Time Series Prediction of Nuclear Thermal Propulsion Reactor Parameters Based on Latent Space Nonlinear Operator Learning

Authors: Zhao Pengcheng, Liu Zijing, Li Wei, Zeng Yiling, Mo Junyu

Date: 2025-08-19T00:00:00+00:00

Abstract

To address the issues of poor real-time performance and low computational efficiency in multi-physics coupling modeling for nuclear reactors, this study employs a temporal prediction method for temperature fields based on Latent Space Neural Operator (Latent-DeepONet). By constructing a lightweight "encoding-operator learning-decoding" architecture, temperature field data of 230,000–1,800,000 dimensions is compressed into a 100-dimensional latent space, an improved Deep Operator Network (DeepONet) is utilized to learn the dynamic evolution laws of the latent space, and a physics-constrained decoder is combined to achieve high-precision reconstruction. Validation based on a nuclear-thermal coupling benchmark dataset generated by the computational fluid dynamics software OpenFOAM demonstrates: in 40-second iterative predictions, the average relative error of the fuel region temperature field is below 1%, and the coolant temperature field error is below 0.5%; in 100-second long-step predictions, the maximum relative error is controlled within 0.5%. The iterative prediction model takes less than 200s per prediction, achieving over 500× acceleration compared to traditional CFD methods; long-step prediction takes 79.23s per prediction, which is less than the prediction step size, meeting real-time prediction requirements. This study provides a lightweight digital twin solution for transient condition simulation and safety control of nuclear reactors.

Full Text

Temporal Prediction of Parameters in Nuclear Thermal Propulsion Reactors via Latent Space Nonlinear Operator Learning

ZENG Yiling[1], MO Junyu[1], LI Wei[2*], LIU Zijing[1], ZHAO Pengcheng[1]
[1]School of Nuclear Science and Technology, University of South China, Hengyang 421200, China
[2]School of Resource Environment and Safety Engineering, University of South China, Hengyang 421200, China

Abstract

[**Background**] In nuclear thermal propulsion reactor engineering, real-time prediction of multi-physics temperature fields under transient operating conditions has long been hindered by the inherent computational inefficiency and high dimensionality of coupled neutronics-thermal-hydraulic simulations. Traditional computational fluid dynamics (CFD) methods, while accurate, struggle to meet real-time decision-making requirements due to their prohibitive computational costs. [**Purpose**] This study aims to develop a lightweight digital twin framework for achieving real-time, high-fidelity temperature field predictions in nuclear reactors, thereby enabling rapid safety assessment and control strategy optimization. [**Methods**] By constructing a lightweight "encoding-operator learning-decoding" architecture, temperature field data of 230,000–1.8 million dimensions are compressed into a 100-dimensional latent space. An improved Deep Operator Network (DeepONet) learns the dynamic evolution within this latent space, and a physics-constrained decoder enables high-precision reconstruction. [**Results**] Validation using a nuclear-thermal coupling benchmark dataset generated by the computational fluid dynamics software OpenFOAM demonstrates that in 40-second iterative predictions, the average relative error of the fuel region temperature field remains below 1%, while the coolant temperature field error stays under 0.5%. In 100-second long-step predictions, the maximum relative error is controlled within 0.5%. Each iteration of the predictive model consumes less than 200 seconds, achieving a 2–3 order-of-magnitude acceleration compared to traditional CFD methods. Long-step predictions require only 79.23 seconds—shorter than the prediction interval—thus satisfying real-time requirements. **Conclusions** This study provides a lightweight digital twin solution for transient condition simulation and safety control of nuclear thermal propulsion reactors.

Keywords: L-DeepONet; nuclear thermal propulsion reactor; real-time prediction; autoencoder; multiphysics coupling

---

Nuclear thermal propulsion reactors operate under extreme high-temperature and high-pressure conditions, where abnormal temperature variations affect coolant density and moderation capability, alter neutron absorption effects in fuel, and may increase cladding rupture risk, thereby threatening reactor safety. Traditional numerical simulation methods cannot solve the dynamic systems governed by partial differential equations in real-time due to prohibitive computational costs. Consequently, more efficient approaches are needed for reactor temperature parameter prediction and estimation.

Artificial intelligence-driven surrogate models for high-dimensional multi-physics numerical calculations have found widespread application in nuclear reactor engineering [?, ?, ?], offering solutions for real-time prediction of complex differential equation models. Deep Neural Networks (DNN) [?], as essential tools for universal function approximation in machine learning, have achieved remarkable progress in image processing, natural language processing (NLP), and engineering design optimization [?, ?, ?]. They can adapt to different initial conditions, boundary conditions, model parameters, and geometric domains [?]. While traditional DNNs learn mappings between finite-dimensional vector spaces, emerging neural operators approximate mappings between infinite-dimensional Banach spaces to learn differential equation operators. The Deep Operator Network (DeepONet), proposed in 2019 [?, ?], utilizes DNNs to construct surrogate models for partial differential equations and has demonstrated rapid inference and high generalization accuracy in nuclear reactor surrogate modeling [?], though its performance degrades with increasing system scale and complexity [?, ?]. To address this limitation, Vivek Oommen first combined DeepONet with autoencoders (AE) to simulate two-phase microstructure growth described by the Cahn-Hilliard equation [?], while Zhang et al. [?] further integrated AE with neural operators for high-dimensional stochastic problems. However, DeepONet trained in latent space still requires systematic evaluation of prediction accuracy and generalization capability. The L-DeepONet framework proposed by Kontolati et al. [?] has been tested for real-time prediction of highly nonlinear and multiscale systems in high-dimensional domains, demonstrating superior prediction accuracy and computational efficiency compared to existing techniques. Additionally, reactor parameter calculation and prediction must consider not only kinetic models but also complex boundary conditions and heat transfer factors [?, ?, ?], which neural operator training can effectively handle, indicating significant potential for real-time temperature prediction in nuclear thermal propulsion reactors.

This study first employs the CFD program OpenFOAM [?] to obtain high-fidelity temperature field snapshot data for a nuclear thermal propulsion reactor. Following the L-DeepONet framework, an AE model is trained to achieve latent space mapping of high-dimensional inputs/outputs. DeepONet is then trained in the latent space for temporal prediction, and the predictions are reconstructed to the high-dimensional space through AE decoding, thereby establishing a temporal prediction methodology for nuclear thermal propulsion reactor temperatures.

1.1 Latent Space Nonlinear Operator Temporal Prediction Model

L-DeepONet is a neural network architecture designed to learn nonlinear operators in complex physical systems. It utilizes an autoencoder to reduce high-dimensional data to a low-dimensional latent space, then trains a DeepONet model in this space to improve computational efficiency and prediction accuracy [?].

The method comprises two main steps: (1) performing nonlinear dimensionality reduction on input and output data x, y using an invertible technique, and (2) training a DeepONet model in the latent space and inverse-transforming predictions back to the original space. The autoencoder consists of an encoder ϕ_{encoder} that projects high-dimensional data to the latent space and a decoder ϕ_{decoder} that reconstructs from latent space to original space, with θ_{encoder} and θ_{decoder} representing their trainable parameters. During training, the autoencoder projects prediction samples back to the original space to evaluate accuracy in the original dimensional data. After training, the L-DeepONet can be directly applied for real-time prediction of complex differential equation models in physical systems.

Let $r = \phi_{\text{encoder}}(x)$ denote the dimensionality-reduced data, and G_θ represent the approximated latent operator with parameters θ on DeepONet. The optimal parameter set $\theta_{\text{ae}} = \{\theta_{\text{encoder}}, \theta_{\text{decoder}}\}$ is obtained by minimizing the loss function:

$$\mathcal{L}_{\text{ae}} = \|\phi_{\text{decoder}}(\phi_{\text{encoder}}(D)) - D\|_2^2$$

where $D \equiv \{x, y\}$ denotes the combined input-output dataset and $\|\cdot\|_2$ represents the standard Euclidean norm. Experiments were conducted on three autoencoder variants: single-hidden-layer vanilla AE, multi-layer AE (MLAE), and convolutional AE (CAE) with multiple hidden layers and convolutional layers. Results showed MLAE achieved optimal performance.

1.1.1 Latent Space Mapping

Dimensionality reduction not only accelerates DeepONet training but also reveals intrinsic data structures, helping the model learn more discriminative features and improving prediction efficiency and accuracy. Since dimensionality-reduced data must be restored to high-dimensional space after training and prediction (requiring an inverse mapping), and the goal is to enhance DeepONet operator accuracy and prediction efficiency rather than identify the most advantageous method, the autoencoder was selected for its architectural flexibility and inherent invertibility, as shown in [Figure 1: see original paper].

To train the autoencoder model AE_θ and optimize hyperparameters, including determining the optimal latent dimension d (where $d \ll D_y$), consider a timedependent PDE where d_x corresponds to input space dimensionality and m_s, m_t represent spatial and temporal discretization dimensions. To feed PDE outputs as image-like data into the autoencoder, the outputs are reshaped into different snapshots and merged into a single dataset $\{x, y\}$. The encoder and decoder are trained simultaneously by minimizing the loss function to obtain the optimal parameter set θ_{ae}.

1.1.2 Latent Space Nonlinear Operator

After AE training, dimensionality-reduced data $\{r_x, r_y\}$ are generated, and DeepONet G_θ approximates the latent representation mapping, where θ represents trainable parameters. As shown in [Figure 2: see original paper], DeepONet comprises two concurrent deep neural networks: a branch network encodes input r_x at spatial locations, while a trunk network inputs temporal coordinates $\{t_1, ..., t_m\}$ in the reduced space and evaluates PDE outputs. The operator for input realization x_1 can be expressed as:

$$G_\theta(x_1) = \sum_{k=1}^{p} b_k \cdot t_k$$

where b_k and t_k are output vectors from the branch and trunk networks' final hidden layers, respectively, and p is a hyperparameter controlling their sizes. DeepONet' s trainable parameters θ are obtained by minimizing:

$$\mathcal{L}_{\text{DeepONet}} = \frac{1}{N} \sum_{i=1}^{N} \| G_\theta(r_x^{(i)}) - r_y^{(i)} \|_2^2$$

After obtaining optimal parameters, the trained model can transform new input predictions back to the original space using the decoder ϕ_{decoder} to obtain the approximate full-dimensional output $\hat{y}_{\text{rec}}$.

[Figure 2: see original paper] L-DeepONet framework diagram

For model training and prediction, the computational configuration includes an Intel i7-12700FK processor, RTX4070/12G GPU, and 32GB RAM. OpenFOAM 8 was used to establish a multi-physics coupling model for the nuclear thermal propulsion reactor, generating coolant, fuel, and cladding temperature field data from 0–200s for training and prediction. To comprehensively validate model performance, two training/test set partition strategies were employed: (1) For coolant and fuel temperature field predictions, 10s of training data with a 2s sliding window were used to predict up to 40s iteratively. (2) For cladding temperature field prediction, the first 100s served as training data and 101s-200s as test data for one-shot long-step prediction.

Model validation metrics include maximum temperature error, mean error, relative error, error distribution contour plots, and discussion of error sources and spatial distribution patterns. For coolant validation, additional statistical quantities such as error standard deviation were analyzed.

1.2 Multi-Physics Coupling Model

The reactor studied in this work is a nuclear thermal propulsion reactor—a specialized nuclear propulsion device for aircraft. Due to high manufacturing costs, safety regulations, and cost constraints, full-operating-condition experimental data are difficult to obtain, making simulation data the primary data source. Therefore, this study relies on simulation data instead of experimental data.

1.2.1 Nuclear Thermal Propulsion Reactor Design

The simulation physical model adopts a small nuclear thermal propulsion reactor that operates under extreme high-temperature and high-pressure environments, necessitating close monitoring of reactor parameters and driving research on real-time transient prediction of core parameters [21].

As shown in [Figure 3: see original paper], the system comprises a diffuser, air-cooled solid-core reactor, and nozzle. Air is decelerated and compressed by the diffuser, flows subsonically through reactor channels to absorb fission heat, then expands and accelerates through the nozzle. Fuel elements employ a honeycomb arrangement with a total core height of 120 cm and fuel region radius of 36 cm.

[**Figure 3: see original paper**] Schematic diagram of nuclear thermal propulsion reactor engine

1.2.2 Multi-Physics Coupling Model

The small nuclear thermal propulsion reactor core consists of thousands of coolant flow channels in a honeycomb arrangement. A simplified CFD approach treats the entire region as a continuum and solves velocity-pressure distributions via a porous media model. The governing equations for single-phase compressible flow in porous media are:

Mass conservation:

$$\frac{\partial}{\partial t}(\gamma\rho) + \nabla \cdot (\rho \mathbf{U}) = 0$$

Momentum conservation:

$$\frac{\partial}{\partial t}(\rho \mathbf{U}) + \nabla \cdot (\rho \mathbf{U}\mathbf{U}) = -\nabla P + \nabla \cdot \quad + \mathbf{S}_{\mathbf{U}}$$

Energy conservation:

$$\frac{\partial}{\partial t}(\rho C_p T) + \nabla \cdot (\rho C_p T \mathbf{U}) = \nabla \cdot (k \nabla T) + Q'''$$

where ρ is fluid density, $\mathbf{U}$ is superficial velocity, P is pressure, C_p is specific heat at constant pressure, Q''' is volumetric heat source, and γ is porosity representing the fluid volume fraction.

Reactor evolution is calculated using a point reactor model:

$$\frac{dn(t)}{dt} = \frac{k_{\text{eff}} - 1 - \beta}{l} n(t) + \sum_{i=1}^{6} \lambda_i C_i(t)$$

$$\frac{dC_i(t)}{dt} = \frac{\beta_i}{l} n(t) - \lambda_i C_i(t)$$

where k_{eff} is effective multiplication factor, β is delayed neutron fraction, l is neutron lifetime, $n(t)$ is neutron population at time t, λ_i is decay constant for delayed neutron precursor group i, and $C_i(t)$ is precursor concentration.

For heat transfer, a conjugate heat transfer (CHT) model is employed:

$$q'' = -k_f \left.\frac{\partial T}{\partial n}\right|_f = -k_s \left.\frac{\partial T}{\partial n}\right|_s$$

where subscripts f and s denote fluid and solid, respectively, and q'' represents heat flux continuity at fluid-solid interfaces.

2.1 Temperature Field Snapshot Acquisition

Design parameters for the nuclear thermal propulsion reactor model are detailed in . Commercial software STAR-CCM++ was used to generate structured meshes, as shown in [Figure 4: see original paper], with approximately 230,000 coolant cells, 1.8 million fuel region cells, and 370,000 cladding cells. For the coolant region, the entire core channel was modeled as a single domain with triangular meshes to ensure convergence.

**** Key design parameters of the nuclear thermal propulsion reactor

OpenFOAM solved the transient from reactor startup to 200s, recording snapshots every second with a total computational cost of 31 hours.

2.2 Autoencoding and Decoding

Using coolant temperature field prediction as an example, the autoencoder employs a three-layer fully connected architecture: the encoder compresses input data to latent space via 1024$→512→$$d_{\text{latent}}$ (ReLU activation), while the decoder uses a symmetric topology for reconstruction. Training utilized the Adam optimizer with a fixed learning rate of 1×10^{-3} and MSE loss, accelerated by GPU.

[**Figure 5: see original paper**] shows the MSE loss curve during training, revealing rapid convergence in epochs 1–10, steady optimization in epochs 10–

30, and stabilization after 20 epochs. To ensure topological feature integrity for neural operator prediction, a relatively high latent space dimension is required. Comparative experiments ([Figure 6: see original paper]) demonstrate that latent dimension d_{latent} exhibits nonlinear relationships with information retention and computational efficiency. At $d_{\text{latent}} = 100$, the model achieves optimal balance between performance and computational cost ($ $57s), making it the final configuration. With this dimension, normalized mean squared error (NMSE) drops below 0.0005, establishing a high-fidelity latent representation for subsequent temporal prediction.

2.3 DeepONet Deep Operator Network

DeepONet comprises Branch Net and Trunk Net sub-networks. Branch Net' s input/output dimensions align with the AE latent space ($d_{\text{latent}} = 100$) with one hidden layer ($d_{\text{latent}} \to 128 \to d_{\text{latent}}$, ReLU activation). Trunk Net encodes spatiotemporal coordinates to generate basis functions interacting with Branch Net outputs, also with one hidden layer ($2 \to 128 \to d_{\text{latent}}$, GELU activation), establishing nonlinear mapping to the next time step. Training used a fixed learning rate of 1×10^{-4}, Adam optimizer, and MSE loss, converging stably after 400 epochs ([Figure 7: see original paper]) without gradient explosion or overfitting.

The iterative prediction strategy employed the first 10s of coolant temperature data as training set, then predicted 11s–12s as test set. After obtaining these predictions, data from 3s–12s became the new training set for predicting 13s–14s, and so on, continuously using the most recent 10s to predict the next 2s until 40s. This dynamic updating of local physical field features enables long-term temporal extrapolation.

3.1 Coolant Temperature Field Prediction Analysis

The model employs a pure data-driven architecture: AE reduces dimensionality ($d_{\text{latent}} = 100$), DeepONet performs nonlinear prediction in latent space, and the decoder reconstructs physical space predictions. Total training time on an NVIDIA GeForce RTX 4070 was 130.48s—386× faster than OpenFOAM' s 14-hour solution.

Error distributions for 5s–40s are shown in [Figure 8: see original paper] and [Figure 9: see original paper]. Errors exhibit quasi-normal distribution with monotonic temporal increase. Absolute errors remain below 2K before 20s, gradually rising to a maximum of 3.17K at 40s. The mean relative error is 0.42%, maximum relative error 0.93%, and standard deviation at 40s is 0.289K. Both mean and standard deviation accumulate and amplify during recursive prediction, following approximate linear growth with accelerated increase in later stages. This stems from the data-driven architecture' s lack of explicit physical constraints (e.g., energy conservation, neutron flux feedback), limiting generalization of system dynamics.

[**Figure 8: see original paper**] Error distribution of coolant temperature field at 15s-40s
[**Figure 9: see original paper**] Error trend evolution in coolant temperature field during 15s-40s

Different latent dimensions affect feature interpretability and final performance. [Figure 10: see original paper] shows final R^2 values across dimensions, demonstrating that increasing latent dimension improves feature explanation and reduces cumulative error from information loss. For intuitive representation of prediction accuracy, [Figure 11: see original paper] and [Figure 12: see original paper] present 2D outlet and 3D coolant temperature field error contours from 5s-40s. Errors accumulate approximately linearly without abrupt growth, indicating sustained stability in capturing temporal dynamics—suitable for medium-to-long-term transient prediction in reactors.

[**Figure 10: see original paper**] Final R^2 for different latent dimensions
[**Figure 11: see original paper**] 2D outlet coolant temperature field during 5s-40s
[**Figure 12: see original paper**] 3D coolant temperature field from 5s to 40s

3.2 Fuel Temperature Field Prediction Analysis

Due to the fuel region' s 7.8× higher mesh count than coolant, computational cost increases significantly. The autoencoder structure was optimized to 256$\rightarrow$128$\rightarrow$$d_{\text{latent}}$ neurons while maintaining $d_{\text{latent}} = 100$. Other hyperparameters remained consistent. Training on RTX 4070 took 192.83s, still achieving $ 261×$ speedup over traditional solvers.

Prediction results and error contours are shown in [Figure 13: see original paper]. Errors increase gradually with time steps, with larger absolute errors concentrated in the fuel pellet center. At 40s, maximum positive error reaches 5.3K (0.87% relative error), while domain-averaged absolute error is 3.1K (0.52% relative error). The fuel center is the hottest region with intense heat conduction-nuclear reaction coupling, large temperature gradients, and strong nonlinearity—making it challenging for the model to capture. This could be addressed in future work by adding physics constraints to the neural network [**?**].

[**Figure 13: see original paper**] Fuel temperature field from 5s to 40s

3.3 Cladding Temperature Field Prediction Analysis

Neural network structure and hyperparameters were restored to coolant settings, but instead of 5s-40s iterative prediction, the first 100s served as training data and 101s-200s as test data for one-shot long-step prediction. On RTX 4070, prediction took 79.23s—less than the 100s test interval—successfully achieving real-time capability.

Maximum prediction error was 3.8K, with mean error at 200s only 2.2K and relative error $ $0.5%. Three data points were selected as measurement locations to compare predicted versus actual temperature evolution over time. [Figure 14: see original paper] shows excellent agreement between predicted and true curves.

[**Figure 14: see original paper**] Comparison between predicted and actual temperature values at measurement points

Cladding temperature field predictions and error contours ([Figure 15: see original paper]) show relatively uniform error distribution that gradually increases with time steps. The training phase relies on CFD-generated benchmark data (31 hours), while prediction operates independently of CFD solvers. Long-step prediction' s 79.23s execution time being shorter than the prediction window (100s) satisfies real-time control requirements. This enables state evolution to be completed before actual physical processes occur, providing a critical time window for early anomaly identification and safety intervention. Specifically, by real-time predicting dynamic evolution trends of key parameters like temperature, early warning mechanisms can be triggered to assist operators in adjusting control strategies, thereby enhancing operational safety under extreme conditions. This characteristic holds reference significance for real-time anomaly warning in nuclear thermal propulsion reactors.

[**Figure 15: see original paper**] Cladding temperature field from 101s to 200s

In summary, the model' s core applicability lies in multi-physics temperature prediction for nuclear thermal propulsion reactors under transient conditions, particularly for cores with dense honeycomb fuel channel arrays. The model can be embedded in reactor warning systems to provide early warnings for transients such as power steps and coolant flow fluctuations.

Compared to typical AI surrogate models like Graph Neural Networks (GNN), which—despite handling unstructured meshes through node feature aggregation —have limited spatiotemporal extrapolation capabilities (only generating 31s-40s snapshots from 30s of data), this model achieves long-term rolling predictions and long-step forecasting [**?**], demonstrating stronger extrapolation ability.

Conclusions

This study developed a real-time temperature field prediction model for nuclear thermal propulsion reactors based on latent space nonlinear operators (L-DeepONet), combining OpenFOAM high-fidelity numerical simulation data to achieve efficient extrapolation prediction under multi-physics coupling. Key conclusions are:

1. The model compresses temperature fields from high-dimensional space ($D_x \geq 10^6$) to low-dimensional latent space ($d_{\text{latent}} = 100$) via autoencoder, reducing DeepONet trainable parameters to 0.01%-0.001% of traditional full-domain models. Prediction achieves 2–3 orders of magnitude acceleration over CFD simulation, with training time (79.23s) shorter than

the test time window (100s)—a critical feature for real-time anomaly warning.

2. In 40s rolling predictions, mean relative errors for fuel and coolant temperature fields remain within 0.52% and 0.42%, respectively. In 100s long-step prediction, cladding mean relative error stays below 0.5% with maximum relative error under 1%, demonstrating high-accuracy multi-scenario capability.

Future research should focus on three aspects: (1) Introducing attention mechanisms to assign higher weights to latent features in high temperature-gradient regions where current prediction errors are significant. (2) Combining limited measured data with transfer learning to reduce dependence on pure simulation data. (3) Adopting interpretable dimensionality reduction techniques to enhance theoretical explainability of the latent space and prediction process.

Author Contributions Statement

ZENG Yiling: Overall paper design, L-DeepONet model design and computation, manuscript drafting.
LI Wei: Construction of nuclear-thermal coupling model using OpenFOAM, generation of original data, critical review of intellectual content, supervision, and supportive contributions.
MO Junyu: Assisted ZENG Yiling in L-DeepONet model parameter tuning.
LIU Zijing: Data analysis.
ZHAO Pengcheng: Funding acquisition, administrative/technical/material support, supervision, and supportive contributions.

References

[1] Elhareef M H, Wu Z. Physics-informed neural network method and application to nuclear reactor calculations: A pilot study[J]. Nuclear Science and Engineering, 2023, 197(4): 601-622.
[2] Prantikos K, Chatzidakis S, Tsoukalas L H, et al. Physics-informed neural network with transfer learning (TL-PINN) based on domain similarity measure for prediction of nuclear reactor transients[J]. Scientific reports, 2023, 13(1): 16840.
[3] Chen J, Liu X, Zhao P, et al. Research on reactor neutron flux and keff prediction method based on optimized extreme learning machine model[J]. Nuclear Technology, 2024, 47(10): 181-190.
[4] Sze V, Chen Y H, Yang T J, et al. Efficient processing of deep neural networks: A tutorial and survey[J]. Proceedings of the IEEE, 2017, 105(12): 2295-2329.
[5] Zeng F, Zeng Y, Zhao P, et al. Nonlinear reduced-order analysis of three-dimensional thermal stratification phenomenon in the upper plenum of a lead-bismuth cooled fast reactor based on a graph neural network[J]. Progress in Nuclear Energy, 2025, 188: 105874.
[6] Li Y, Zhao Y, Jia X, et al. Multimodal edge detection network with collabo-

rative CNN and Transformer[J]. Computer Engineering and Applications, 2025: 1-12.
[7] Kutyniok G, Petersen P, Raslan M, et al. A theoretical analysis of deep neural networks and parametric PDEs[J]. Constructive Approximation, 2022, 55(1): 73-125.
[8] Lu L, Jin P, Pang G, et al. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators[J]. Nature machine intelligence, 2021, 3(3): 218-229.
[9] Lu L, Jin P, Karniadakis G E. Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators[J]. arXiv preprint arXiv:1910.03193, 2019.
[10] Kobayashi K, Alam S B. Deep neural operator-driven real-time inference to enable digital twin solutions for nuclear energy systems[J]. Scientific reports, 2024, 14(1): 2101.
[11] Sharma Priyadarshini M, Venturi S, Panesi M. Application of DeepOnet to model inelastic scattering probabilities in air mixtures[C]//AIAA Aviation 2021 Forum. 2021: 3144.
[12] Zanardi I, Venturi S, Panesi M. Adaptive physics-informed neural operator for coarse-grained non-equilibrium flows[J]. Scientific reports, 2023, 13(1): 15497.
[13] Oommen V, Shukla K, Goswami S, et al. Learning two-phase microstructure evolution using neural operators and autoencoder architectures[J]. npj Computational Materials, 2022, 8(1): 190.
[14] Zhang J, Zhang S, Lin G. Multiauto-deeponet: A multi-resolution autoencoder deeponet for nonlinear dimension reduction, uncertainty quantification and operator learning of forward and inverse stochastic problems[J]. arXiv preprint arXiv:2204.03193, 2022.
[15] Kontolati K, Goswami S, Em Karniadakis G, et al. Learning nonlinear operators in latent spaces for real-time predictions of complex dynamics in physical systems[J]. Nature Communications, 2024, 15(1): 5101.
[16] Li W, Liu X, Chai X, et al. Development of neutron diffusion and transport algorithms based on finite volume method[J]. Nuclear Technology, 2025, 48(01): 165-178.
[17] Wang T, Liu Z, Zhao P, et al. Research on neutron flux maximization design method for high-flux lead-bismuth reactor based on BP neural network[J]. Nuclear Technology, 2024, 47(10): 159-171.
[18] Zhou M, Yi X, Liu G. Optimization of temperature feedback effects and core design for gas-cooled fast reactor[J]. Nuclear Technology, 2025, 48(01): 179-187.
[19] Jasak H. OpenFOAM: Open source CFD in research and industry[J]. International journal of naval architecture and ocean engineering, 2009, 1(2): 89-94.
[20] Cao Q, Goswami S, Karniadakis G E. Laplace neural operator for solving differential equations[J]. Nature Machine Intelligence, 2024, 6(6): 631-640.
[21] Deng J, Liu X, Guan C, et al. Conceptual design and its optimization of an air-cooled nuclear reactor for aircraft propulsion considering both thermal and neutronic performance[J]. Annals of Nuclear Energy, 2023, 181: 109591.

[22] Mosca L, Benjamin E J, Berra K, et al. Effectiveness-based guidelines for the prevention of cardiovascular disease in women—2011 update: a guideline from the American Heart Association[J]. Journal of the American College of Cardiology, 2011, 57(12): 1404-1423.

Note: Figure translations are in progress. See original paper for figures.