CE7454 Final Project: The New Multifidelity-DeepONet and its Application on Data Center Thermal Prediction.

Minghao LI
G2204568C
minghao002@e.ntu.edu.sg

Haozhuang Chi
G2204892K
chih0001@e.ntu.edu.sg

Wong Siew Chien
G2205169F
wong1343@e.ntu.edu.sg

Abstract

With the increasing demand of data processing, efficient thermal management in data centers has become a critical challenge and we are proposing a new way to predict data center thermal distribution using our new Multifidelity-DeepONet. Due to the nature of data center operating environments, which do not allow any fluctuations in temperature, we often lack sufficient training data for thermal prediction from sensors. The temperature readings typically remain constant, making it challenging to train AI models for thermal prediction. We developed a Multifidelity-DeepONet that leverages both high-fidelity CFD simulations and real-time, low-fidelity sensor data. We also proved that Multifidelity-DeepONet has better performance compare to all the others baseline methods in our experiments. The source code and the results of the experiments can be found at https://github.com/liminghao0914/CE7454-final.

1. Introduction

This section justifies the main reason behind our proposed Multifidelity-DeepONet approach which leverages both high-fidelity CFD simulations and low-fidelity sensor data. We also conduct a series of experiments to validate the effectiveness of our approach in various scenarios to show its advantages over conventional methods.

1.1. DeepONet

Deep Operator Networks (DeepONets) [3] have the ability to learn operators mapping between infinite-dimensional function spaces. Therefore, DeepONets can effectively handle complex function-to-function relationships.

DeepONets are made up of two sub-networks: the branch network and the trunk network. The branch network processes the input functions, while the trunk network deals with the coordinates of the points where the output function is evaluated. The final output is a linear combination of the outputs from these two sub-networks.

The operation of a DeepONet can be mathematically represented as:

$$G(u)(y) = \sum_{k=1}^{p} b_k \cdot t_k, \tag{1}$$

where G is the operator that the network aims to learn, u is the input function, and y represents the points of evaluation. In this expression, b_k are the outputs from the branch network corresponding to the input function U, and t_k are the outputs from the trunk network corresponding to the evaluation points y. The parameter p represent the number of terms in the linear combination and is a design choice in the network architecture.

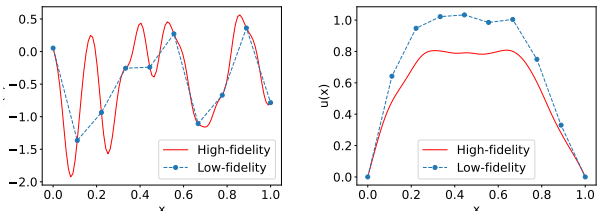
This equation demonstrates DeepONet's capability to map complex relationships in data-driven environments, and its really beneficial for scenario like data center thermal prediction that has less training data range available.

1.2. Multifidelity-DeepONet

In complex simulation models that require high computational resources, Multifidelity techniques are commonly used [1]. Our Multifidelity-DeepONet extends the concept of DeepONets by incorporating multiple fidelity levels of data, aiming to improve its ability to leverage the detailed information from high-fidelity simulations and the real-time relevance of low-fidelity data. This approach is particularly effective in scenarios where high-fidelity data is limited or costly to acquire, as the low-fidelity data can significantly helps the learning process. In our scenario, high-fidelity data is derived from detailed Computational Fluid Dynamics (CFD) simulations, which are expensive to obtain, while low-fidelity data comes from real-time sensor readings.

2. Numerical Experiments

In this section, we choose two scenarios, i.e. Poisson equation and reaction-diffusion to test the performance of Multi-fidelity DeepONet as a PDE solver.



(a) A random sample f (b) low- and high-fidelity u

Figure 1. Examples of the Poisson equation. (a) An example of randomly sampled f for testing. (b) The corresponding low- and high-fidelity solutions.

2.1. Poisson Equation

The Poisson equation is a partial differential equation of elliptic type with broad utility in mechanical engineering and theoretical physics. It arises, for instance, in the solutions of partial differential equations that describe physical phenomena such as heat conduction, electrostatics, diffusion, and quantum mechanics.

Mathematical Model. We examine a 1D Poisson equation to evaluate different multifidelity methods and demonstrate the effectiveness of multifidelity DeepONet. The Poisson equation is given by:

$$\frac{d^2u}{dx^2} = 20f(x), \quad x \in [0, 1], \tag{2}$$

with Dirichlet boundary conditions $u(0) = u(1) = 0$. The goal is to learn the operator $G : f \mapsto u$ from the forcing term $f(x)$ to the PDE solution u, where f is sampled from a Gaussian random field with zero mean.

Data Generation. The high-fidelity and low-fidelity datasets are generated by solving the Poisson equation using the finite difference method with different mesh sizes Δx. For high-fidelity solutions, $\Delta x = 1/99$ is used, and for low-fidelity solutions, $\Delta x = 1/9$. Figure 1 displays a randomly sampled f and the corresponding low- and high-fidelity solutions.

Results. The experiment focused on evaluating various multifidelity (MF) strategies for DeepONet, using a small high-fidelity and a large low-fidelity dataset. The results, illustrated in Figure 2, reveal that both MF-DeepONet and Normal DeepONet achieved commendable performance. MF-DeepONet, in particular, excelled by effectively utilizing the combined data, outshining the Normal DeepONet marginally.

The High-Fidelity approach, despite being competent, was slightly less effective compared to the multi-fidelity methods. This highlights the importance of integrating multi-fidelity data for complex PDE problems like the Poisson equation. The Low-Fidelity model, however, showed markedly inferior performance, underscoring the limitations of using low-resolution data alone.

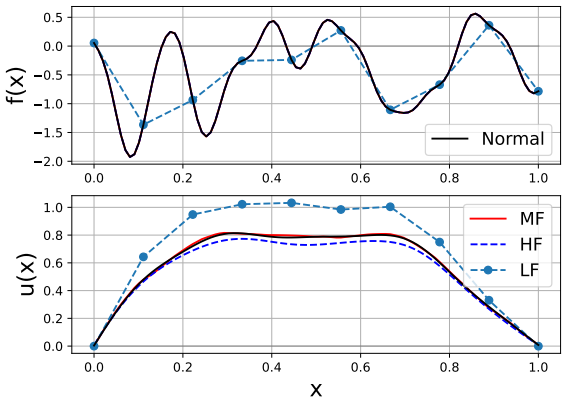


Figure 2. Learning poison equation with two types of dataset (small-range high-fidelity dataset and large-range low-fidelity dataset) for Normal and Multi-fidelity DeepONet.

Overall, these observations affirm that multi-fidelity data integration can significantly bolster the predictive accuracy of DeepONet models, especially when high-fidelity data is limited.

2.2. Reaction-Diffusion System

A diffusion-reaction system refers to a set of chemical reactions where the reactants are distributed in space and time through a process of diffusion. These systems are described by PDEs which combine the effects of spatial movement (diffusion) and chemical transformation (reaction). We use the typical advection-diffusion-reaction (ADR) system to illustrate the performance of multi-fidelity DeepONet.

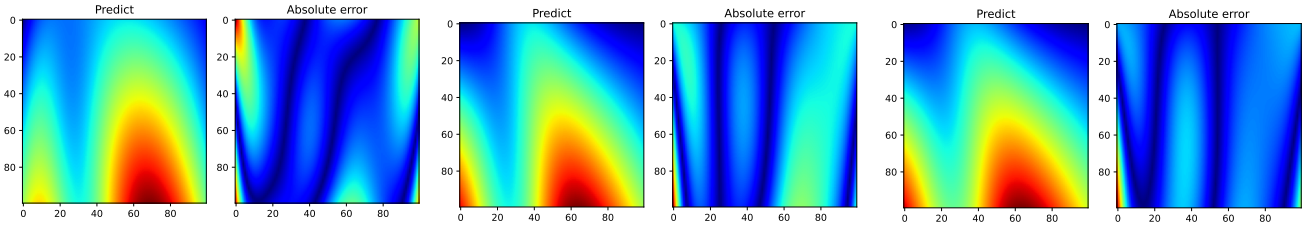
Mathematical Model. The ADR equation combines the three processes, i.e. Advection (the transport of a substance), Diffusion (the spreading of a substance), and Reaction, into a single mathematical model. The equation takes the form

$$\frac{\partial S}{\partial t} = \nabla.(Su) + \nabla.(D\nabla S) + \rho(S), \tag{3}$$

where u is the concentration of the substance, t is time, and x is the spatial variable. The functions $k(x)$, $v(x)$, $g(u)$, and $f(x, t)$ represent the diffusion coefficient, the advection speed, the reaction term, and any external sources or sinks of the substance, respectively.

Data Generation. With the help of DeepXDE [], we generate a time-dependent dataset with 40 boundary conditions and 20 initial conditions. There are 320 sensors in the trunk net for training and 566 sensors for testing.

Results. The results in Figure 3 show the performance of each type of DeepONet fed with different fidelity data samples. The high error area (red) in Figure 3a indicates that DeepONet cannot perform well with small-range high-fidelity data. With wide-range low-fidelity data, there's no obvious high error in Figure 3b, which is mainly caused



(a) DeepONet with small-range HF Data-A (b) DeepONet with wide-range LF Data-A (c) Multi-fidelity DeepONet with MF Data-A

Figure 3. Predicted solution of three types of trained DeepONets in the test dataset. Data-A and Data-B are two random selected samples.

by model overfitting. However, the deviation in the training data can lead to a large medium error area (green). As shown in Figure 3c, Multi-fidelity DeepONet mitigates the above issues.

For multi-fidelity DeepONet, the low-fidelity solver is assumed to be perfect, and its prediction is used as an input to the DeepONet. We apply residual learning and input augmentation techniques to enhance the model's accuracy. The residual learning approach significantly reduces the error, achieving an m.s.e. as low as 0.0021 ± 0.0007. Input augmentation approach II uniformly outperforms the high-fidelity only model, with the smallest error of 0.0052 ± 0.0033. However, combining residual learning with input augmentation does not yield further improvement in this case, as shown in Figure 2.

3. Data Center Thermal Prediction

In this study, we focus on applying the Multifidelity-DeepONet for thermal prediction in data centers. Our goal is to predict temperature distributions and identify potential hotspots that could damage the hardware. [2]

3.1. Problem Setup

The main challenge lies in capturing the complex interplay of factors influencing data center temperatures. These include the spatial layout of servers and cooling systems, varying operational loads, and environmental conditions. Accurately modeling these elements is crucial for realistic temperature predictions. [4]

Our objective is twofold: firstly, to develop a predictive model capable of real-time thermal mapping across different zones of a data center, and secondly, to use this model to preemptively identify hotspots.

3.2. Data Generation

In this study, we combines high-fidelity simulations and real-world sensor data for our model.

High-Fidelity Simulated Data: Derived from Computational Fluid Dynamics (CFD) simulations, this data provides detailed thermal mappings based on energy conservation principles, aligning with PDE form of thermodynamics. These simulations, essential for model benchmarking, offer a precise understanding of thermal behaviors and are further simplified into ordinary differential equations (ODE) for transient heat transfer processes.

Low-Fidelity Real-world Data: Comprising sensor data from data centers, this dataset captures real-time operational conditions, including CRAC supply and return temperatures and room sensor readings. Despite potential limitations in sensor accuracy and stability, this data is vital for understanding day-to-day operational dynamics.

Simulated CRAC Placement Data: Using random locations for 16 CRAC units within predefined spatial boundaries, we simulate the potential impact of CRAC unit placements on thermal dynamics. This additional layer of simulation aids in comprehending the spatial influence on temperature distribution.

Data Integration and Processing: Both high-fidelity and low-fidelity data, along with simulated CRAC placement data, are integrated and processed, ensuring consistency and facilitating effective model training. This multifaceted dataset not only captures the detailed CFD simulations and CRAC placements but also reflects the actual operational conditions observed in the sensor data.

3.3. Mathematical Exposition

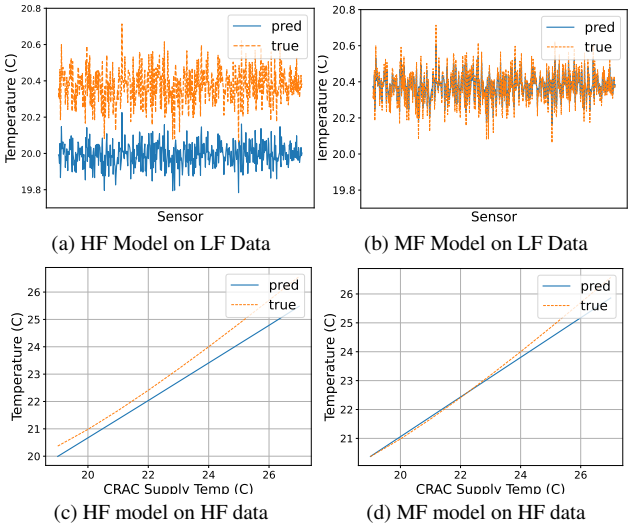
Our mathematical model for thermal prediction in data centers integrates concepts from the study [5]. We employ computational fluid dynamics and heat transfer (CFD/HT) techniques, with the core model based on energy conservation laws expressed in simplified differential equations. The transient heat transfer within the data center is modeled as:

$$\frac{\partial T_z}{\partial t} + U \cdot \nabla T_z = \Gamma_{\text{eff}} \cdot \nabla^2 T_z + Q, \tag{4}$$

where T_z is the zone air temperature, U the air velocity vector, Γ_{eff} the diffusion coefficient, and Q the heat load from IT equipment and CRACs. To accommodate practical modeling scenarios, we assume a uniform temperature distribution within the data hall. This simplifies our model to:

$$\frac{dz}{dt} = \sigma \cdot (s_{\text{in}} - z) + \frac{IT}{\rho_{\text{air}} \cdot c_p \cdot V} \tag{5}$$

where σ is the air recirculation ratio, s_{in} the supply air temperature, and z the zone air temperature. IT represents



(a) HF Model on LF Data (b) MF Model on LF Data

(c) HF model on HF data (d) MF model on HF data

Figure 4. The interpolation (Figure 4a, Figure 4b) and extrapolation (Figure 4c, Figure 4d) performance analysis of multifidelity-DeepONet. The corresponding MAEs for the four experiments are 0.378, 0.024, 0.403, 0.106.

IT power usage, ρ_{air} the air density, c_p the air heat capacity, and V the data hall volume.

This model effectively captures the dynamics of heat transfer in data centers, which helps us to carry out thermal predictions accurately.

3.4. Results

This section presents the experimental outcomes from employing both the standard DeepONet and the Multifidelity-DeepONet models for predicting thermal dynamics in data centers.

DeepONet Results. The standard DeepONet, renowned for its operator learning proficiency, was employed to model complex thermal behavior in data centers. Its iterative training approach and focus on loss minimization led to a refined understanding of temperature distributions.

Multifidelity-DeepONet Results. The Multifidelity-DeepONet, integrating data from varied fidelity levels, demonstrated enhanced extrapolation skills. This integration led to more accurate predictions in diverse operational scenarios, as depicted in the comparative figure set below.

3.5. Performance of Multifidelity-DeepONet

Accuracy Analysis. In Figure 4a, it is evident that the high-fidelity (HF) model, when trained exclusively on high-fidelity data, struggles to accurately predict low-fidelity (LF) data as the mean average error sits higher at 0.378. The predictions consistently undervalue the actual temperatures. This discrepancy indicates a limitation in the HF model's learning scope, as it fails to capture some critical features present in LF data.

Contrastingly, Figure 4b shows that the multifidelity (MF) model achieves better accuracy and lower mean average error of 0.024. The predicted and actual values align closely, suggesting that the MF model effectively learns and incorporates features from both HF and LF data. This integrated learning approach enables the MF model to adapt more accurately to the real-world conditions represented in the LF data.

Extrapolation Analysis. Figure 4c show that HF models and MF models have 0.403 abd 0.106 mean average error respectively. The HF model's predictions start diverging from the actual values as the temperature range increases. This divergence suggests a limitation in the model's extrapolation capacity, possibly due to a lack of exposure to varying data ranges during training. This limitation highlights the potential inadequacy of HF data in capturing broader operational scenarios.

In contrast, Figure 4d demonstrates that the MF model maintains closer alignment with the actual values, even as the data range varies. This observation underscores the effectiveness of integrating HF and LF data, as it allows the MF model to generalize better across different operational conditions and data ranges.

3.6. Conclusion

Our analysis confirms the superior performance of the Multifidelity-DeepONet in both accuracy and extrapolation, especially when dealing with diverse data scenarios. The integration of multifidelity data not only enhances the model's learning capacity but also ensures robust predictions across varying conditions and ranges. This capability is particularly valuable in dynamic environments like data centers, where the ability to adapt to changing conditions is crucial for efficient thermal management.

References

[1] Nicola Demo, Marco Tezzele, and Gianluigi Rozza. A deeponet multi-fidelity approach for residual learning in reduced order modeling. *Advanced Modeling and Simulation in Engineering Sciences*, 10(1), July 2023. 1

[2] Yogesh Fulpagare and Atul Bhargav. Advances in data center thermal management. *Renewable and Sustainable Energy Reviews*, 43:981–996, 2015. 3

[3] Lu Lu, Pengzhan Jin, Guofei Pang, et al. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, Mar. 2021. 1

[4] Roger R Schmidt, Ethan E Cruz, and M Iyengar. Challenges of data center thermal management. *IBM Journal of Research and Development*, 49(4.5):709–723, 2005. 3

[5] Ruihang Wang, Zhiwei Cao, Xin Zhou, Yonggang Wen, and Rui Tan. Phyllis: Physics-informed lifelong reinforcement learning for data center cooling control. In *ACM E-Energy*, pages 114–126, 2023. 3