Research on Transfer Learning of Neural Network-based Thermal-Hydraulic Code Surrogate Model

Hyojun Yi [a,b], Hyeonmin Kim [c], Seunghyoung Ryu [a,b*]
[a] *Sejong University*
[b] *Artificial Intelligence and Robotics Institute*
[c] *Korea Atomic Energy Research Institute*
[*] *Corresponding author: shryu@sejong.ac.kr*

Keywords : deep-learning, surrogate model, thermal-hydraulic, transfer learning

1. Introduction

Thermal-hydraulic (TH) code simulations often incur substantial computational cost, with a single accident scenario simulation requiring minutes to hours. To mitigate this burden, data-driven deep-learning surrogate models that emulate TH simulation outputs have been actively investigated [1, 2]. Training such surrogates typically requires a dataset of paired accident scenarios and corresponding TH simulation results. However, for the same reason that motivates surrogate modeling in the first place, constructing these training datasets can itself be prohibitively time-consuming.

A key practical issue is that accident scenario representations and the associated TH output variables (THOVs) vary across reactor designs and TH codes. Specifically, the types and numerical definitions of events used to parameterize scenarios—as well as the resulting sets of THOVs (e.g., temperature, pressure, and flow rates)—can differ significantly. This raises an important question: must we regenerate a large-scale dataset and retrain a surrogate model from scratch whenever the reactor type or the underlying TH simulation code changes? If so, the cost of data generation becomes a major bottleneck for deploying deep-learning surrogates in practice.

To address this challenge, we investigate a transfer-learning framework that leverages knowledge learned from data generated for a particular reactor or TH simulation code. Specifically, we benchmark multiple transfer-learning schemes in a unified setup to assess whether transfer is beneficial.

2. Preliminaries & Methods

2.1 Surrogate Modeling

A deep-learning surrogate model for a TH code learns a simulator-induced mapping from an accident scenario representation to time-dependent THOVs. We consider two scenario input forms: (1) a scenario vector $\boldsymbol{x} = [x_1, \dots, x_d] \in \mathbb{R}^d$, where d is the number of scenario features, and (2) a time-varying scenario series $\boldsymbol{X} = [\boldsymbol{x}_1, \dots, \boldsymbol{x}_T] \in \mathbb{R}^{T\times d}$, where $x_t \in \mathbb{R}^d$ and T is the sequence length. The THOVs are a multivariate time series $\boldsymbol{Y} = [\boldsymbol{y}_1,\ \boldsymbol{y}_2, \dots, \boldsymbol{y}_T] \in \mathbb{R}^{T\times c}$, where $\boldsymbol{y}_t \in \mathbb{R}^c$ contains c output variables at time step t. The surrogate is a parametric function $f_\theta(\cdot)$ that approximates the simulator mapping: $\hat{Y} = f_\theta(\boldsymbol{x})$ or $\hat{Y} = f_\theta(\boldsymbol{X})$.

2.2 Transfer Learning

A source domain $\mathcal{S}$ (sepcific reactor) is considered with a dataset $D_{\mathcal{S}}$ and a target domain $\mathcal{T}$ (a different reactor) where only a small dataset $D_{\mathcal{T}}$ is available due to the high cost of generating TH simulations. The objective of transfer learning is to leverage knowledge learned in $\mathcal{S}$ to improve data-efficiency and predictive performance in $\mathcal{T}$. To this end, a surrogate model f_θ is first pre-trained on $D_{\mathcal{S}}$ to obtain source parameters $\theta^{(\mathcal{S})}$. The pre-trained model is then adapted to the target domain via fine-tuning, i.e., continuing training on $D_{\mathcal{T}}$ initialized with $\theta^{(\mathcal{S})}$ rather than with random parameters.

2.3 Fine-Tuning

Fine-tuning is one of the transfer learning strategies. As the term implies, it adapts a pre-trained model to a new target domain by adjusting its parameters. This can be expressed as $W_{\mathcal{T}} = W_{\mathcal{S}} + \Delta W$, where $W_{\mathcal{S}}$ denotes the pre-trained weights and ΔW represents the weight updates learned from target-domain.

2.4 Linear Probing then Full Fine-Tuning

Direct fine-tuning (DFT) of all parameters from the start can distort pre-trained representations, especially when the target domain differs from the source domain and the randomly initialized input/output heads are newly attached. In this setting, early updates are driven by unstable gradients from the input/output heads, which can rapidly overwrite useful pre-trained features and may lead the optimization toward suboptimal solutions.

To reduce such feature distortion, a two-stage adaptation strategy-linear probing followed by full fine-tuning (LP-FT) [3]-is employed. In stage 1 (linear probing), the pre-trained backbone is frozen and only the input/output heads are trained on the $D_{\mathcal{T}}$, allowing the newly added layers to learn a stable mapping from pre-trained features to target inputs/outputs. In stage 2 (full fine-tuning), the entire network is unfrozen and training continues with a smaller learning rate, enabling gradual backbone adaptation while better preserving the pre-trained feature structure.

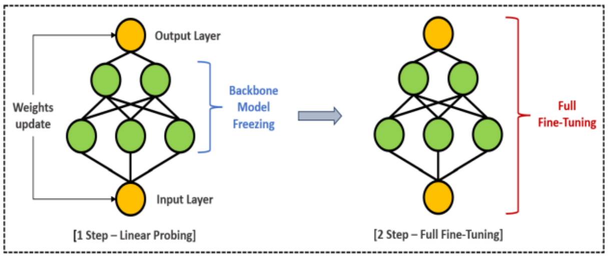

Fig. 1. The two-step transfer learning process: (1) linear probing and (2) full fine-tuning.

3. Experiments and Results

3.1 Datasets and Experimental settings

For source-domain pre-training, the $D_{\mathcal{S}}$ is generated for the OPR-1000 using the Nuclear Plant Analyzer (NPA) code. The dataset consists of paired scenario series inputs and corresponding THOVs. For target-domain learning-including fine-tuning and training from scratch-the $D_{\mathcal{T}}$ is generated for the i-SMR using the Multi-dimensional Analysis of Reactor Safety (MARS) code. This dataset consists of paired scenario vector inputs and corresponding THOVs.

3.2 Source Domain Pre-Training

Given the size of the source dataset $D_{\mathcal{S}}$ (≈1.1M samples), we pre-train the LCT model [2] iteratively by partitioning $D_{\mathcal{S}}$ into 50K-sample chunks and training sequentially on each chunk. To avoid rapidly overwriting knowledge learned from earlier chunks during iterative pre-training, a small learning rate of 1×10^{-5} is used.

3.3 Target Domain Fine-Tuning

For a fair comparison across target-domain training schemes, same model configuration is used in all experiments, including the full architecture with the same backbone network (LCT [2]) and set of training hyperparameters. The only scheme-specific difference is introduced in LP-FT: during linear probing, a learning on order of 10^2 higher than that used in full fine-tuning is employed to adapt the initialized input/output heads while keeping the pre-trained backbone fixed.

3.4 Results Analysis

To compare the performance of three training scheme-training from scratch (Scratch), DFT, and LP-FT-we report mean squared error (MSE), mean absolute error (MAE), and root mean squared error (RMSE). The corrsponding results are summarized in Table I.

LP-FT consistently achieves the best performance across the evaluated metrics, outperforming the Scratch by approximately 4%, followed by DFT. These results demonstrate that fine-tuning effectively mitigates the distribution shift between $D_{\mathcal{S}}$ and $D_{\mathcal{T}}$, outperforming random initialization. Furthermore, LP-FT yields an additional performance boost over DFT, confirming that its progressive adaptation enables better target-domain generalization.

Table I: Performance comparison of Scratch, DFT, and LP-FT across evaluation metrics.

Metric	**Training Scheme**		
	Scratch	**DFT**	**LP-FT**
MSE	.0260(.09)	<u>.0256</u>(.09)	**.0249**(.09)
MAE	.0481(.10)	<u>.0479</u>(.10)	**.0463**(.10)
RMSE	.0939(.11)	<u>.0937</u>(.11)	**.0907**(.11)

4. Conclusions

This study provides an early investigation of transfer learning across different reactors and TH simulation codes. We compare three learning schemes for target domain (Scratch, DFT, and LP-FT), and find that fine-tuning consistently improves performance over training from Scratch, suggesting the potential of cross-reactor/code transfer. LP-FT further delivers the most stable gains via progressive adaptation. Future work will explore representations that map heterogeneous datasets into a shared space.

ACKNOWLEDGMENT

This work was partially supported in part by the National Research Council of Science & Technology (NST) grant from the Korea government (MSIT) (No. GTL24031-000), in part by the IITP (Institute of Information & Communications Technology Planning & Evaluation) - ICAN (ICT Challenge and Advanced Network of HRD) grant funded by the Korea government (Ministry of Science and ICT) (IITP-2026-00436528), and in part by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (RS-2024-00454514)

REFERENCES

[1] Ryu, Seunghyoung, et al. "Probabilistic deep learning model as a tool for supporting the fast simulation of a thermal–hydraulic code." *Expert Systems with Applications* 200 (2022): 116966.
[2] Yi, Hyojun, Hyeonmin Kimb, and Seunghyoung Ryua. "Hybrid Deep Learning-Based Surrogate Model for Thermal-Hydraulic Codes with LSTM, CNN, and Transformer."
[3] Kumar, Ananya, et al. "Fine-tuning can distort pretrained features and underperform out-of-distribution." *arXiv preprint arXiv:2202.10054* (2022).