ANS M&C 2021 - The International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering · Raleigh, North Carolina · October 3–7, 2021

INVESTIGATION INTO THE USE OF MACHINE LEARNING ASSISTED PREDICTION OF NODAL PARAMETERS FOR REDUCED ORDER NEUTRONIC SIMULATION MODELS*

Madhumitha Ravichandran[1], Cole A. Gentry[2], and Matteo Bucci[1]

[1]Department of Nuclear Science and Engineering,
Massachusetts Institute of Technology, Cambridge, MA 02139

[2]Oak Ridge National Laboratory,
PO Box 2008, Oak Ridge, TN 37831-6170

madhuar@mit.edu, cgentryca@ornl.gov, mbucci@mit.edu

ABSTRACT

Deep neural networks (DNNs) were trained to predict nodal cross sections, chi values, and assembly discontinuity factors (ADFs) for pressurized water reactor (PWR) 2D pin cell models and 2D lattice models to assess the feasibility of using DNNs as nodal parameter generators. Separate DNNs were trained for each individual nodal parameter to improve prediction accuracy and to transfer learning employed to reduce dataset volume requirements. DNNs were found to train and predict well for pin cell and lattice models when provided with sufficient data, and the required number of data points required to develop accurate lattice DNNs could be significantly reduced through transfer learning using a previously trained pin cell DNN.

KEYWORDS: Machine Learning, Deep Neural Network, Nodal, PWR

1. INTRODUCTION

The Virtual Environment for Reactor Applications (VERA) [1]—which was developed for high-fidelity modeling of light-water reactors (LWRs) through the Consortium for Advanced Simulations of Light Water Reactors (CASL) [2]—has recently been extended to perform simulations of molten salt reactors and other advanced reactors [3–6]. Although VERA can provide high-resolution computations at a substantially lower computational cost compared with traditional ultrafine resolution methodologies (e.g., Monte Carlo and computational fluid dynamics), it is still too computationally expensive for analyses that demand high simulation throughput, such as design optimization or uncertainty quantification.

Historically, this issue has been addressed for LWRs by using reduced order models (ROMs) like the traditional two-step procedure [7–9]. However, such high-performant and computationally expedient ROMs are not as well-established for the myriad of advanced reactor concepts under active development,

and establishing such ROMs can require considerable effort and resources. Given the recent advances in artificial neural networks (ANNs) [10–12] and the ability to use tools like VERA to generate high-fidelity synthetic training data, the accelerated development of useful high-fidelity high-throughput ROMs could be possible.

One such ROM approach currently being considered is illustrated in Fig. 1 (machine learning components are labeled with *ML*). This framework generally follows the more traditional LWR two-step, procedure-based nodal method, but it modifies the nodal parameter functionalization and pin power reconstruction steps to use ANNs.

This paper focuses on the nodal parameter generator (highlighted in yellow in Fig. 1), which is initially being developed for pressurized water reactor (PWR) simulations as a proof of principle. The goal of this generator is to train deep neural networks (DNNs) to predict nodal parameters for each node in a full-core model by using full-core features such as pin-wise thermal hydraulic (TH) conditions, assembly enrichments, boron concentrations, and control rod positions as input. This high degree of functionalization distinguishes the proposed approach from the more traditional method, in which nodal parameters are functionalized with respect to node-average conditions and fitting of data generated on a 2D lattice-wise basis. As such, ROM accuracy is generally expected to be an improvement over the traditional method, and it should require less development effort to address the *neighbor effects* and reflector modeling challenges typically encountered with 2D lattice-based approaches.

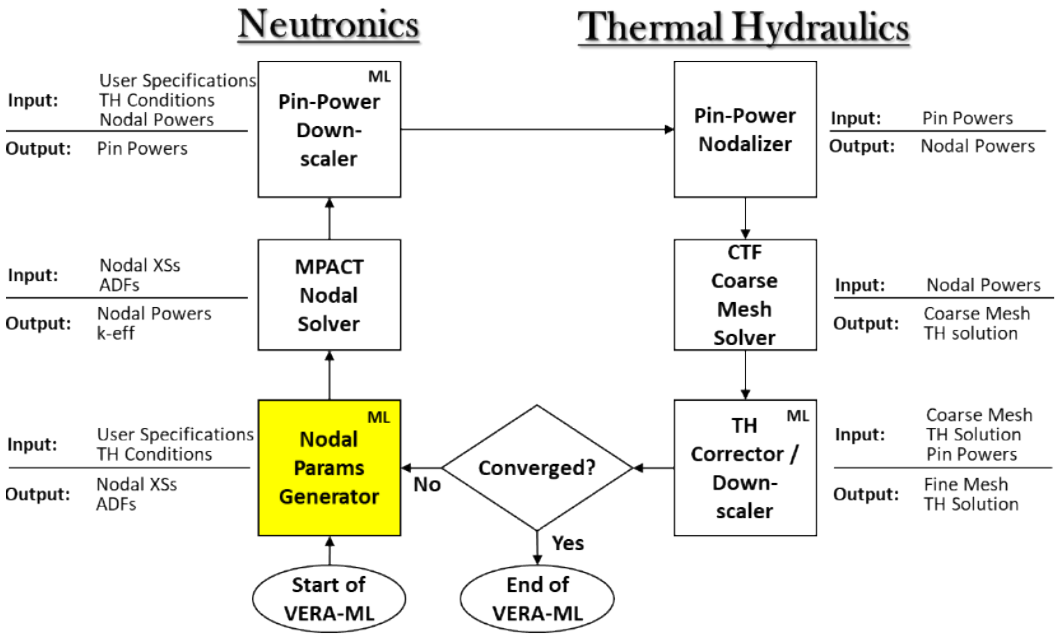

Figure 1. Proposed ROM approach

One significant concern of this approach is generating sufficient training data since DNNs are typically regarded as *data hungry*. As shown in Table 1, the number of input features (e.g., pin-wise TH conditions, assembly enrichments) and output features (e.g., nodal parameters for each assembly node and axial position) quickly expand as the domain is increased, which ultimately translates into a need for more training data points and a sharp increase in data generation compute costs.

To address this issue, the transfer learning of smaller domain models such as single 2D pin cells up to larger domain models like 2D lattices is considered. The ultimate goal of such a transfer-learning

approach is (1) to train primarily on smaller domain models that can easily and quickly generate large quantities of low-volume data, such as 2D pin cell models, and (2) to gradually scale the model up to 2D lattices, 3D assemblies, and ultimately to 3D full-core models, all while decreasing the number of datasets generated for each larger domain model.

This paper presents the current progress in developing these DNNs for 2D pin cell and 2D lattice predictions, as well as the efforts being made to address data requirements through transfer learning.

Table I. Comparison of input/output features and simulation computational costs for different LWR training domains.

Model	Number of input features	Number of output features	Data generation compute cost for statepoint (core hours)
2D pin cell	5	112	5.5e-4
2D lattice	895	112	0.01
3D assembly (TH-coupled)	43,711	6,496	3.2
3D core (TH-coupled)	8,436,031	1,253,728	368

2. GENERATION OF HIGH-FIDELITY NEUTRONICS SYENTHETIC DATA WITH MPACT SOLVER

To develop the data-driven DNNs to predict PWR neutronics parameters, a conventional 17 × 17 PWR lattice configuration that corresponds to the CASL progression problem 2A was considered [13].The VERA neutronics solver MPACT [14] was used to perform high-fidelity 2D single-pin and lattice simulations and calculate the relevant nodal parameters: assembly discontinuity factor (ADF), group neutron energy fraction (CHI), scattering cross sections (XSS), transport cross section (XSTR), removal cross section (XSRM), fission cross section (XSF), nu*fission cross section (NXSF), and kappa*fission cross section (KXSF).

Data were generated for 2D pin cell and 2D lattice models, and input features were randomly sampled on the bounds summarized in Table II. Because TH-coupled cases could not readily be run for these 2D models, fuel temperature, clad temperature, and moderator density were randomly sampled independently of one another. Although this does not reflect reality, it may at least allow the DNNs to learn the interplay of the independent effects. TH conditions for the 2D lattices were generated by using random curved second-order polynomial surfaces with varying numbers of peaks and troughs (1–6) to provide more informationally rich samples as opposed to using simple pin-wise random sampling, which neutronically produces similar results for each lattice simulation: that is, pin-wise random sampling is less informative and is not conducive to allowing DNNs to learn relationships. An example is shown in Fig 3.

For transfer learning, the input features of the 2D pin cell models were unfolded into a 2D lattice input feature equivalent. Pin cell features were copied into each 2D lattice fuel cell, assembly enrichment was set to the pin enrichment, control rods were assumed to be withdrawn, and guide-tube inner coolant densities were set to 0.7 g/cc. Target output features to be predicted were the same for the 2D pin cell and 2D lattice models, so no special unfolding was required for transfer learning, although this will not be the case for larger domain models.

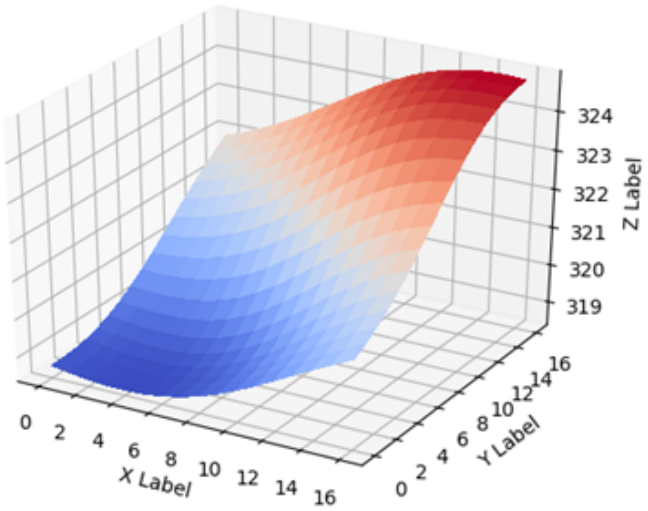

Figure 2. Example 17 × 17 Temperature Distribution

Table II. Range of input for MPACT.

Variable name	Range
Fuels enrichment values	1.8 and 4.9 wt% ^{235}U
Boron	0–2,000 ppm
Control rod position	in or out
Fuel temperature	286–1,386.25°C
Cladding temperature	286–357°C
Moderator density	0.66–0.743 $\frac{g}{cc}$

3. DNN AND TRANSFER-LEARNING FRAMEWORK DESIGN

Synthetic data generated via MPACT simulations were divided into training and testing datasets in the ratio of 0.7:0.3. All analyses and results presented in this work were performed using Python with the Keras deep learning package and TensorFlow backend. Several deep and shallow network architectures were tested and optimized, with the final architecture comprising 12 layers (hidden + output) apart from the input layer. The 12 layers followed a deep-down architecture, with the first layer comprising 211 neurons, and every layer having half the number of neurons as the previous layer. A separate neural network was used for each individual cross section and ADF at different energy groups (i.e., multiple single-output networks to cover all parameters). This was used instead of a single mapping between all inputs and outputs after spending considerable time attempting single mapping (i.e., multiple-input and multiple-output or multitask learning [MTL]). The separate neural networks performed much better than single mapping. It is postulated that the different outputs share or are correlated with very different input features, and hence the knowledge transfer is ineffective when using MTL as a result of low model capacity. Since there is no consensus on the advantages of MTL over the multiple single-output networks in general, and the superiority of one method over another is highly problem-specific, selection of the approach must be determined via trial and error. If performance were identical in this instance, then MTL would be preferred given the large number of outputs predicted and the effort needed to train the independent models. Hence, efforts are still being made to arrive at an MTL mapping that performs at the desired level of accuracy.

The rectified linear unit (ReLU) transfer function was used for the hidden layers, and the output layer used a linear activation function. Dropout regularization and early stopping was used to improve generalization, along with the Adam optimization for error function minimization. Roughly 8,000 data points were generated for 2D lattice simulations, and 4,000 data points were generated for 2D pin cell simulations. The range-to-margin ratio was high for the dataset, and the accuracy requirement was on the order of 10^{-5}. For this reason, instead of using a conventional mean-squared error as the error function, a multiplier was used along with the squared error to penalize heavily on the predicted errors for all outputs. The multiplier was tested and optimized on the range of 1 to 1,000 to reduce testing error.

For transfer learning, different strategies were investigated, including complete fine-tuning, partial layers freezing, and complete freezing of hidden layer weights (except output layers). For the sake of brevity, only the results of complete fine-tuning are presented in this work. In this approach, a set of initializer or *seed* weights were generated based on the 2D pin cell dataset. This was accomplished by (1) converting the 2D pin cell input features into a lattice pseudo-equivalent (i.e., copying 2D pin cell input features into each fuel cell of a 2D lattice while assuming some base conditions for non-fuel cell related features), (2) training a DNN to predict the pin cell nodal output using the pseudo-equivalent lattice input features, and (3) extracting the resulting weights of the trained DNN. These seed weights are then used as the initial weights for 2D lattice training, which are then fine-tuned to reduce the error function on the 2D lattice dataset.

It should be noted that though the aforementioned freezing strategies may be useful, they typically are most effective when the functionality of each individual layer of the DNN is well understood. In the case of convolutional neural networks (CNNs) for image recognition, each layer typically provides specific functionality—edge detection, corner detection, etc.—which can be understood through comprehensive examination of the developed CNN [15]. As such, one might readily identify which layers should be frozen and which should be fine-tuned based on the target application. However, in the present case, the DNN is solving a regression model, mapping the relationship between multiple inputs to the relevant output, and making it difficult to identify which layers should be frozen. Extensive time was spent testing different layer freezing strategies, but the results were at best comparable to a complete fine-tuning.

4. RESULTS AND DISCUSSION

Predicted parameter error was assessed using three measures: direct relative error of the predicted values against the reference value, as well as k-eff and nodal power errors determined by comparing reference nodal simulations against those using predicted values. When assessing nodal simulation errors, predicted values were tested individually with all other nodal parameters set to their reference values excepting the singular predicted value to be tested, as well as aggregately (AGGR), in which all predicted nodal values are used together.

Error results for non–transfer-learning pin cell predictions and lattice predictions are provided in Fig. 3 and Fig. 4 respectively. It should be noted that though the pin cell model consists of a single pin, it was still split into 4 nodes, so nodal power errors are still assessable. As can be seen for both pin cell and lattice cases, the errors are generally small, with k-eff errors less than 100 pcm and nodal power errors less than 0.25%.

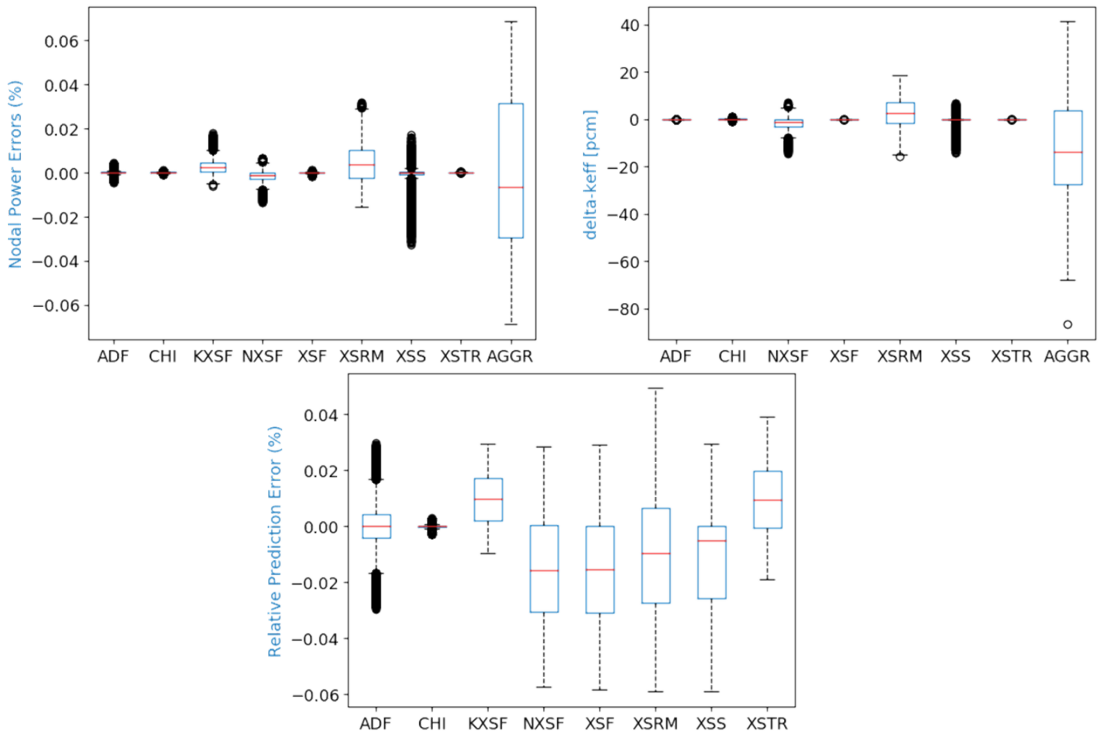

Figure 3. Nodal Parameter Prediction Errors of DNN vs. MPACT Reference for 2D Pin Cell

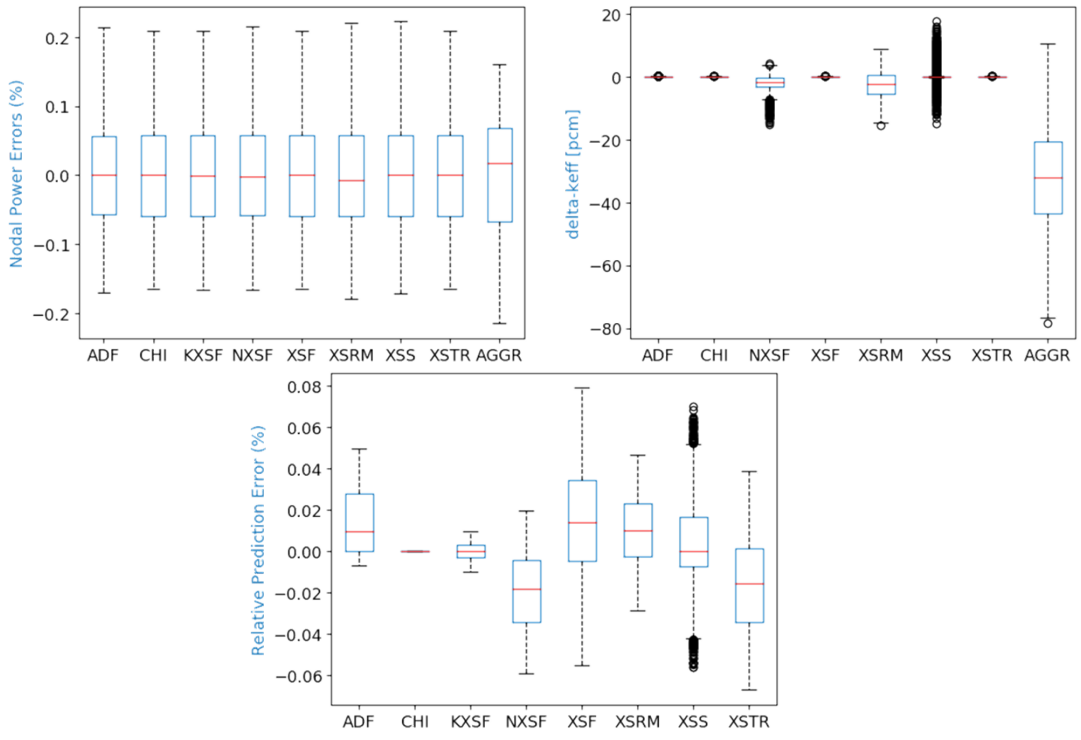

Figure 4. Nodal Parameter Prediction Errors of DNN vs. MPACT Reference for 2D Lattice

A similar comparison was made in Fig. 5 and Fig. 6 to show the efficacy of improving 2D lattice DNN performance via transfer learning from 2D pin cell DNNs for which only ~400 2D lattice data points were used to train the lattice models, but one set of DNNs employed transfer learning from the 2D pin cell DNNs, and the other did not. As seen in Figure 5, without transfer learning, DNN predictions were quite poor given the small dataset size available for training. Nodal power errors were in some cases greater than 6%, with k-eff errors exceeding 4,000 pcm. With transfer learning, these errors were greatly reduced, with nodal errors being less than 1% and k-eff errors less than 1,000 pcm. This demonstrates that transfer learning may be used to substantially improve DNN accuracy, and it could potentially serve as a means for greatly reducing the volume of training data required.

Training costs for these models on a DGX box with 1X Tesla V100 GPUs were 18.4 GPU hours for the 16,000 data point 2D lattice DNN, and 12.1 combined GPU hours for the transfer learning-based 2D lattice DNN (*combined* meaning the training of both the pin cell DNN used for transfer learning and the 2D lattice DNN). These training costs do not include the multiple iterations of architecture and hyperparameter tuning involved in developing the presented models.

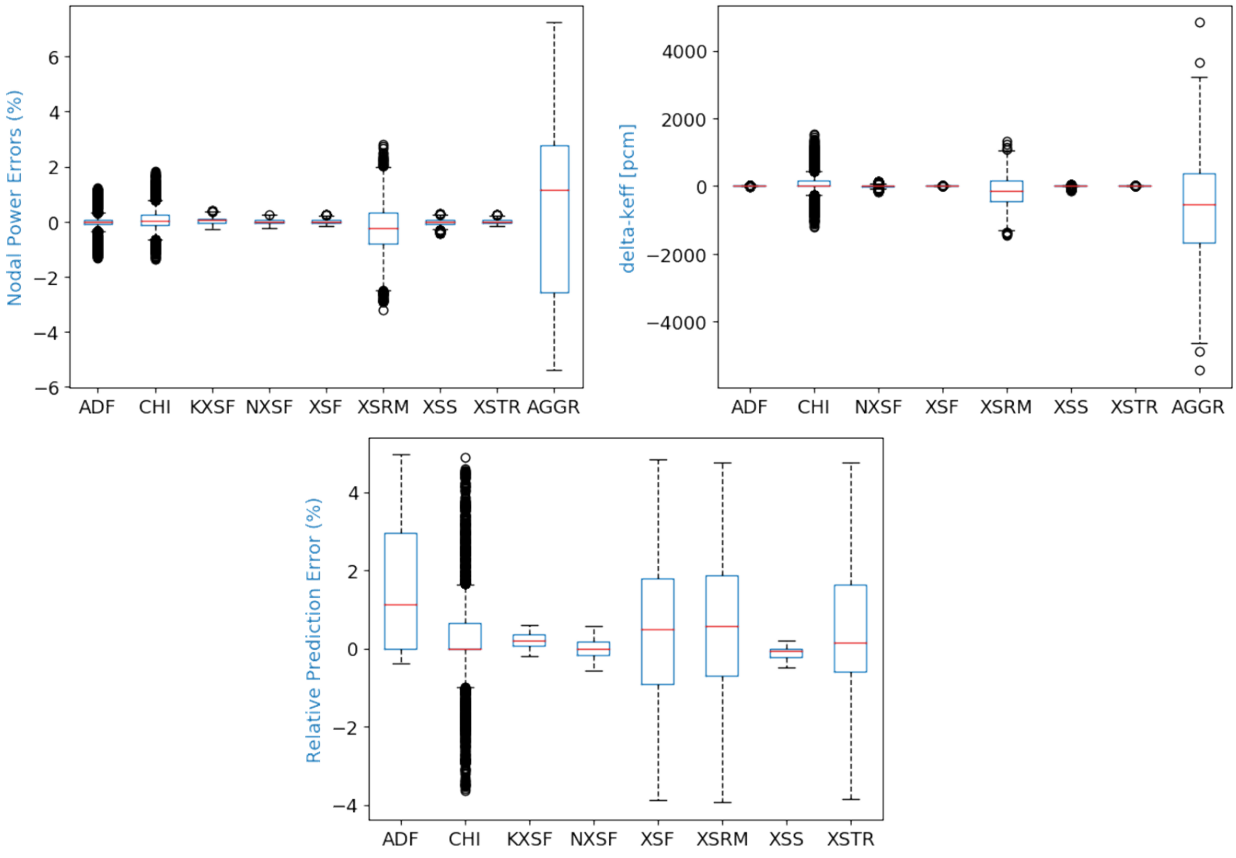

Figure 5. 2D Lattice Nodal Parameter Prediction Errors of DNN vs. MPACT Reference without Transfer Learning

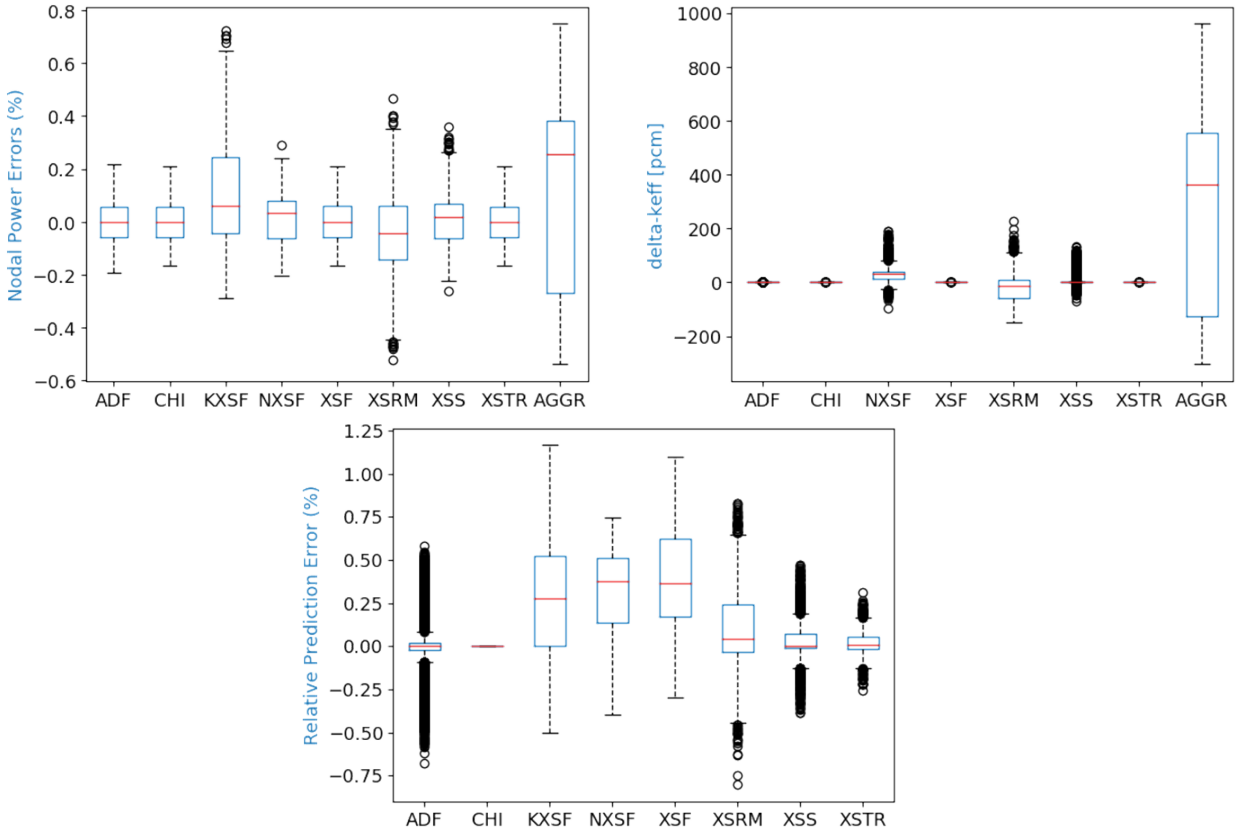

Figure 6. 2D Lattice Nodal Parameter Prediction Errors of DNN vs. MPACT Reference with Transfer Learning

5. CONCLUSIONS

This work demonstrates the ability of DNNs to predict nodal parameters for both PWR 2D pin cell and 2D lattice. These DNNs were trained by using synthetic high-fidelity simulations generated by MPACT over a range of different TH conditions, boron concentrations, control rod positions, and assembly enrichments, and they were able to predict the nodal parameters to a reasonable level of accuracy. Furthermore, a transfer-learning framework was developed and validated that could effectively use the trained network on a lower domain model (i.e., 2D pin cell) and fine-tune the network to predict the nodal parameters for a higher domain model (i.e., 2D lattice). This framework was used to demonstrate the viability of transfer learning as a means for reducing the size of higher domain training sets and computational training costs.

Future work aims to extend the DNN-based nodal parameter predictions up to 3D assembly domains and ultimately 3D cores with TH-coupled training data. Transfer learning will also be further investigated for these larger domains to help mitigate large data requirements. Future work will also explore different architectures that can effectively be used for MTL by suitably modifying the error function and employing advanced tools for accelerated convergence [16] to reduce the number of separate DNNs needed, along with embedding physical information [11] into the training process, which could make DNNs even more reliable.

Efforts will also be made to perform comparisons against more traditional nodal data functionalization methods (e.g., tabular interpolation, polynomial fitting) to assess performance as compared to various figures of merit. The computational expenses of generating sufficient training sets for and training of the

DNNs presented in this paper are much greater than the equivalent costs associated with typical functionalization techniques. However, these typical functionalization techniques are often challenged when faced with more than a few input features or when extrapolating beyond the available data [17, 18]. DNNs more easily afford a much greater number of degrees of freedom (i.e., pin-wise vs. nodal average features), and though their extrapolatable characteristics are still uncertain [19, 20], recent improvements in physics-informed deep networks [11, 21, 22, 23] provide evidence of methods for training DNNs to extrapolate beyond the domains of its labelled training set.

ACKNOWLEDGMENTS

This research was sponsored by the Laboratory Directed Research and Development Program of Oak Ridge National Laboratory (ORNL), which is managed by UT-Battelle LLC, for the US Department of Energy (DOE).

This research used resources of the Compute and Data Environment for Science at ORNL, which is supported by the DOE Office of Science under Contract No. DE-AC05-00OR22725.

REFERENCES

1. J. Turner et al., "The Virtual Environment for Reactor Applications (VERA): Design and Architecture". *Journal of Computational Physics*, **326**: pp. 544–568 (2016).
2. Consortium for Advanced Simulation of Light Water Reactors (CASL). Available from: https://www.casl.gov/.
3. A. Graham et al., "Development of Molten Salt Reactor Modeling and Simulation Capabilities in VERA," in *Global / Top Fuel*. Seattle, WA (2019).
4. A. Graham et al., "Extending MPACT to 2D Hexagonal Geometry," in *ANS Annual Meeting* (2020).
5. R. Taylor, "Implementation of Multi-phase Species Transport into VERA-CS for Molten Salt Reactor Analysis." 2019, University of Tennessee, Knoxville, TN, USA.
6. C. Gentry et al., "Initial Benchmarking of ChemTriton and MPACT MSR Modeling Capabilities." *Transactions of the American Nuclear Society*, **117** (2017).
7. H. Finnemann et al., "Interface Current Techniques for Multidimensional Reactor Calculations." *Atomkernenergie*, **30(2)**: pp. 123–128 (1977).
8. K. Koebke, "A New Approach to Homogenization and Group Condensation" (1980).
9. K. Smith, "Spatial Homogenization Methods for Light Water Reactor Analysis." Massachusetts Institute of Technology (1980).
10. B.N. Hanna et al., "Machine-Learning Based Error Prediction Approach for Coarse-Grid Computational Fluid Dynamics (CG-CFD)." *Progress in Nuclear Energy*, **118** (2020).
11. M. Raissi et al., "Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations." *Journal of Computational* Physics, **378**: pp. 686–707 (2019).
12. J. Berg et al., "A Unified Deep Artificial Neural Network Approach to Partial Differential Equations in Complex Geometries." *Neurocomputing*, **317**: pp. 28–41 (2018).
13. A. Godfrey, "VERA Core Physics Benchmark Progression Problem Specification." CASL-U-2012-0131-004 (2014).
14. B. Collins et al., "Stability and Accuracy of 3D Neutron Transport Simulations Using the 2D/1D Method in MPACT." *Journal of Computational Physics*, **326**: pp. 612–628 (2016).
15. X. Li et al., "H-DenseUNet: Hybrid Densely Connected UNet for Liver and Tumor Segmentation from CT Volumes." *IEEE Transactions on Medical Imaging*, 2018. 37(12): pp. 2663–2674.
16. A. Jagtap et al., "Adaptive Activation Functions Accelerate Convergence in Deep and Physics-Informed Neural Networks." *Journal of Computational Physics*, **404** (2020).

17. D. G. Cacuci, *Handbook of Nuclear Engineering: Vol. 1: Nuclear Engineering Fundamentals; Vol. 2: Reactor Design; Vol. 3: Reactor Analysis; Vol. 4: Reactors of Generations III and IV; Vol. 5: Fuel Cycles, Decommissioning, Waste Disposal and Safeguards*: Springer Science & Business Media (2010).
18. K. Ivanov and M. Avramova, Challenges in Coupled Thermal-Hydraulics and Neutronics Simulations for LWR Safety Analysis. *Annals of Nuclear Energy*. 34(6): pp. 501–513 (2007).
19. G. Hooker, *Diagnostics and Extrapolation in Machine Learning*. 2004: Stanford University.
20. K. Xu et al., *How Neural Networks Extrapolate: From Feedforward to Graph Neural Networks*. arXiv preprint arXiv:2009.11848 (2020).
21. X. Zhao et al., "On the Prediction of Critical Heat Flux Using a Physics-Informed Machine Learning-Aided Framework." *Applied Thermal Engineering*. 164: p. 114540 (2020).
22. M. Lutter, C. Ritter, and J. Peters, *Deep Lagrangian Networks: Using Physics as Model Prior for Deep Learning*. arXiv preprint arXiv:1907.04490 (2019).
23. X. Yang et al., "Predictive Large-Eddy-Simulation Wall Modeling via Physics-Informed Neural Networks." *Physical Review Fluids*. 4(3): p. 034602 (2019).