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: 2025-06-25T00:00:00+00:00

Abstract

To address the issues of poor real-time performance and low computational efficiency in nuclear reactor multi-physics coupling modeling, this study employs a temperature field temporal prediction method based on latent space neural operator (Latent-DeepONet). By constructing a lightweight "encoder-operator learning-decoder" architecture, 230,000–1,800,000-dimensional temperature field data 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 that: in 40-second iterative predictions, the average relative error of the temperature field in the fuel region is below 1%, and the error in the coolant temperature field is below 0.5%; in 100-second long-step predictions, the maximum relative error is controlled within 0.5%. The iterative prediction model requires less than 200s per prediction, achieving over 500× speedup compared to traditional CFD methods; long-step prediction requires 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 nuclear reactor transient condition simulation and safety control.

Full Text

Preamble

Learning Nonlinear Operators in Latent Spaces for Real-Time Prediction of Nuclear Thermal Propulsion Reactor Parameters

Yiling Zeng[1], Wei Li[2], Junyu Mo[1], Zijing Liu[1], Pengcheng Zhao[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 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**] A lightweight "encoding-operator learning-decoding" architecture was constructed to compress temperature field data of 230,000–1.8 million dimensions into a 100-dimensional latent space. An improved Deep Operator Network (DeepONet) was employed to learn dynamic evolution within this latent space, while a physics-constrained decoder was used for high-precision reconstruction. [**Results**] Validation using a nuclear-thermal coupling benchmark dataset generated by the computational fluid dynamics software OpenFOAM shows: 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 remains under 0.5%. Each iteration of the predictive model takes less than 200 seconds, achieving over 500 times acceleration compared to traditional CFD methods; a single long-step prediction takes 79.23 seconds, which is shorter than the prediction interval, thus meeting real-time prediction 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; high-temperature gas-cooled reactor; real-time prediction; autoencoder; multiphysics coupling

---

Nuclear thermal propulsion reactors operate under extreme high-temperature and high-pressure conditions for extended periods. Abnormal temperature variations affect coolant density and moderation capability, altering neutron absorption effects in the fuel and potentially increasing cladding rupture risk, thereby threatening reactor safety. However, traditional numerical simulation methods struggle to solve the partial differential equation-governed dynamic systems within the reactor in real-time due to prohibitive computational costs. Therefore, more efficient methods are needed for reactor temperature parameter prediction and estimation.

Artificial intelligence-driven surrogate models for high-dimensional multi-physics numerical simulations have found widespread application in nuclear reactor research[1][2][3], providing a promising approach for real-time prediction of complex differential equation models in physical systems. Deep Neural Networks (DNN)[4], as a crucial tool in machine learning for universal function approximation, have achieved remarkable success in image processing, Natural Language Processing (NLP), and engineering design optimization[5][6][7]. These networks can adapt to varying initial conditions, boundary conditions, model parameters, and geometric domains[8]. Traditionally, DNNs learn mapping functions between finite-dimensional vector spaces, whereas the emerging neural operators learn differential equation operators by approximating mappings between infinite-dimensional Banach spaces. The Deep Operator Network (DeepONet), proposed in 2019[9][10], leverages DNNs to construct surrogate models for conventional partial differential equations and has demonstrated rapid inference and high generalization accuracy in nuclear reactor surrogate model construction[11]. However, its performance degrades with increasing system scale and complexity[12][13]. To address this limitation, Vivek Oommen first combined DeepONet with Autoencoders (AE) to simulate two-phase microstructure growth described by the Cahn-Hilliard equation[14], and Zhang et al.[15] further integrated AE with neural operators for high-dimensional stochastic problems. Nevertheless, DeepONet trained in latent spaces still requires systematic evaluation regarding prediction accuracy and generalization capability. The L-DeepONet framework proposed by Kontolati et al.[16], tested through real-time predictions of highly nonlinear and multiscale systems in high-dimensional domains, has demonstrated superior prediction accuracy and computational efficiency compared to existing techniques. Furthermore, reactor parameter calculation and prediction must consider not only kinetic models but also complex boundary conditions and heat transfer factors[17][18][19], which neural operator training can effectively handle, indicating the method' s potential for real-time temperature parameter prediction in nuclear thermal propulsion reactors.

This study first employs the CFD software OpenFOAM[20] to obtain high-fidelity temperature field snapshot data for the nuclear thermal propulsion reactor. Following the L-DeepONet framework, an AE model is trained to achieve latent space mapping of high-dimensional inputs/outputs. Subsequently, a DeepONet operator is 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

Latent Space Nonlinear Operator (L-DeepONet) is a neural network architecture designed to learn nonlinear operators in complex physical systems. It employs an autoencoder to reduce high-dimensional data to a low-dimensional latent space, where a DeepONet model is subsequently trained to enhance computational efficiency and prediction accuracy[21].

The methodology comprises two primary steps: (1) nonlinear dimensionality reduction of input and output data {x, y} through an invertible reduction technique, and (2) training a DeepONet model in the latent space and inverse-transforming predicted samples back to the original space. The encoder projects high-dimensional data into the latent space, while the decoder reconstructs data from the latent space to the original dimension. During training, the encoder can project predicted samples back to the original space to evaluate their accuracy. After training, the L-DeepONet can be directly applied for real-time prediction of complex differential equation models in physical systems.

Let r denote the reduced data, G_ the approximated latent operator with trainable parameters in DeepONet. The encoder and decoder represent the two components of the dimensionality reduction method.

1.1.1 Latent Space Mapping Dimensionality reduction of data representation can not only accelerate DeepONet training but also reveal intrinsic data structures, enabling the model to learn more discriminative features and thereby improving prediction efficiency and accuracy. Considering that reduced data must be restored to the high-dimensional space after training and prediction—requiring an inverse mapping—and that the objective of dimensionality reduction is to enhance DeepONet operator accuracy and prediction efficiency rather than to identify the most advantageous method, the Autoencoder (AE) was selected for data reduction due to its architectural flexibility and inherent inverse mapping capability, as illustrated in Figure 1 [Figure 1: see original paper].

To train the autoencoder model and perform hyperparameter tuning to determine the optimal latent dimension d (where d < Dy), consider a time-dependent partial differential equation where dx corresponds to the input space dimension, and ms, mt represent the dimensions of spatial and temporal discretization, respectively. To feed the PDE output as image-like data into the autoencoder model, the PDE output is reshaped into distinct snapshots. Finally, both components of the autoencoder model are trained simultaneously by merging input and output data into a unified dataset {x, y}. The encoder and decoder are denoted as encoder: {x, y} → r and decoder: r → {x, y}, with trainable parameters _{encoder} and _{decoder}, respectively. The optimal parameter set _{ae} = { _{encoder}, _{decoder}} is obtained by minimizing the loss function:

L_{ae} = ||{x, y} - decoder(encoder({x, y}))||^2_R

where || · ||_2 denotes the standard Euclidean norm, and {x, y} represents the reconstructed dataset of merged input and output data. Experiments were conducted on three autoencoder models: single-hidden-layer AE (vanilla-AE), multi-layer AE (MLAE), and convolutional AE (CAE) with multiple hidden layers that convolve data through convolutional layers. The MLAE demonstrated the best performance.

1.1.2 Latent Space Nonlinear Operator (L-DeepONet) After AE training is completed, reduced data {x, y} is generated, and DeepONet G_ is used to approximate the latent representation mapping, where represents the model' s trainable parameters. As shown in Figure 2 [Figure 2: see original paper], DeepONet consists of two concurrent deep neural networks: a branch network that encodes the input function x at sensor locations, and a trunk network that inputs temporal coordinates {t_1, ···, t_m} at locations in the reduced space and evaluates the PDE output. The operator for the solution at input realization x_1 can be expressed as:

G_ (x_1)(t) = Σ_{k=1}^p b_k(x_1) × τ_k(t)

where b_k represents the output vector of the branch network, τ_k represents the output vector of the trunk network, and p is a hyperparameter controlling the final hidden layer sizes of both branch and trunk networks. The trainable parameters of DeepONet are obtained by minimizing the loss function:

L_{DeepONet} = ||y_r - G_ (x)||^2

where y_r denotes the actual reduced output and ŷ_r denotes the predicted reduced output. To feed the reduced output data into L-DeepONet' s branch network, it is reshaped into dimension d. After obtaining optimal parameters , the trained model can be applied to new input x D to generate reduced output, which is then transformed back to the original space using the decoder to obtain the approximate full-dimensional output ŷ_{rec} Y. Figure 2 illustrates the Latent DeepONet (L-DeepONet) framework.

1.2 Physical Model and Multi-Physics Coupling

1.2.1 Nuclear Thermal Propulsion Reactor Design This study employs a small-scale nuclear thermal propulsion reactor as the physical model. Operating under extreme high-temperature and high-pressure conditions for extended durations, this reactor requires close monitoring of its parameters, which has motivated research into real-time transient prediction of core parameters for nuclear thermal propulsion reactors[21].

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

1.2.2 Multi-Physics Coupling Model Since the entire core of the small-scale nuclear thermal propulsion reactor consists of thousands of coolant flow channels arranged in a honeycomb pattern, a simplified approach for CFD simulation of such complex flow structures is to treat the entire region as a unified domain and solve for velocity and pressure distributions using a porous media model. The governing equations for single-phase compressible fluid in the porous media model are as follows:

[Equation placeholders for mass, momentum, and energy conservation equations]

The above equations represent conservation of mass, momentum, and energy, where denotes fluid density, U is the superficial flow velocity, P represents system pressure, Cp is specific heat at constant pressure, Q is volumetric heat source, and γ is porosity representing the fraction of fluid volume in the solution domain.

For reactor dynamics, a point reactor model is employed:

[Equation placeholder for point reactor model]

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

For heat transfer problems, a Conjugate Heat Transfer (CHT) model is applied, with the principle:

[Equation placeholder for CHT model]

where T/ n represents the normal temperature gradient at the fluid-solid interface, q is heat flux, and k is thermal conductivity.

Real-Time Prediction Results for Nuclear Thermal Propulsion Reactor Coolant Temperature

2.1 Temperature Field Snapshot Acquisition

In establishing the nuclear thermal propulsion reactor model, specific design parameters are presented in Table 1 .

Table 1. Key parameters of the air-cooled nuclear engine concept design

[Table content with thermal-hydraulic and reactor physics parameters]

The computational mesh was generated using the commercial software STAR-CCM++, as shown in Figure 4 [Figure 4: see original paper]. The coolant region contains approximately 230,000 cells; the fuel region comprises approximately 1.8 million cells; and the cladding region contains about 370,000 cells.

Figure 4. Mesh structure

For the coolant region meshing, the channels within the core were modeled as an integrated coolant domain, significantly reducing the cell count. Triangular meshes were employed in the coolant region to ensure convergence.

The simulation was solved using OpenFOAM, obtaining transient data snapshots from reactor startup until 200 seconds post-startup, with recordings at one-second intervals. The total solution time was 31 hours.

2.2 Autoencoding and Decoding

Taking coolant temperature field prediction as an example, the autoencoder (AE) adopts a three-layer fully connected network architecture: the encoder progressively compresses input data to the latent space through 1024\$→512→$d_{latent}(ReLUactivation), whilethedecoderemploysasymmetrictopologytoreconstructlatentfeatu$
3}\$, mean square error (MSE) loss function, and GPU-accelerated tensor operations.

Figure 5 [**Figure 5: see original paper**] illustrates the MSE loss curve during training. The results show a rapid convergence phase during epochs 1-10, a steady optimization phase during epochs 10-30, and stabilization with minimal fluctuations after 20 epochs.

Figure 5. Decreasing trend of the AE loss function

For neural operator prediction requirements, a sufficiently high latent space dimension must be maintained to ensure preservation of data topological features. Through comparative experimental analysis, as shown in **Figure 6** [**Figure 6: see original paper**], the latent dimension d_{latent} exhibits a nonlinear relationship with information retention rate and computational efficiency. When d_{latent} = 100, the model achieves optimal balance between MSE (0.0005) and computational time (\$ \$57 s), thus this parameter was selected as the final reduction dimension.

Figure 6. Effect of latent dimension on dimensionality reduction

2.3 DeepONet Deep Operator Network

DeepONet comprises two sub-networks: Branch Net and Trunk Net. Branch Net' s input/output dimensions align with the AE latent space (d_{latent} = 100) and includes one hidden layer (d_{latent} → 128 → d_{latent}, ReLU activation). Trunk Net encodes spatiotemporal coordinates to generate basis functions that interact with Branch Net outputs, also containing one hidden layer (2 → 128 → d_{latent}, GELU activation), thereby establishing a nonlinear mapping from latent features to the next time step. Training employs a fixed learning rate of 1\$×10^{-4}\$, Adam optimizer, and MSE loss function. As shown in **Figure 7** [**Figure 7: see original paper**], the loss function converges after 400 training epochs.

Figure 7. Decreasing trend of the DeepONet loss function

This study first utilizes the first 10 seconds of coolant temperature field data as the training set, with the 11th and 12th seconds as the test set for prediction. After obtaining temperature field data for seconds 11 and 12, data from seconds 3-12 serves as the training set, and seconds 13-14 as the test set. This process iterates continuously, always using the most recent 10-second window as training data and the subsequent two seconds as test data, enabling real-time prediction up to 40 seconds. This strategy achieves long-term temporal extrapolation by dynamically updating local physical field features.

2.3.1 Coolant Temperature Field Prediction Analysis The model employs a purely data-driven architecture: AE achieves high-dimensional data reduction ($d_{latent} = 100$), DeepONet performs nonlinear prediction in the latent space, and the decoder reconstructs results back to physical space. Total training time on an NVIDIA GeForce RTX 4070 GPU was 130.48 seconds, representing a 386-fold computational efficiency improvement compared to the 14-hour OpenFOAM solution.

The temperature field error distribution and trend from 5s to 40s are presented in **Figures 8** [**Figure 8: see original paper**] and **9** [**Figure 9: see original paper**].

Figure 8. Error distribution of coolant temperature field from 5s to 40s

Figure 9. Error trend evolution of coolant temperature field from 5s to 40s

Prediction errors exhibit quasi-normal distribution characteristics and increase monotonically with time. Absolute errors remain below 2 K before 20 seconds, then gradually rise, reaching a maximum absolute error of 3.17 K at 40 seconds. The calculated average relative error of the temperature field is 0.42%, with a maximum relative error of 0.93%. For more intuitive visualization of prediction accuracy, **Figures 10** [**Figure 10: see original paper**] and **11** [**Figure 11: see original paper**] present two-dimensional coolant outlet temperature fields and three-dimensional coolant temperature field error contours from 5s to 40s.

Figure 10. Two-dimensional coolant outlet temperature field from 5s to 40s

Figure 11. Three-dimensional coolant temperature field from 5s to 40s

2.3.2 Fuel Temperature Field Prediction Analysis Due to the fuel region’ s high mesh count being 7.8 times greater than the coolant mesh, computational costs increase substantially. Therefore, the autoencoder (AE) architecture was optimized: neuron counts were reduced to $256 \rightarrow 128 \rightarrow d_{latent}$ while maintaining $d_{latent} = 100$; other hyperparameters remained consistent with the coolant prediction model. Total training time on an NVIDIA

GeForce RTX 4070 GPU was 192.83 seconds, still demonstrating significant efficiency advantages over traditional solvers (speedup ratio $\geq$ 261). Predicted fuel temperature fields and error contours are presented in **Figure 12 [Figure 12: see original paper]**.

Figure 12. Fuel temperature field from 5s to 40s

The error contours reveal that errors gradually increase with time steps, with larger absolute errors concentrated in the central fuel pellet region. At 40 seconds, the maximum positive error is 5.3 K (relative error 0.87%), while the domain-averaged absolute error is 3.1 K (relative error 0.52%). The fuel pellet center represents the highest temperature region in the reactor, characterized by intense coupling between heat conduction and nuclear reactions, large temperature gradients, and strong nonlinearity, making it challenging for the model to capture accurately. This phenomenon could potentially be addressed in future work by incorporating physics-based constraints into the neural network[23].

2.3.3 Cladding Temperature Field Prediction Analysis The neural network architecture and hyperparameters were reverted to match those used for coolant prediction, but with the first 100 seconds as training data and seconds 101-200 as test data for one-time long-step real-time prediction of cladding temperature. Prediction time on an NVIDIA GeForce RTX 4070 GPU was 79.23 seconds, less than the 100-second test interval, successfully demonstrating real-time prediction capability.

After prediction completion, the maximum error was 3.8 K, while the average error at 200 seconds was only 2.2 K, with relative error not exceeding 0.5%. Three data points were selected as temperature measurement points to compare temporal temperature variations with predictions, thereby evaluating prediction accuracy. **Figure 13 [Figure 13: see original paper]** presents the comparison results, showing that predicted curves closely match actual values, indicating good prediction performance.

Figure 13. Temperature prediction at measurement point locations

As shown in **Figure 14 [Figure 14: see original paper]**, predicted cladding temperature fields and error contours demonstrate relatively uniform error distribution, though overall errors exhibit an upward trend with increasing time steps.

Figure 14. Cladding temperature field from 101s to 200s

Since the model requires only 79.23 seconds for a 100-second long-step prediction —less than the actual time interval—it satisfies the real-time requirement of "prediction time < actual time interval" for nuclear reactor transient conditions. This characteristic enables the model to complete state evolution before the actual physical process occurs in the reactor, providing a critical time window for early identification of abnormal conditions and safety intervention. Specifically, by real-time prediction of dynamic evolution trends for key parameters such as temperature, early warning mechanisms can be triggered to assist operators in timely control strategy adjustments, thereby effectively enhancing operational safety of nuclear thermal propulsion reactors under extreme conditions. This feature holds significant reference value for real-time anomaly warning in nuclear thermal propulsion reactor operation.

Conclusions

This study develops a real-time temperature field prediction model for nuclear thermal propulsion reactors based on Latent Space Nonlinear Operator (L-DeepONet), combining high-fidelity numerical simulation data from OpenFOAM to achieve efficient extrapolation prediction under multi-physics coupling conditions. The main conclusions are as follows:

(1) The model compresses temperature fields from high-dimensional space ($D\_x \geq 10^6$) to low-dimensional latent space (d_{latent} = 100) through autoencoders, reducing DeepONet trainable parameters to 0.01%-0.001% of traditional full-domain models. Prediction time is accelerated by 3-4 orders of magnitude compared to CFD simulation, achieving real-time prediction where training time (79.23 s) is less than the test time window (100 s). This characteristic holds significant reference value for real-time anomaly warning in nuclear thermal propulsion reactor operation.

(2) In 40-second rolling predictions, average relative errors for fuel and coolant temperature fields remain stable within 0.52% and 0.42%, respectively. In 100-second long-step predictions, cladding average relative error stays below 0.5% with maximum relative error under 1%, demonstrating the model' s capability for high-accuracy prediction across multiple scenarios.

Author Contributions

Yiling Zeng was responsible for overall paper design, L-DeepONet model design and computation, and manuscript drafting. **Wei Li** was responsible for obtaining raw data using OpenFOAM and critically reviewing the intellectual content. **Junyu Mo** assisted Yiling Zeng in debugging L-DeepONet model parameters; **Zijing Liu** performed data analysis; **Pengcheng Zhao** secured research funding, provided administrative, technical, or material support, and offered guidance 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 Jingyu, Liu Xiyang, Zhao Pengcheng, et al. Prediction method of reactor neutron flux and keff based on the optimized extreme learning machine model[J]. Nuclear Techniques, 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 Yonghui, Zhao Yao, Jia Xiaohong, et al. Multi-modal edge detection network based on the collaboration of CNN and Transformer[J]. Computer Engineering and Applications, 2025-3-2: 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 Wei, Liu Xiaojing, Chai Xiang, et al. Development of neutron diffusion and transport algorithms based on finite volume method[J]. Nuclear Techniques, 2025, 48(01): 165-178.

[17] Wang Tong, Liu Zijing, Zhao Pengcheng, et al. Design method of high-flux lead-bismuth cooled reactor neutron flux maximization based on BP neural network[J]. Nuclear Techniques, 2024, 47(10): 159-171.

[18] Zhou Mengfei, Yi Xuan, Liu Guoming. Optimization of reactivity temperature feedback and core design of GFR[J]. Nuclear Techniques, 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.