstruction of the physical fields in locations near control rod interfaces is challenging, as the behaviour in these regions is not easily captured with high fidelity. Readers are referred to Figure 10b for the true field value on this axis, and Figures (13; 14) depict the prediction error of various methods on this axis.

However, our results also showed that even in such difficult cases, our methods can achieve good reconstruction for the most part. While some techniques performed worse around the interfaces, the majority of our proposed methods still facilitated reconstruction quality that is comparable to other areas across the domain. Overall, this validation indicates that the proposed approaches hold promise for representing complex reactor physics behaviours with suitable accuracy, even when challenges arise locally due to design complexities like control rod insertion points.

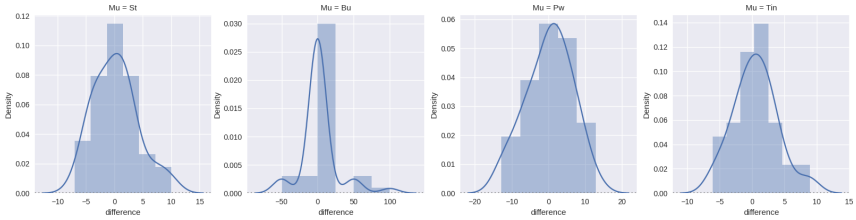
D. Evaluation of State Estimation Phase by Time Cost

In this section, we showcase an overview of the time costs (we take the mean of the 40 specimens) associated with each proposed algorithm in the state estimation phase, which can be found in Table 3. Optimal values are highlighted in bold, and the second optimal was highlighted with underlining.

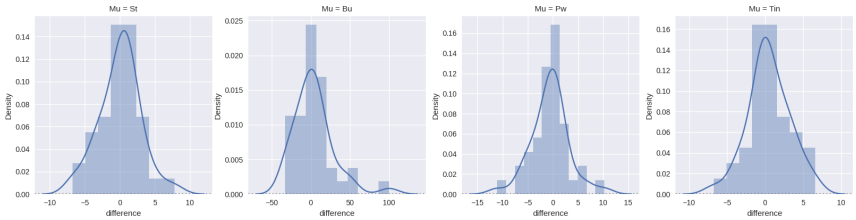
It is worth mentioning that the NN method demonstrates notable speed. On the other hand, the metaheuris-



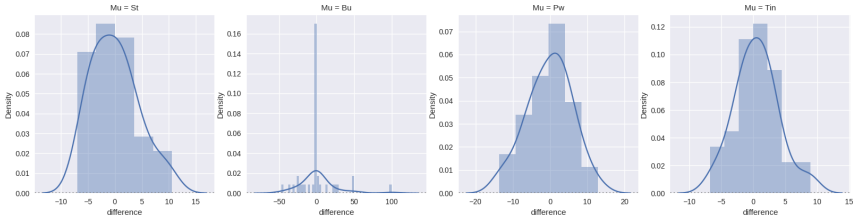
(a) Clean Observation (KNN)



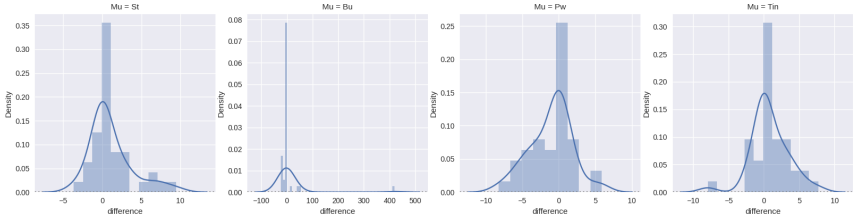
(b) $\sigma = 5\%$ (KNN)



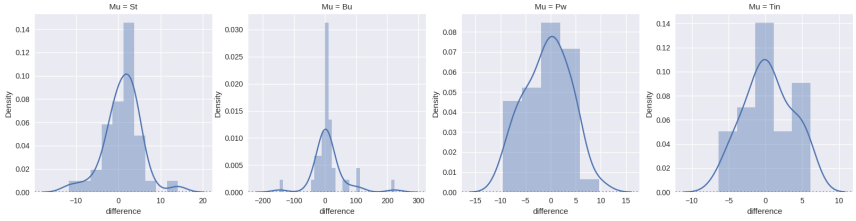
(c) Clean Observation (KNNLHS)



(d) $\sigma = 5\%$ (KNNLHS)



(e) Clean Observation (FSA)

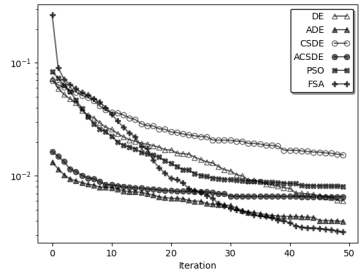


(f) $\sigma = 5\%$ (FSA)

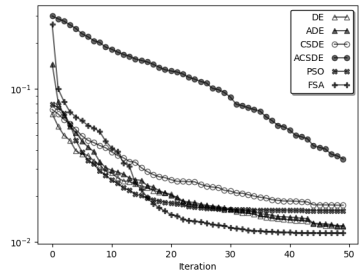
Fig. 8: The prediction deviation of μ with inverse model based on the clean or noised ($\sigma = 5\%$) observation data.

TABLE 2: Comparison of reconstructed power field errors using different algorithms.

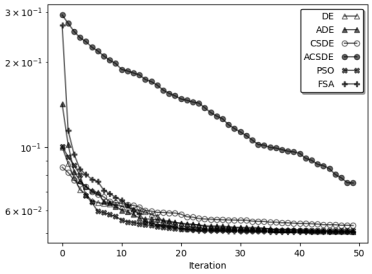
noise level	0	0.01	0.02	0.03	0.04	0.05
			Mean L_2			
KNN	1.0662E-02	1.1128E-02	1.1461E-02	1.1461E-02	1.3296E-02	1.3541E-02
LHS	4.0457E-02	4.0210E-02	4.0823E-02	4.0754E-02	4.1138E-02	4.1790E-02
ALHS	3.4294E-02	3.4510E-02	3.4837E-02	3.4865E-02	3.5218E-02	3.7221E-02
DE	5.3219E-03	6.6446E-03	7.8485E-03	9.2407E-03	**1.1051E-02**	1.3777E-02
ADE	**3.8119E-03**	6.8077E-03	7.8185E-03	9.1585E-03	1.1134E-02	1.3526E-02
CSDE	1.7029E-02	1.6581E-02	1.7011E-02	1.7390E-02	1.8874E-02	2.0011E-02
ACSDE	7.6471E-03	4.4970E-02	4.5157E-02	4.6044E-02	4.7066E-02	4.8563E-02
PSO	1.0093E-02	1.1502E-02	1.3355E-02	1.4640E-02	1.6516E-02	1.7939E-02
FSA	3.8796E-03	**5.1561E-03**	**6.5906E-03**	**8.1864E-03**	1.0106E-02	**1.2991E-02**
NN	2.6737E-02	9.8060E-02	1.3113E-01	1.6029E-01	1.8785E-01	2.1606E-01
LHS2STEPS	3.1108E-02	3.1410E-02	3.1664E-02	3.2919E-02	3.3712E-02	3.4975E-02
KNNLHS	7.0615E-03	9.0129E-03	9.6693E-03	1.0514E-02	1.2406E-02	1.3408E-02
			STD L_2			
KNN	5.3359E-03	5.5301E-03	5.2201E-03	5.2201E-03	**5.5587E-03**	**5.7721E-03**
LHS	2.5494E-02	2.5404E-02	2.5938E-02	2.5916E-02	2.5778E-02	2.5591E-02
ALHS	2.3719E-02	2.3621E-02	2.3377E-02	2.3131E-02	2.2688E-02	2.2074E-02
DE	5.3195E-03	4.6991E-03	4.6245E-03	**4.6625E-03**	5.7638E-03	8.1052E-03
ADE	**3.1455E-03**	**4.6463E-03**	**4.5439E-03**	4.6656E-03	6.0923E-03	7.7264E-03
CSDE	9.7295E-03	7.3634E-03	7.3791E-03	7.2028E-03	7.1571E-03	7.0336E-03
ACSDE	5.2449E-03	2.2811E-02	2.4206E-02	2.4100E-02	2.3438E-02	2.2736E-02
PSO	1.2374E-02	5.3048E-03	6.9606E-03	6.2216E-03	6.6237E-03	7.2507E-03
FSA	6.2602E-03	3.5832E-03	3.5640E-03	3.7438E-03	4.6444E-03	6.0076E-03
NN	3.4667E-02	7.3152E-02	8.3294E-02	9.8081E-02	1.1251E-01	1.2140E-01
LHS2STEPS	2.1742E-02	2.1617E-02	2.1475E-02	2.1528E-02	2.1408E-02	2.0572E-02
KNNLHS	4.9637E-03	5.3025E-03	4.9676E-03	5.8222E-03	5.9288E-03	6.1111E-03
			Mean L_∞			
KNN	4.1213E-02	4.3991E-02	4.4840E-02	4.4840E-02	4.9788E-02	**4,8775E-02**
LHS	1.3126E-01	1.2567E-01	1.2326E-01	1.1998E-01	1.1904E-01	1,2479E-01
ALHS	1.0932E-01	1.1062E-01	1.1454E-01	1.1529E-01	1.1662E-01	1,2263E-01
DE	2.4302E-02	2.7539E-02	3.2492E-02	3.8239E-02	4.4672E-02	5,2324E-02
ADE	**1.5357E-02**	2.8079E-02	3.2408E-02	3.7967E-02	4.4269E-02	5,1787E-02
CSDE	5.8757E-02	6.0849E-02	6.3653E-02	6.5595E-02	6.9252E-02	7,2312E-02
ACSDE	3.1167E-02	1.4168E-01	1.4022E-01	1.4177E-01	1.4747E-01	1,4928E-01
PSO	4.6008E-02	4.6052E-02	5.2500E-02	5.6560E-02	6.2328E-02	6,7784E-02
FSA	1.6543E-02	**2.2407E-02**	**2.7837E-02**	**3.3938E-02**	**4.0081E-02**	4,9525E-02
NN	8.1834E-02	2.3886E-01	3.0443E-01	3.6030E-01	3.8488E-01	4,1158E-01
LHS2STEPS	1.0144E-01	1.0014E-01	1.0167E-01	1.0810E-01	1.1415E-01	1,1607E-01
KNNLHS	2.7072E-02	3.7594E-02	4.0961E-02	4.3185E-02	4.7107E-02	5,0309E-02
			STD L_∞			
KNN	2.7190E-02	3.0633E-02	2.8849E-02	2.8849E-02	2.8941E-02	2.9448E-02
LHS	8.7321E-02	8.2894E-02	7.8177E-02	7.6610E-02	7.7353E-02	8.1824E-02
ALHS	8.8403E-02	8.9334E-02	8.8679E-02	8.8367E-02	8.9894E-02	8.7995E-02
DE	2.2486E-02	1.8169E-02	1.9161E-02	1.9406E-02	2.4870E-02	3.2087E-02
ADE	**1.5143E-02**	1.8220E-02	1.9221E-02	2.0268E-02	2.5709E-02	3.0895E-02
CSDE	4.0726E-02	2.6558E-02	2.8031E-02	2.8521E-02	2.6937E-02	2.9098E-02
ACSDE	2.1241E-02	6.5362E-02	6.7719E-02	7.0399E-02	6.9278E-02	6.8662E-02
PSO	5.6853E-02	2.4197E-02	2.5349E-02	2.5845E-02	2.8182E-02	3.1462E-02
FSA	2.3181E-02	**1.5810E-02**	**1.5634E-02**	**1.6278E-02**	**2.0246E-02**	**2.6014E-02**
NN	9.1032E-02	1.6232E-01	1.8105E-01	1.8648E-01	2.0080E-01	1.9789E-01
LHS2STEPS	7.1699E-02	6.9753E-02	7.1235E-02	7.3554E-02	7.4717E-02	7.2363E-02
KNNLHS	2.2045E-02	2.6180E-02	2.4270E-02	2.8580E-02	2.9683E-02	3.2208E-02



(a) Clean Observation

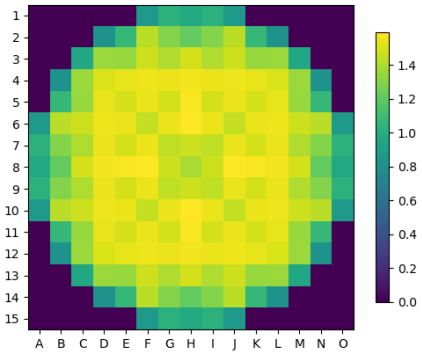


(b) $\sigma = 1\%$

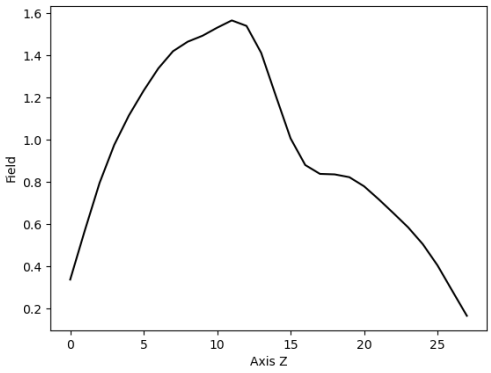


(c) $\sigma = 5\%$

Fig. 9: The convergence rate of the cost function (defined by the L_2 norm) with different metaheuristic optimizers. The x-axis represents the iteration (generation) and we set the maximum iteration (generation) to 50.



(a) The radial power distribution of the 11th axial plane



(b) The axial power distribution of the D01 fuel assembly

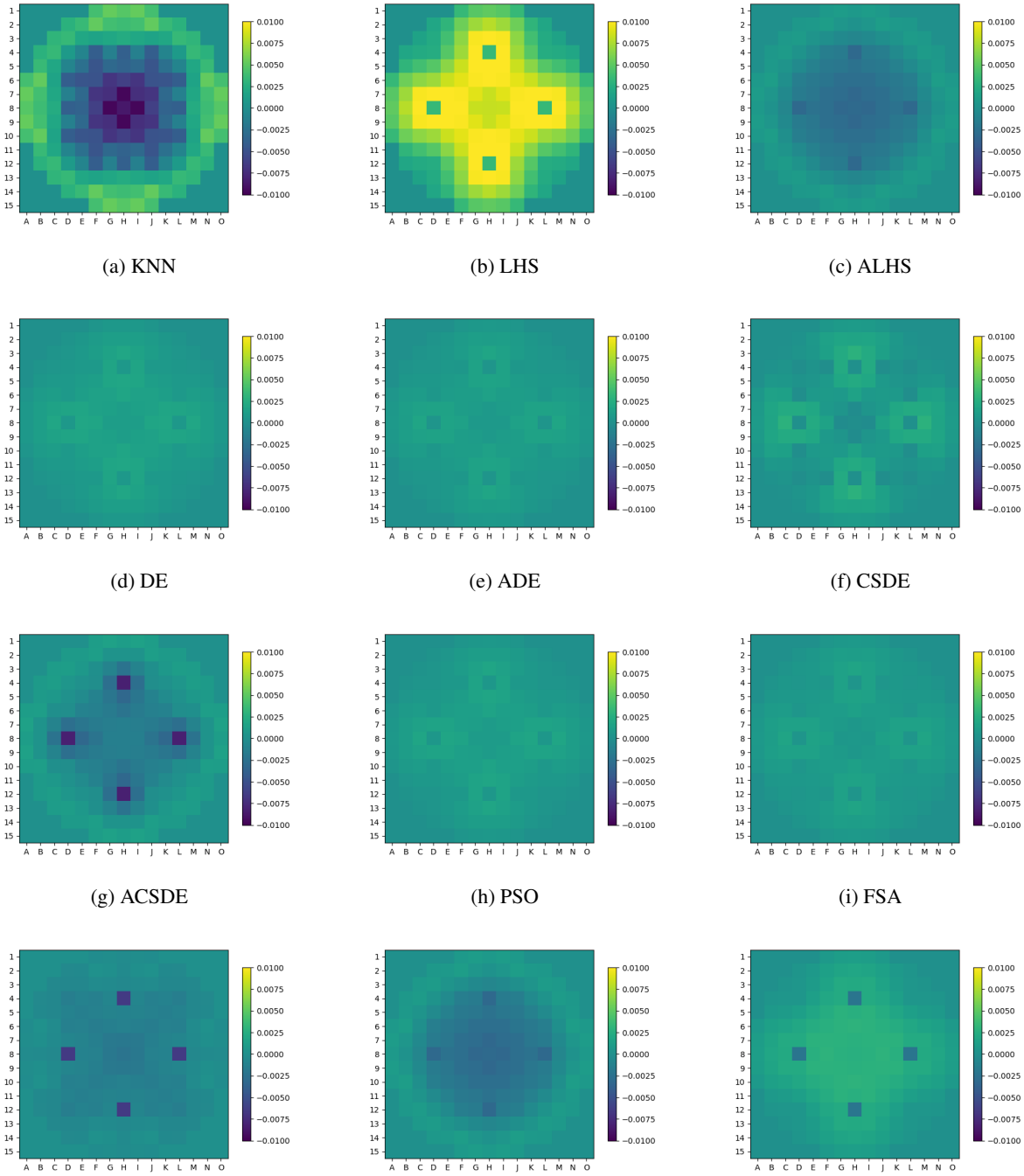
Fig. 10: The true value of the power distribution on one specimen in test dataset over the core in a realistic HPR1000 reactor.

tic algorithms exhibit relatively higher time costs. This is primarily attributed to the repeated computation of the cost function in Equation 2.12, which occurs 50 times for each population member over 50 iterations/generation cycles, resulting in a total of 2500 cost function evaluations. Moreover, these algorithms involve additional computations and sorting operations for information exchange and identification of the best solutions, contributing to the discrepancy in time cost.

Furthermore, it is important to note that the methods LHS, ALHS, and LHS2STEPS have approximately the same time cost. In algorithm LHS, we simply set the sampling number to 1000. In algorithms ALHS and LHS2STEPS, we set the first-stage sampling number to 750, and the second-stage sampling is repeated five times, with each repetition sampling number uniformly set to 50, adding up to 1000 in total. Model-based algorithms like KNN and its hybrid KNNLHS enjoy relatively low time costs, i.e., less than one second, which are acceptable in the industry.

E. A Comprehensive Evaluation of Various Algorithms

In the state estimation phase, the comparison of proposed algorithms in terms of time cost and accuracy is presented in Figure 15. The KNNLHS algorithm demonstrates a commendable level of accuracy, achieving solutions within 0.1 seconds for both clean and noise-contaminated observation backgrounds. Consequently, KNNLHS can be considered suitable for real-time online inverse modelling in the context of RODT. Moreover, in



(a) KNN (b) LHS (c) ALHS

(d) DE (e) ADE (f) CSDE

(g) ACSDE (h) PSO (i) FSA

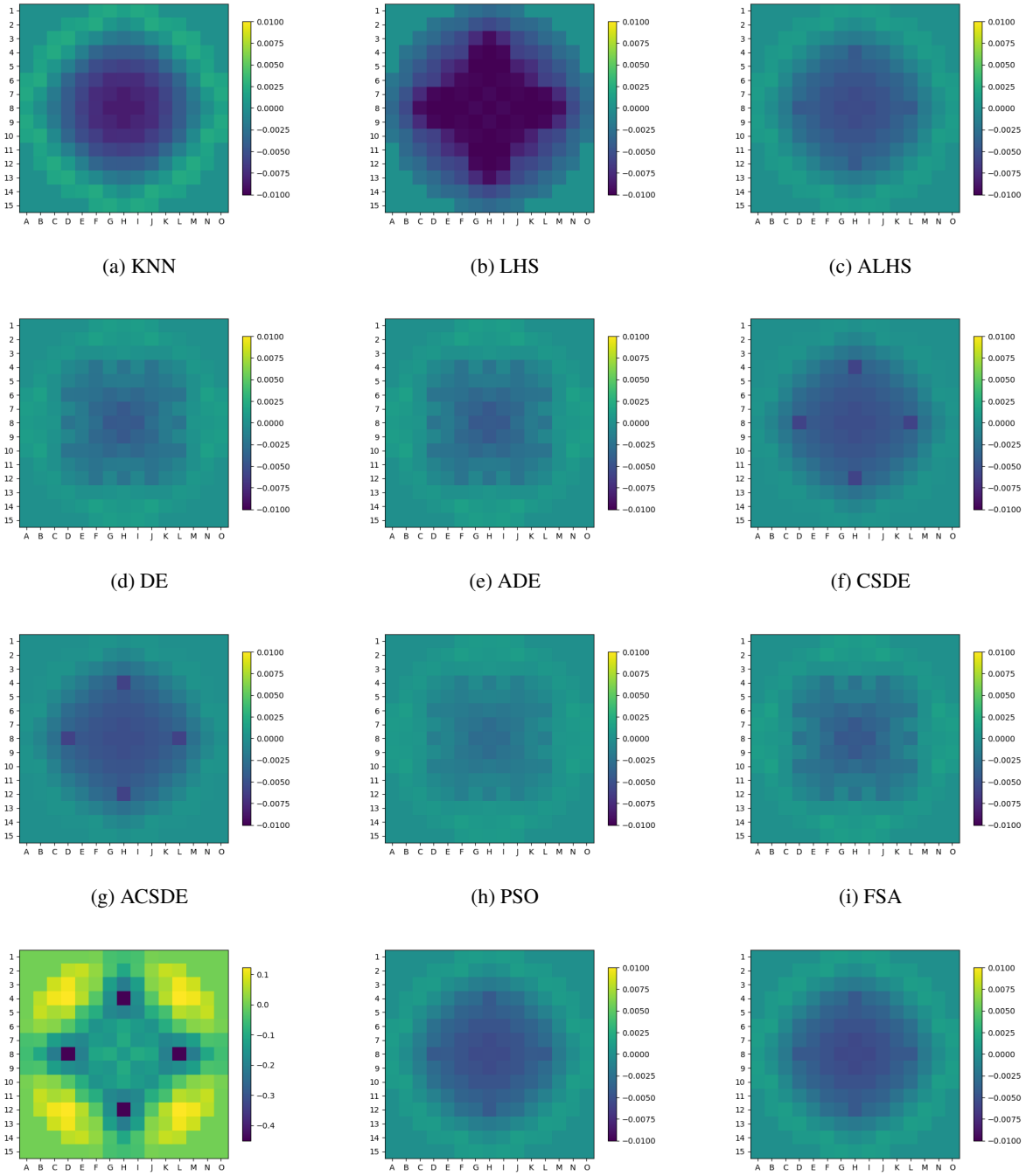
(j) NN (k) LHS2STEP (l) KNNLHS

Fig. 11: The reconstructed relative errors of radial power distribution on the 11th axial plane with clean observations.

scenarios where time constraints are not a limiting factor, the FSA, DE, and ADE algorithms also provide reasonably accurate solutions.

VI. CONCLUSION AND FUTURE WORKS

In this work, we developed and implemented a series of powerful algorithms into the RODT, aiming to address challenges in parameter identification and state estima-



(a) KNN (b) LHS (c) ALHS

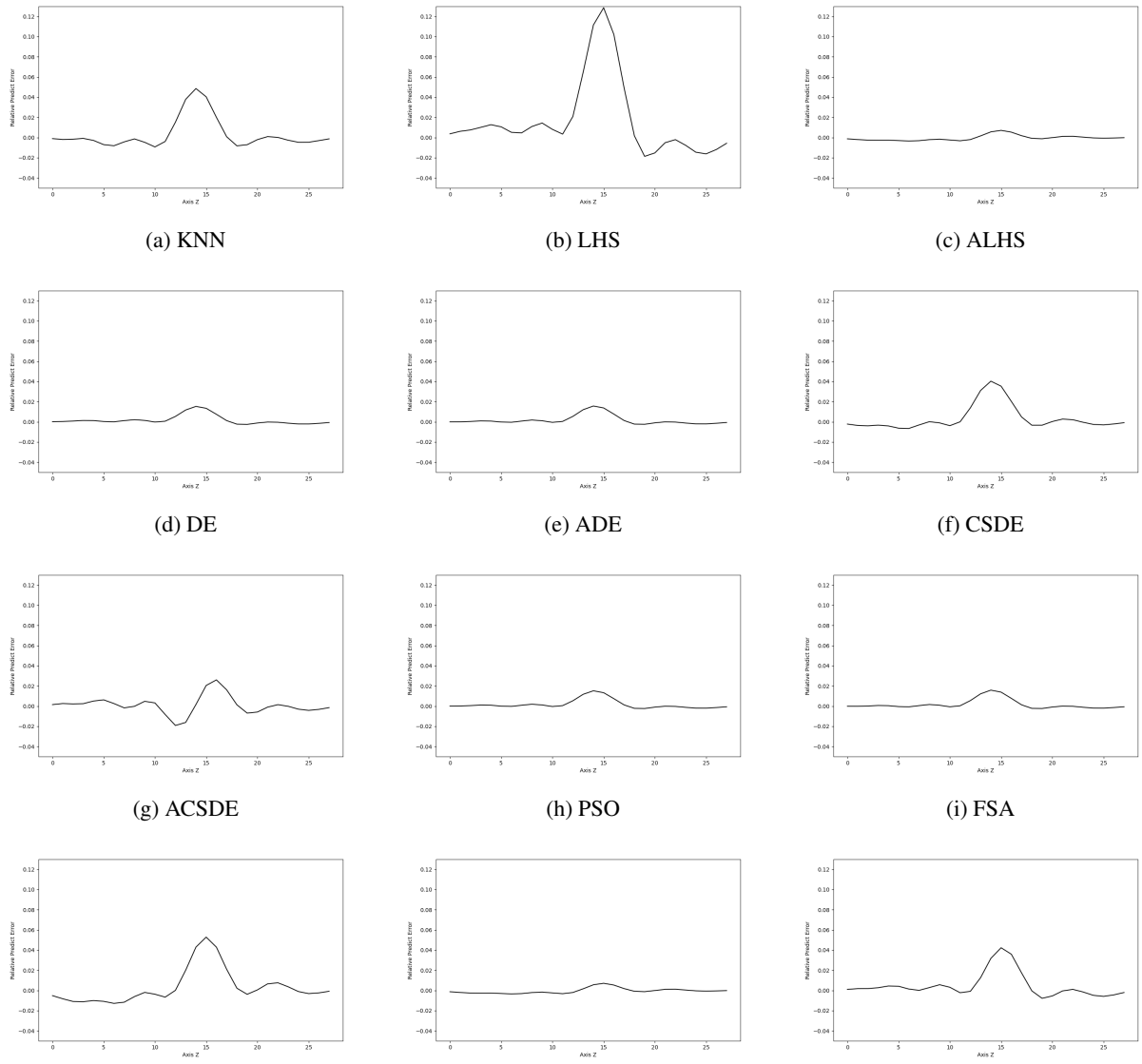
(d) DE (e) ADE (f) CSDE

(g) ACSDE (h) PSO (i) FSA

(j) NN (k) LHS2STEP (l) KNNLHS

Fig. 12: The reconstructed relative errors of radial power distribution on the 11th axial plane with observations of noise level $\sigma = 5\%$.

tion for nuclear reactor monitoring and optimization applications. Novel hybrid optimization algorithms including ADE, CSDE, ACSDE and KNNLHS were designed and implemented to effectively solve the inverse problems posed by the discontinuous surrogate forward modelling approach. Both deterministic and metaheuristic optimization methods were investigated for their applicability to nuclear engineering systems characterized by



(a) KNN (b) LHS (c) ALHS

(d) DE (e) ADE (f) CSDE

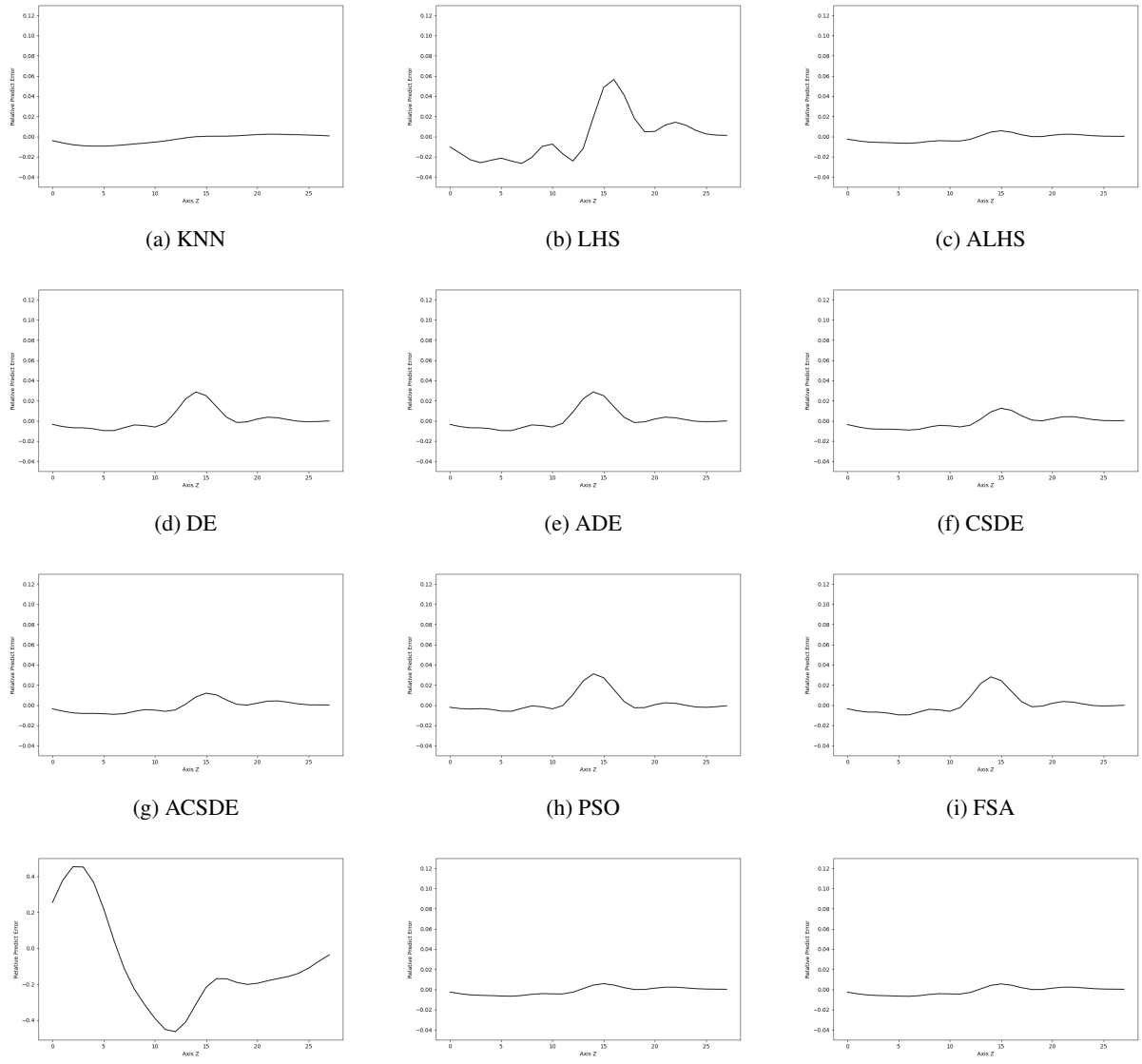
(g) ACSDE (h) PSO (i) FSA

(j) NN (k) LHS2STEP (l) KNNLHS

Fig. 13: The predicted relative error for the D01 fuel assembly with clean observations.

TABLE 3: Comparison of time cost for different algorithms.

noise level	0	0.01	0.02	0.03	0.04	0.05
KNN	6.3696E-03	3.1396E-03	3.1599E-03	3.1594E-03	3.1915E-03	3.1527E-03
LHS	7.3946E-01	7.3326E-01	7.2510E-01	7.3279E-01	7.2927E-01	7.2180E-01
ALHS	7.1437E-01	7.3674E-01	7.1443E-01	7.1613E-01	7.1730E-01	7.1050E-01
DE	3.0745E+00	2.9459E+00	2.8510E+00	2.9223E+00	2.8299E+00	2.7874E+00
ADE	2.9249E+00	2.9339E+00	2.8915E+00	2.9284E+00	2.9743E+00	3.0290E+00
CSDE	4.7069E+00	4.6679E+00	4.6691E+00	4.6531E+00	4.6615E+00	4.6227E+00
ACSDE	4.8054E+00	4.6563E+00	4.6374E+00	4.6886E+00	4.6303E+00	4.7069E+00
PSO	2.4404E+00	2.5169E+00	2.4969E+00	2.3198E+00	2.3551E+00	2.3151E+00
FSA	2.7848E+00	2.8079E+00	2.7324E+00	2.7212E+00	2.7090E+00	2.6977E+00
NN	**4.1338E-04**	**4.2277E-04**	**4.0678E-04**	**4.0900E-04**	**4.0602E-04**	**4.0516E-04**
LHS2STEPS	7.0137E-01	7.0546E-01	7.0171E-01	7.0508E-01	7.0827E-01	7.0630E-01
KNNLHS	7.0515E-02	6.4439E-02	6.3790E-02	6.4109E-02	6.3136E-02	6.3097E-02



(a) KNN (b) LHS (c) ALHS

(d) DE (e) ADE (f) CSDE

(g) ACSDE (h) PSO (i) FSA

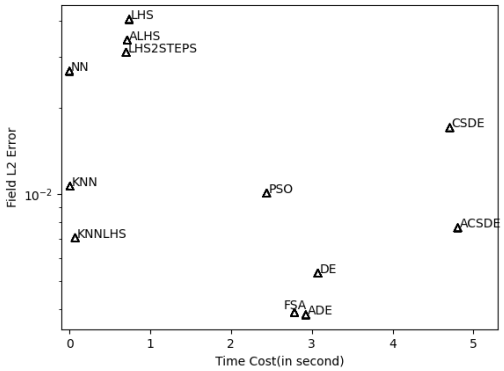
(j) NN (k) LHS2STEP (l) KNNLHS

Fig. 14: The predicted relative error for the D01 fuel assembly with observations of noise level $\sigma = 5\%$.

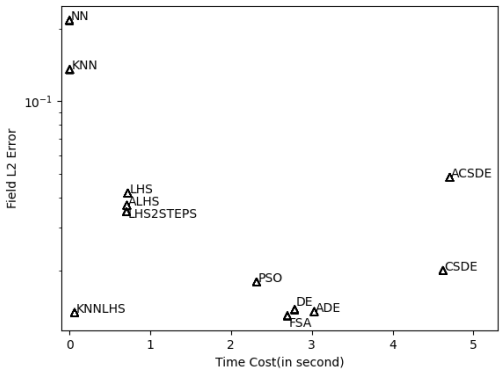
high-dimensional, non-linear relationships. The integration of coarse-grid and fine-grid searching strikes an optimal balance between computational efficiency and solution accuracy.

Through extensive experimentation and evaluation, we conducted a thorough comparative analysis of parameter identification accuracy, field reconstruction accuracy, and convergence profile. The FSA exhibits the fastest rate of convergence, and the proposed ADE method follows closely behind. Moreover, integrating the standard DE or CSDE algorithms within SVC coarse grid search framework enables these techniques to effectively escape local optima in early iterations while maintaining relatively low cost function values, surpassing their standalone implementations. Additionally, it is observed that pairing the LHS approach within the KNN surrogate model enhances accuracy by mitigating discontinuity issues through sampling locally around the initial guess. Thus, our approaches effectively address the challenge of discontinuity and non-differentiability in the surrogate forward mapping generated by KNN model. Conversely, NN method experiences declining accuracy when faced with noise-contaminated measurements, likely due to over-reliance on gradient information that proves less effective under such uncertainty.

Our algorithms demonstrate high computational effi-



(a) Cases with clean observations



(b) Cases with observations of noise level $\sigma = 5\%$

Fig. 15: Comaparation of the synthetic performance for different inverse algorithms.

ciency, with solutions obtained within one second, and as fast as $\leq$ 0.08 seconds for KNN, KNNLHS, and NN. They also exhibit robustness in handling noise-contaminated backgrounds. Our work contributes to realizing RODT, providing effective, efficient, and reliable support for real-world nuclear power plant operation and sensor data. These algorithms establish a promising foundation for advanced systems modelling and engineering decision support.

By specifically tailoring solutions to nuclear inverse and surrogate modelling challenges, this research advances the state-of-the-art RODT methodology. The improved parameter identification and state estimation now will facilitate the optimization of reactor monitoring, control and performance through data-driven DT applications. In the future, novel methods could be contributed to our RODT system and with the same methodology, we can support the development and implementation of these transformative technologies in other industries in need of real-time DTs.

Continued work validating across diverse systems and progressively integrating multi-physics modelling holds promise to fully realize RODTs' potential for transforming nuclear engineering practice with powerful real-time and decision-support capabilities. Realization of this potential will require ongoing progress, collaboration and open innovation across expertise areas.

COMPETING INTERESTS

The authors declare no competing interests.

ACKNOWLEDGMENTS

This research is partially sponsored by the Natural Science Foundation of Shanghai (No.23ZR1429300), the Innovation Funds of CNNC (Lingchuang Fund, Contract No. CNNC-LCKY-202234), and the Project of Nuclear Power Technology Innovation Center of Science Technology and Industry for National Defense (Contract No. HDLCXZX-2023-HD-039-02).

[1] F. Tao, H. Zhang, A. Liu, et al., Digital twin in industry: State-of-the-art. IEEE Transactions on industrial informatics **15**, 2405–2415 (2018).

[2] W.G. Michael, Virtually intelligent product systems: Digital and physical twins. 2019

[3] K.Y.H. Lim, P. Zheng, C.H. Chen, A state-of-the-art survey of digital twin: techniques, engineering product lifecycle management and business innovation perspectives. Journal of Intelligent Manufacturing **31**, 1313–1337 (2020).

[4] M.W. Grieves, J.H. Vickers, Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems. 2017

[5] M. Liu, S. Fang, H. Dong, et al., Review of digital twin about concepts, technologies, and industrial applications. Journal of Manufacturing Systems **58**, 346–361 (2021).

[6] M. Singh, E. Fuenmayor, E.P. Hinchy, et al., Digital twin: Origin to future. Applied System Innovation **4**, 36 (2021).

[7] E. Örs, R. Schmidt, M. Mighani, et al., A conceptual framework for ai-based operational digital twin in chemical process engineering. 2020 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC) 1–8 (2020).

[8] J. Kraft, S. Kuntzagk, Engine fleet-management: The use of digital twins from a mro perspective. 2017

[9] F. Chinesta, E. Cueto, E. Abisset-Chavanne, et al., Virtual, digital and hybrid twins: A new paradigm in data-based engineering and engineered data. Archives of Computational Methods in Engineering **27**, 105–134 (2018).

[10] M. Adams, X. Li, L. Boucinha, et al., Hybrid digital twins: A primer on combining physics-based and data analytics approaches. IEEE Software **39**, 47–52 (2022). doi:10.1109/MS.2021.3134042

[11] H. Song, M. Song, X. Liu, Online autonomous calibration of digital twins using machine learning with application to nuclear power plants. Applied Energy **326**, 119995 (2022). doi:https://doi.org/10.1016/j.apenergy.2022.119995

[12] A.K. Sleiti, J.S. Kapat, L. Vesely, Digital twin in energy industry: Proposed robust digital twin for power plant and other complex capital-intensive large engineering systems. Energy Reports **8**, 3704–3726 (2022). doi:https://doi.org/10.1016/j.egyr.2022.02.305

[13] H. Gong, S. Cheng, Z. Chen, et al., Data-enabled physics-informed machine learning for reduced-order modeling digital twin: Application to nuclear reactor physics. Nuclear Science and Engineering **196**, 1–26 (2022). doi:10.1080/00295639.2021.2014752

[14] X. Li, S. Wang, W. Zhou, et al., in *2019 11th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, Research on fault diagnosis algorithm based on convolutional neural network. Vol. 1, 2019, pp. 8–12. doi:10.1109/IHMSC.2019.00010

[15] H. Gong, S. Cheng, Z. Chen, et al., An efficient digital twin based on machine learning svd autoencoder and generalised latent assimilation for nuclear reactor physics. Annals of Nuclear Energy **179**, 109431 (2022). doi:10.1016/j.anucene.2022.109431

[16] H. Gong, T. Zhu, Z. Chen, et al., Parameter identification and state estimation for nuclear reactor operation digital twin. Annals of Nuclear Energy **180**, 610041 (2023). doi:10.1016/j.anucene.2022.109497

[17] A.C. Antoulas, D.C. Sorensen, S. Gugercin, A survey of model reduction methods for large-scale systems. 2000

[18] R. Arcucci, L. Mottet, C. Pain, et al., Optimal reduced space for variational data assimilation. Journal of Computational Physics **379**, 51–69 (2018). doi:10.1016/j.jcp.2018.10.042

[19] J.P. Argaud, B. Bouriquet, F. de Caso, et al., Sensor placement in nuclear reactors based on the generalized empirical interpolation method. J. Comput. Phys. **363**, 354–370 (2018).

[20] P. Benner, M. Ohlberger, A. Cohen, et al., *Model reduction and approximation: theory and algorithms*, (SIAM, 2017)

[21] P. Benner, S. Gugercin, K. Willcox, A survey of projection-based model reduction methods for parametric dynamical systems. SIAM Review **57**, 483–531 (2015). arXiv:https://doi.org/10.1137/130932715, doi:10.1137/130932715

[22] M.G. Kapteyn, D.J. Knezevic, K.E. Willcox, Toward predictive digital twins via component-based reduced-order models and interpretable machine learning. 2020

[23] D. Hartmann, M. Herz, U. Wever, *Model Order Reduction a Key Technology for Digital Twins*, (Springer International Publishing, Cham, 2018), pp. 167–179. doi:10.1007/978-3-319-75319-5$_8$

[24] J. Hammond, R. Chakir, F. Bourquin, et al., Pbdw: A non-intrusive reduced basis data assimilation method and its application to an urban dispersion modeling framework. Applied Mathematical Modelling **76**, 1–25 (2019). doi:https://doi.org/10.1016/j.apm.2019.05.012

[25] A. Rasheed, O. San, T. Kvamsdal, Digital twin: Values, challenges and enablers from a modeling perspective. IEEE Access **8**, 21980–22012 (2020). doi:10.1109/ACCESS.2020.2970143

[26] E. Nadal, F. Chinesta, P. Díez, et al., Real time parameter identification and solution reconstruction from experimental data using the proper generalized decomposition. Computer Methods in Applied Mechanics and Engineering **296**, 113–128 (2015). doi:https://doi.org/10.1016/j.cma.2015.07.020

[27] V. Harish, A. Kumar, Reduced order modeling and parameter identification of a building energy system model through an optimization routine. Applied Energy **162**, 1010–1023 (2016). doi:https://doi.org/10.1016/j.apenergy.2015.10.137

[28] H. Fu, H. Wang, Z. Wang, Pod/deim reduced-order modeling of time-fractional partial differential equations with applications in parameter identification. Journal of Scientific Computing **74**, 220–243 (2016).

[29] Q. Ding, Y. Wang, Z. Chen, Parameter identification of reduced-order electrochemical model simplified by spectral methods and state estimation based on square-root cubature kalman filter. Journal of Energy Storage **46**, 103828 (2022). doi:https://doi.org/10.1016/j.est.2021.103828

[30] Y. Maday, O. Mula, A generalized empirical interpolation method: Application of reduced basis techniques to data assimilation. Springer INdAM Series **4**, 221–235 (2013). doi:10.1007/978-88-470-2592-9$_1$3

[31] Y. Maday, A.T. Patera, J.D. Penn, et al., A parameterized-background data-weak approach to variational data assimilation: formulation, analysis, and application to acoustics. International Journal for Numerical Methods in Engineering **102**, 933–965 (2015). doi:https://doi.org/10.1002/nme.4747

[32] P. Binev, A. Cohen, W. Dahmen, et al., Data assimilation in reduced modeling. SIAM/ASA J. Uncertain. Quantification **5**, 1–29 (2015).

[33] H. Gong, Data assimilation with reduced basis and noisy measurement: Applications to nuclear reactor cores. Ph.D. thesis, Sorbonne université (2018)

[34] R. Arcucci, L. Mottet, C. Pain, et al., Optimal reduced space for variational data assimilation. Journal of Computational Physics **379**, 51–69 (2019). doi:https://doi.org/10.1016/j.jcp.2018.10.042

[35] H. Gong, Z. Chen, Y. Maday, et al., Optimal and fast field reconstruction with reduced basis and limited observations: Application to reactor core online monitoring. Nuclear Engineering and Design **377**, 111113 (2021). doi:https://doi.org/10.1016/j.nucengdes.2021.111113

[36] F. Di Rocco, D.G. Cacuci, Sensitivity and uncertainty analysis of a reduced-order model of nonlinear bwr dynamics: I. forward sensitivity analysis. Annals of Nuclear Energy **148**, 107738 (2020). doi:https://doi.org/10.1016/j.anucene.2020.107738

[37] S. Peitz, S. Ober-Blöbaum, M. Dellnitz, Multiobjective optimal control methods for the navier-stokes equations using reduced order modeling. Acta Applicandae Mathematicae **161**, 171–199 (2018).

[38] P. Chen, A. Quarteroni, G. Rozza, Reduced order methods for uncertainty quantification problems. ETH Zurich, SAM Report **3**,.

[39] Y. Liu, X. Sun, N.T. Dinh, Validation and uncertainty quantification of multiphase-cfd solvers: A data-driven bayesian framework supported by high-resolution experiments. Nuclear Engineering and Design **354**, 110200 (2019). doi:https://doi.org/10.1016/j.nucengdes.2019.110200

[40] M. Braun, Reduced order modelling and uncertainty propagation applied to water distribution networks. 2019

[41] G. Carere, M. Strazzullo, F. Ballarin, et al., A weighted pod-reduction approach for parametrized pde-constrained optimal control problems with random inputs and applications to environmental sciences. Comput. Math. Appl. **102**, 261–276 (2021).

[42] N. Demo, G. Ortali, G. Gustin, et al., An efficient computational framework for naval shape design and optimization problems by means of data-driven reduced order modeling techniques. Bollettino dell'Unione Matematica Italiana **14**, 211–230 (2020).

[43] S.A. Renganathan, Koopman-based approach to nonintrusive reduced order modeling: Application to aerodynamic shape optimization and uncertainty propagation. AIAA Journal **58**, 2221–2235 (2020).

[44] N.T. Mücke, L.H. Christiansen, A.P. Engsig-Karup, et al., Reduced order modeling for nonlinear pde-constrained optimization using neural networks. 2019 IEEE 58th Conference on Decision and Control (CDC) 4267–4272 (2019).

[45] M. Heinkenschloss, D. Jando, Reduced order modeling for time-dependent optimization problems with initial value controls. SIAM J. Sci. Comput. **40**,.

[46] J.V. Aguado, D. Borzacchiello, C. Ghnatios, et al., A simulation app based on reduced order modeling for manufacturing optimization of composite outlet guide vanes. Advanced Modeling and Simulation in Engineering Sciences **4**, 1–26 (2017).

[47] L. Iapichino, S. Ulbrich, S. Volkwein, Multiobjective pde-constrained optimization using the reduced-basis method. Advances in Computational Mathematics **43**, 945–972 (2017).

[48] M.L. Zhang, Z.H. Zhou, Ml-knn: A lazy learning approach to multi-label learning. Pattern Recognit. **40**, 2038–2048 (2007).

[49] X. Wu, V. Kumar, J.R. Quinlan, et al., Top 10 algorithms in data mining. Knowledge and Information Systems **14**, 1–37 (2007).

[50] S. Zhang, X. Li, M. Zong, et al., Learning k for knn classification. ACM Trans. Intell. Syst. Technol. **8**, 1–19 (2017). doi:10.1145/2990508

[51] A. Tarantola, Inverse problem theory - and methods for model parameter estimation. 2004

[52] M. Frangos, Y. Marzouk, K. Willcox, et al., Surrogate and reduced-order modeling: a comparison of approaches for large-scale statistical inverse problems. Large-Scale Inverse Problems and Quantification of Uncertainty 123–149 (2010).

[53] P. An, Y. Ma, P. Xiao, et al., Development and validation of reactor nuclear design code corca-3d. Nuclear Engineering and Technology **51**, 1721–1728 (2019). doi:https://doi.org/10.1016/j.net.2019.05.015

[54] N. El-Sahlamy, M. Hassan, A. Khedr, et al., Study of rod ejection accident at hot zero power condition in a pwr using relap5. Progress in Nuclear Energy **144**, 104100 (2022). doi:https://doi.org/10.1016/j.pnucene.2021.104100

[55] Y. Maday, Reduced basis method for the rapid and reliable solution of partial differential equations. Proceedings oh the International Congress of Mathematicians **3**, 1255–1270 (2006).

[56] M.A. Grepl, Y. Maday, N.C. Nguyen, et al., Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations. ESAIM: Mathematical Modelling and Numerical Analysis **41**, 575–605 (2007).

[57] J.S. Hesthaven, G. Rozza, B. Stamm, et al., *Certified reduced basis methods for parametrized partial differential equations*, Vol. 590, (Springer, 2016)

[58] C. Eckart, G.M. Young, The approximation of one matrix by another of lower rank. Psychometrika **1**, 211–218 (1936).

[59] S. Arridge, P. Maass, O. Öktem, et al., Solving inverse problems using data-driven models. Acta Numerica **28**, 1–174 (2019).

[60] R. Ahuja, J. Orlin, Inverse optimization. Operations Research **49**, 771–783 (2001). doi:10.1287/opre.49.5.771.10607

[61] R.M. Lewis, V. Torczon, M.W. Trosset, Direct search methods: then and now. Journal of Computational and Applied Mathematics **124**, 191–207 (2000).

[62] J.C. Helton, F.J. Davis, Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems. Reliability Engineering & System Safety **81**, 23–69 (2003).

[63] R. Biedrzycki, J. Arabas, D. Jagodziński, Bound constraints handling in differential evolution: An experimental study. Swarm and Evolutionary Computation **50**, 100453 (2019).

[64] N.D. Lagaros, M. Kournoutos, N.A. Kallioras, et al., Constraint handling techniques for metaheuristics: a state-of-the-art review and new variants. Optimization and Engineering **24**, 2251–2298 (2023).

[65] M. Montemurro, A. Vincenti, P. Vannucci, The automatic dynamic penalisation method (adp) for handling constraints with genetic algorithms. Computer Methods in Applied Mechanics and Engineering **256**, 70–87 (2013).

[66] S. Das, P.N. Suganthan, Differential evolution: A survey of the state-of-the-art. IEEE Transactions on Evolutionary Computation **15**, 4–31 (2011). doi:10.1109/TEVC.2010.2059031

[67] R. Storn, K.V. Price, Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization **11**, 341–359 (1997).

[68] R. Eberhart, J. Kennedy, in *Proceedings of the IEEE international conference on neural networks*, Particle swarm optimization. Vol. 4, Citeseer, 1995, pp. 1942–1948

[69] S. Geman, D. Geman, Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. IEEE Transactions on Pattern Analysis and Machine Intelligence **6**, 721–741 (1984). doi:10.1109/TPAMI.1984.4767596

[70] S. Kirkpatrick, C. Gelatt, M. Vecchi, in *Readings in Computer Vision*, Optimization by simulated annealing. (Morgan Kaufmann, San Francisco (CA), 1987), pp. 606–615 doi:https://doi.org/10.1016/B978-0-08-051581-6.50059-3

[71] T. Guilmeau, E. Chouzenoux, V. Elvira, in *2021 IEEE Statistical Signal Processing Workshop (SSP)*, Simulated annealing: a review and a new scheme. 2021, pp. 101–105. doi:10.1109/SSP49050.2021.9513782

[72] V.P. Tran, G.T. Phan, V.K. Hoang, et al., Evolutionary simulated annealing for fuel loading optimization of vver-1000 reactor. Annals of Nuclear Energy **151**, 107938 (2021). doi:https://doi.org/10.1016/j.anucene.2020.107938

[73] H. Szu, R. Hartley, Fast simulated annealing. Phys. Lett. A **122**, 721–741 (1984).

[74] X.S. Yang, S. Deb, in *2009 World congress on nature & biologically inspired computing (NaBIC)*, Cuckoo search via lévy flights. Ieee, 2009, pp. 210–214

[75] A.K. Jain, J. Mao, K.M. Mohiuddin, Artificial neural networks: A tutorial. Computer **29**, 31–44 (1996).

[76] J. Brajard, A. Carrassi, M. Bocquet, et al., Combining data assimilation and machine learning to emulate a dynamical model from sparse and noisy observations: A case study with the lorenz 96 model. Journal of Computational Science **44**, 101171 (2020). doi:https://doi.org/10.1016/j.jocs.2020.101171

[77] Y. Yang, H. Gong, S. Zhang, et al., A data-enabled physics-informed neural network with comprehensive numerical study on solving neutron diffusion eigenvalue problems. Annals of Nuclear Energy **183**, 109656 (2023). doi:https://doi.org/10.1016/j.anucene.2022.109656

[78] M. Salah, J. Trinder, Support vector machines: Optimization and validation for land cover mapping using aerial images and lidar data. 2011

[79] B. Zhang, J. Guofan, Z. Jun, Towards automatic freeform optics design: coarse and fine search of the three-mirror solution space. Vol. 10, 2021

[80] J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization. Vol. 13, 2012

[81] X.S. Yang, *Nature-Inspired Metaheuristic Algorithms*, (, 2010)

[82] D. Leon Valido, A. Gonzalez, Mutations as levy flights. Scientific Reports **11**, 9889 (2021). doi:10.1038/s41598-021-88012-1

[83] A. Luque, A. Carrasco, A. Martín, et al., The impact of class imbalance in classification performance metrics based on the binary confusion matrix. Pattern Recognition **91**, 216–231 (2019). doi:https://doi.org/10.1016/j.patcog.2019.02.023

[84] C. Goutte, E. Gaussier, in *Advances in Information Retrieval*, A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. (Springer Berlin Heidelberg, Berlin, Heidelberg, 2005), pp. 345–359

[85] X. Li, Q. Liu, Q. Li, et al., 177 core nuclear design for hpr1000. Nucl. Power Eng. **40**, 0008–00012 (2019).

[86] P. An, Y. Ma, P. Xiao, et al., Development and validation of reactor nuclear design code corca-3d. Nuclear Engineering and Technology **51**, 1721–1728 (2019).