A neural surrogate solver for radiation transfer

Aleksei G. Sorokin *
Department of Applied Mathematics
Illinois Institute of Technology
Chicago, IL 60616
asorokin@hawk.iit.edu

Xiaoyi Lu †
FM Research Division
Norwood, MA 02062
Xiaoyi.Lu@fmglobal.com

Yi Wang
FM Research Division
Norwood, MA 02062
yi.wang@fmglobal.com

Abstract

Radiative transfer is often the dominant mode of heat transfer in fires, and solving the governing radiative transfer equation (RTE) in CFD fire simulations is computationally intensive. This work develops a new versatile toolkit for training neural surrogates to solve various RTEs across different geometries and boundary conditions. We generalize previous work in the area to include unknown boundary conditions and to perform Principal Component Analysis (PCA) for dimension reduction in this context. This enables efficient training of high-dimensional neural surrogate solvers for a large class of RTEs. The mesh free nature of these surrogates enables them to overcome the ray effect suffered by traditional solvers. Our results demonstrate that neural surrogate can provide fast and accurate radiation predictions for practical problems important to fire safety research.

1 Introduction

FireFOAM Wang et al. [2011, 2014] is an open-source CFD solver for large-scale fires, capable of modeling all of the complex physics that occur during an industrial fire, including heat transfer, pyrolysis, turbulent combustion, and water suppression. Despite being highly scalable on parallel computers, the solver takes several days to over two months to simulate practical fires. The high computational cost is primarily attributed to solving pressure and radiation equations.

GPU acceleration in FireFOAM is achieved by employing NVIDIA's AMGx solvers to offload linear solver computations from CPUs to GPUs, significantly reducing the time required to solve the pressure equation. However, since matrix and vector assembly still occur on CPUs, this method is not applicable for radiation calculations using the discrete ordinate method, which would require extensive code refactoring to be effective on GPUs.

The recent advances in machine learning has transformed the way we approach scientific computing. Mishra and Molinaro Mishra and Molinaro [2021] demonstrated that physics-informed neural networks (PINNs) can effectively solve the radiative transfer equation (RTE) with fixed temperature and absorption coefficients, noting their ease of implementation, speed, robustness, and accuracy. Recent research Lu and Wang [2024] has expanded this approach by using physics-informed deep operator networks (PI-DeepONets) to solve parameterized RTEs, achieving significant speedups for 1D radiation problems.

* gratefully acknowledges support from NSF grant DMS-2316011
† corresponding author

D3S3: Data-driven and Differentiable Simulations, Surrogates, and Solvers @ NeurIPS 2024.

The present study extends the work of Lu and Wang [2024] by developing surrogate RTE solvers for multi-dimensional radiation problems. We introduce a new model formulation that solves RTEs with complex boundary conditions. We use PCA to encode input functions into low-dimensional representations, resulting in more compact DeepONet architectures. PCA has been coupled with operator learning in prior literature, see [Bhattacharya et al., 2021] and the references therein, or see [Tang et al., 2021] for a geological sciML example modeling carbon storage. The new methodology is implemented in a flexible and extensible sciML toolkit for training neural surrogate RTE solvers using physics-informed, data-driven, or hybrid loss functions. We utilize the developed library to solve radiation problems in fire safety research.

2 Methods

In CFD simulations, radiation is incorporated into the energy equation as a volumetric radiative heat loss term. This requires solving the radiative intensity field $I(\boldsymbol{x}, \boldsymbol{s})$ at various locations $\boldsymbol{x}$ and directions $\boldsymbol{s}$ from the radiative transport equation (RTE), given by

$$\mathcal{R}_{\mathrm{RTE}}(I, (\boldsymbol{x}, \boldsymbol{s})) = \boldsymbol{s} \cdot \nabla I(\boldsymbol{x}, \boldsymbol{s}) + \kappa(\boldsymbol{x}) I(\boldsymbol{x}, \boldsymbol{s}) - \kappa(\boldsymbol{x}) I_b(\boldsymbol{x}) = 0, \tag{1}$$

subject to the boundary condition imposed for radiation rays emitting from bounding surfaces to the radiatively participating media within the domain

$$\mathcal{R}_{\mathrm{BC}}(I, (\boldsymbol{x}, \boldsymbol{s})) = \epsilon(\boldsymbol{x}) I_b(\boldsymbol{x}) + \frac{\rho^d(\boldsymbol{x})}{\pi} \int I(\boldsymbol{x}, \boldsymbol{s}') |\boldsymbol{n}(\boldsymbol{x}).\boldsymbol{s}'| d\Omega + \rho^s(\boldsymbol{x}) I(\boldsymbol{x}, \boldsymbol{s}_s) - I(\boldsymbol{x}, \boldsymbol{s}) = 0. \tag{2}$$

$\boldsymbol{n}$ denotes the boundary surface normal, $\boldsymbol{s}_s = \boldsymbol{s} - 2\boldsymbol{n}(\boldsymbol{n}.\boldsymbol{s})$ is the direction of a specular reflection, and $d\Omega$ is the solid angle on the unit sphere Ω.

The input functions of the RTE are the absorption coefficient $u_1 (= -\kappa) \in \mathcal{U}_1$, the black body emissive power $u_2 (= I_b) \in \mathcal{U}_2$, the surface emissivity $u_3 (= \epsilon) \in \mathcal{U}_3$, the diffusive reflection coefficient $u_4 (= \rho^d) \in \mathcal{U}_4$, and the specular reflection coefficient $u_5 (= \rho^s) \in \mathcal{U}_5$. Here $I_b(\boldsymbol{x}) = \sigma / \pi T^4(\boldsymbol{x})$ where T is the temperature and σ is the Stefan–Boltzmann constant. Boundary operators are generalized and parameterized, whereas prior work Lu and Wang [2024] assumed black walls at fixed temperatures. We use a DeepONet to approximate the RTE solution operator $G : \mathcal{U}_1 \times \mathcal{U}_2 \times \mathcal{U}_3 \times \mathcal{U}_4 \times \mathcal{U}_5 \to \mathcal{I}$ which maps input functions to the solution $I(\boldsymbol{x}, \boldsymbol{s}) \in \mathcal{I}$ as

$$G(u_1, \ldots, u_5) = I. \tag{3}$$

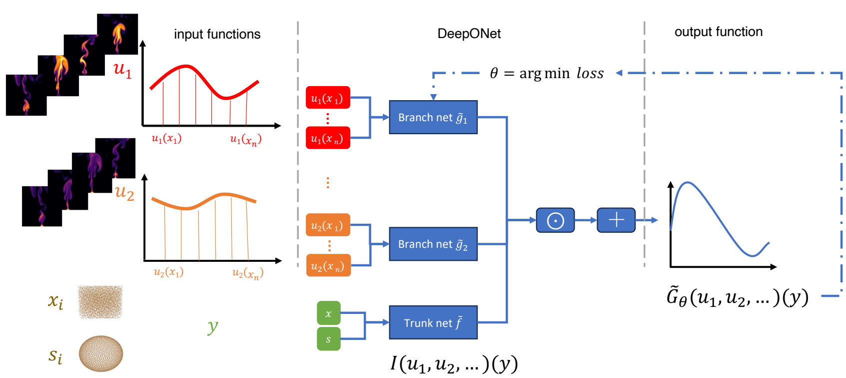

Figure 1: The architecture of DeepONet for learning the mapping between function spaces. We only show two of the five random coefficients for simplicity.

DeepONet comprises two main components: the branch net and the trunk net (see Lu et al. [2021]), as illustrated in Figure 1. As shown in the leftmost panel, the branch nets $g_1, \ \ldots \ g_5$ encode input functions $\boldsymbol{u}_1(\boldsymbol{x}), \ldots \boldsymbol{u}_5(\boldsymbol{x})$ at a fixed number of discrete sensor locations $x_i, \ i = 1, \ldots, n$. The trunk net f encodes the location and direction $\boldsymbol{y} = (\boldsymbol{x}, \boldsymbol{s})$ where the output function $G(\boldsymbol{u}_1, \ldots \boldsymbol{u}_5)(\boldsymbol{y})$ is evaluated within a given domain. The multiple-input DeepONet prediction is

$$\widetilde{G}_\Theta(\boldsymbol{u}^1, \ldots, \boldsymbol{u}^5)(\boldsymbol{y}) = \mathcal{F}_{\Theta_0}(\boldsymbol{y}) \odot \mathcal{G}_{\Theta_1}(\boldsymbol{u}^1) \odot \cdots \odot \mathcal{G}_{\Theta_5}(\boldsymbol{u}^5) + b \tag{4}$$

where $\odot$ is the Hadamard product of the H outputs from each MLP, b is a trainable bias parameter, and $\Theta = \cup_{i=0}^{R} \Theta_i$ aggregates all networks parameters.

The data for training a DeepONet is the Cartesian product of random coefficients $(\boldsymbol{u}_k^1, \ldots, \boldsymbol{u}_k^5)_{k=1}^{N_{\mathrm{RC}}}$ and collocation points $(\boldsymbol{x}_i, \boldsymbol{s}_i)_{i=1}^{N_{\mathrm{C}}}$. The collocation points are input to the trunk net while the random coefficients are sampled at *sensor locations* to get *sensor values* which are input to the branch nets. Our implementation support both aligned datasets, where the sensor locations match the collocation points, and unaligned datasets. Moreover, we lazily construct batches of the Cartesian product dataset on GPU to significantly reduce the time spent loading data.

We support PCA to reduce the number of sensor values. Auto-encoders for SciML have also shown success for dimensionality reduction Kontolati et al. [2023], but they require additional training for the encoder and are not amenable to distributed evaluation. PCA instead can be done via a simple matrix vector multiplication in a distributed manner when integrated in parallel CFD solvers.

The weighted hybrid loss function combines terms for the RTE (1), boundary condition (2), and data from a traditional solver into

$$\mathcal{L}(\Theta) = \omega_{\mathrm{RTE}} \|\mathcal{R}_{\mathrm{RTE}}(\widetilde{G}_\Theta, \mathcal{D}_{\mathrm{RTE}})\| + \omega_{\mathrm{BC}} \|\mathcal{R}_{\mathrm{BC}}(\widetilde{G}_\Theta, \mathcal{D}_{\mathrm{BC}})\| + \omega_{\mathrm{data}} \|\mathcal{R}_{\mathrm{data}}(\widetilde{G}_\Theta, \mathcal{D}_{\mathrm{data}})\|. \quad (5)$$

Here ω are the weights, $\mathcal{R}$ are the residuals, and $\mathcal{D}$ are subsets of the sensor-collocation data with $\mathcal{D}_{\mathrm{data}}$ also containing solution data from the reference solver. The gradient in (1) is evaluated exactly using automatic differentiation, while the integral in (2) is approximated using either Gauss-Legendre quadrature or Quasi-Monte Carlo Niederreiter [1992]. These cubature routines are also used to infer the incident radiation, radiative heat flux, or radiative heat loss.

The core functionalities of the developed sciML library are implemented into two abstract classes. The first constructs the sensor-collocation datasets $\mathcal{D}$ while the second defines the loss function $\mathcal{L}(\Theta)$ by implementing both the residuals $\mathcal{R}$ and the DeepONet $\widetilde{G}_\Theta$. The complete code will be made available upon publications.

3 Numerical Experiments

Numerical experiments are conducted using the developed sciML library `RTENet`, showcasing trained neural surrogates solving RTEs with complex boundary conditions and in practical settings of fire radiation transfer. The selected test problems include a cylinder enclosure problem from [Chui et al., 1992, Fig. 5], the four special cases considered in [Ge et al., 2016, Section 4.3], and the small pool fire case from `FireFOAM/tutorials`.

DeepONet training hyperparameters are summarized below where L_T and L_B are the number of hidden layers in trunk and branch nets respectively, γ is the learning rate, B is the equal batch size across datasets, and E is the number of epochs. Note that the number of network parameters $|\Theta|$ is also a function of the number of sensor locations which is described in the following paragraphs. All training uses the AMSGrad variant of the Adam optimizer.

	dataset			DeepONet				training		
problem	coefficients	N_{RC}	N_{C}	$\|\Theta\|$	L_T	L_B	H	γ	B	E
Chui et al. [1992]	(κ)	200	12K	166K	5	4	128	10^{-3}	2K	500
Ge et al. [2016]	$(\epsilon, \rho^d, \rho^s)$	259	2K	150K	4	3	128	10^{-3}	1K	700
pool fire	(κ, I_b)	4000	365K	3.1M	5	4	512	10^{-4}	33K	12

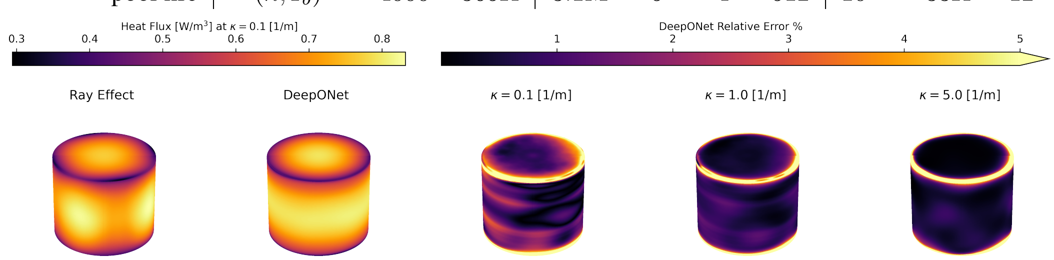

Figure 2: DeepONets overcome the ray effect and yield small pointwise relative errors.

A first test solves gas radiation transfer in a cylindrical enclosure. Boundaries are cold black walls and gas temperature is constant. The DeepONet was trained with $N_{\mathrm{RC}} = 200$ random realizations of constant κ between $\kappa = 0.01\,[\mathrm{m}^{-1}]$ and $\kappa = 6\,[\mathrm{m}^{-1}]$ and no traditional solver data was provided i.e.

this was a purely physics-informed training. L_2 relative errors of 1.53%, 1.17%, and 1.88% were attained for $\kappa = 0.1$, $\kappa = 1$, and $\kappa = 5$ respectively as compared to the analytic gray gas solution. The pointwise relative errors are shown in Figure 2. The discrete ordinate method in FireFOAM, when the angular discretization is coarse, suffers from the ray effect, which is quickly mitigated in DeepONet.

A second test problem in [Ge et al., 2016, Section 4.3] considers gas radiation transfer in a rectangular domain with mixed boundary conditions. The gas temperature is prescribed and the absorption coefficient is constant $\kappa = 0.5\,[\mathrm{m}^{-1}]$, while ϵ, ρ^d, ρ^s are unknown piecewise constant functions satisfying $\epsilon + \rho^d + \rho^s = 1$. The top and bottom boundary have fixed $\epsilon = 1$ while the left and right boundary are random constants ϵ_{LR}, ρ^d_{LR}, and ρ^s_{LR}. The DeepONet is trained on $N_{\mathrm{RC}} = 259$ random realizations of ϵ_{LR}, ρ^d_{LR}, and ρ^s_{LR}, and tested with the parameters specified in [Ge et al., 2016, Section 4.3]. Figure 3 compares the predicted volumetric heat loss along a vertical slice of the domain between DeepONet and the P_7 model from Ge et al. [2016].

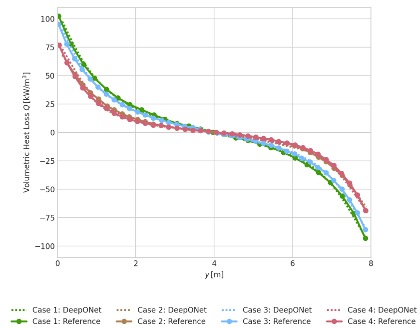

Figure 3: DeepONet predicted heat loss compared to the higher order spherical harmonics P_7 models in Ge et al. [2016].

The third problem trains DeepONet on $N_{\mathrm{RC}} = 4\mathrm{K}$ FireFOAM pool fire simulation snapshots, which use 151×151 cells and 16 discrete ordinates. The high-dimensional datasets are encoded into lower-dimensional latent spaces using PCA, reducing the branch net input size from 23K to 500. Figure 4 compares predicted incident radiation results on withheld testing realizations. Notably, the DeepONet approximation does not suffer from the ray effect observed in the discrete ordinate solver.

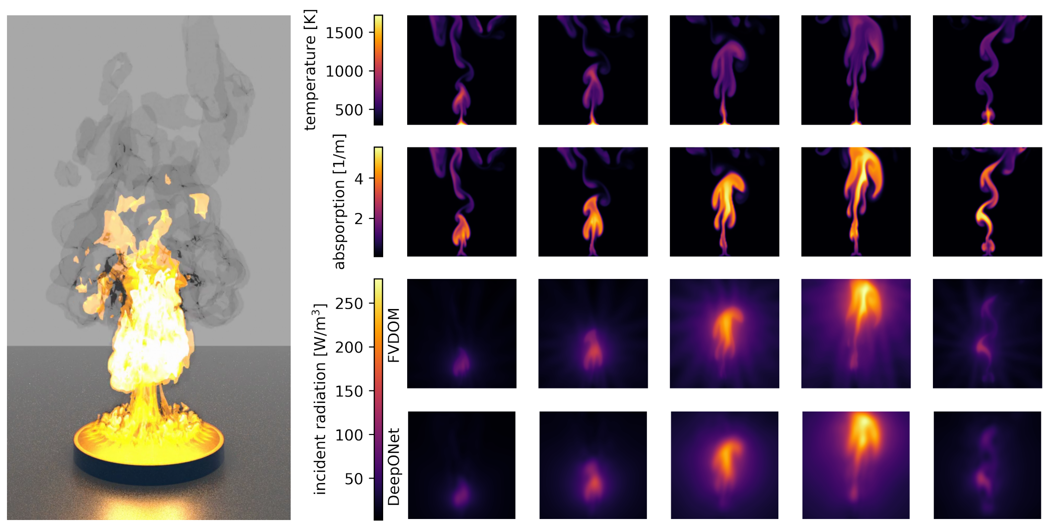

Figure 4: Comparison of predicted incident radiation from temperature and absorption in a pool fire between the DeepONet and a reference Finite Volume Discrete-Ordinate Method (FVDOM) solver.

4 Conclusions

This work introduces a new ML toolkit for training neural surrogates to solve radiation transfer. Users can select from 1, 2, or 3D geometries and random coefficients from the governing equation or boundary condition. PCA can be optionally used to reduce the branch network input sizes for more compact DeepONet architectures. The SciML models are trained using physics-informed, data-driven, or hybrid loss functions. While this article only describes DeepONets, the implementation also supports PINNs. The next step will focus on evaluating performance and deploying neural surrogate solvers in CFD of large-scale fires; this includes three dimensional fire simulations which are immediately supported in the described framework.

References

Kaushik Bhattacharya, Bamdad Hosseini, Nikola B Kovachki, and Andrew M Stuart. Model reduction and neural networks for parametric pdes. *The SMAI journal of computational mathematics*, 7: 121–157, 2021.

EH Chui, GD Raithby, and PMJ Hughes. Prediction of radiative transfer in cylindrical enclosures with the finite volume method. *Journal of Thermophysics and Heat transfer*, 6(4):605–611, 1992.

Wenjun Ge, Michael F Modest, and Somesh P Roy. Development of high-order pn models for radiative heat transfer in special geometries and boundary conditions. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 172:98–109, 2016.

Katiana Kontolati, Somdatta Goswami, George Em Karniadakis, and Michael D Shields. Learning in latent spaces improves the predictive accuracy of deep neural operators. *arXiv preprint arXiv:2304.07599*, 2023.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nat. Mach. Intell.*, 3(3):218–229, 2021.

Xiaoyi Lu and Yi Wang. Surrogate modeling for radiative heat transfer using physics-informed deep neural operator networks. *Proceedings of the Combustion Institute*, 40(1-4):105282, 2024.

Siddhartha Mishra and Roberto Molinaro. Physics informed neural networks for simulating radiative transfer. *J Quant Spectrosc Radiat Transf.*, 270:107705, 2021.

Harald Niederreiter. *Random number generation and quasi-Monte Carlo methods*. SIAM, 1992.

Hewei Tang, Pengcheng Fu, Christopher S Sherman, Jize Zhang, Xin Ju, François Hamon, Nicholas A Azzolina, Matthew Burton-Kelly, and Joseph P Morris. A deep learning-accelerated data assimilation and forecasting workflow for commercial-scale geologic carbon storage. *International Journal of Greenhouse Gas Control*, 112:103488, 2021.

Yi Wang, Prateep Chatterjee, and John L de Ris. Large eddy simulation of fire plumes. *Proceedings of the Combustion Institute*, 33(2):2473–2480, 2011.

Yi Wang, Karl V Meredith, Xiangyang Zhou, Prateep Chatterjee, Yibing Xin, Marcos Chaos, Ning Ren, and Sergey B Dorofeev. Numerical simulation of sprinkler suppression of rack storage fires. *Fire Safety Science*, 11:1170–1183, 2014.