XLVI CILAMCE
VITÓRIA - ES - BRAZIL - 2025
XLVI IBERO-LATIN AMERICAN CONGRESS ON COMPUTATIONAL METHODS IN ENGENEERING

Performance Assessment on Neural Operator Surrogate Models For Porous Media Flows

Gabriel F. Barros[1], Rômulo M. Silva[1], Amanda C. N. de Oliveira[2], Ezequiel S. Santos[1], Rodolfo S. M. Freitas[3], Fernando A. Rochinha[3], Alvaro L. G. A. Coutinho [1]

[1]*Civil Engineering Program - COPPE/UFRJ*
Rio de Janeiro, Brazil
{gabriel.barros}, {romulo.silva}, {ezequiel.santos}, {alvaro}@coc.ufrj.br
[2]*Computer Science Program - COPPE/UFRJ*
Rio de Janeiro, Brazil
amandacno@cos.ufrj.br
[3]*Mechanical Engineering Program - COPPE/UFRJ*
Rio de Janeiro, Brazil
faro@mecanica.coppe.ufrj.br

Abstract.

Although they are resource-demanding for real-world applications, numerical simulations are a powerful tool for investigating physical phenomena. For many applications, especially those involving many-query procedures such as optimization and uncertainty quantification, the extensive use of numerical simulations can be prohibitive due to computational effort. The growing interest in Scientific Machine Learning (SciML) techniques has enabled the construction of data-driven surrogate models that can predict unseen scenarios and reduce dependency on expensive numerical simulations. Amidst the wide variety of SciML models, Fourier Neural Operators (FNOs) emerge as a powerful tool for learning the mapping between functions using spectral convolutions over the problem dimensions as their kernel operations, leading to a more generalizable model for parameter exploitation or time prediction. It is known, however, that high-dimensional tensors are used for data ingestion on FNOs, resulting in undesired spatial complexity and memory footprint when considering these algorithms. In this study, we investigate the computational complexity of using FNOs on a two-phase flow application by identifying bottlenecks and limitations of the existing architectures for porous media flow. PyTorch's Memory Viz profiler is selected as our tool to measure the GPU memory usage, while Weights & Biases is set to instrument the code efficiency. Our surrogate model results are evaluated using SciML metrics for model performance.

Keywords: Fourier Neural Operators, Porous Media Flow, Scientific Machine Learning, High Performance Computing

1 Introduction

High-fidelity numerical simulation of multiphase flow in porous media represents the cornerstone of modern subsurface resource management, from hydrocarbon recovery to geological carbon storage [1]. State-of-the-art simulators, such as Open Porous Media (OPM) Flow, solve complex systems of nonlinear partial differential equations (PDEs) to approximate fluid behavior with high accuracy [2]. The solution process, however, relies on sophisticated numerical techniques that are computationally intensive, particularly for large-scale, 3D heterogeneous models. This immense computational expense makes single forward simulations time-consuming and creates a critical bottleneck for many-query applications, which are essential for decision-making under uncertainty [3, 4]. Tasks such as uncertainty quantification (UQ), history matching, and multi-objective optimization require thousands of forward model evaluations, rendering them prohibitively expensive and limiting the thorough exploration of the design and parameter space [3, 5].

To overcome these computational barriers, scientific machine learning (SciML) has emerged as a transformative paradigm, enabling the development of fast and accurate data-driven proxy models [1, 4]. After learning the patterns obtained from high-fidelity data generated by numerical simulators, these proxies can approximate simulator outputs at a fraction of the computational cost, thereby enabling the many-query applications that were pre-

viously intractable. A particularly powerful and theoretically grounded framework within SciML is that of neural operators, which learn mappings between infinite-dimensional function spaces rather than the finite-dimensional Euclidean spaces typically used in classical neural networks [6]. This property enables them to learn the underlying solution operator of a PDE family, making them not only inherently resolution-invariant but also substantially more generalizable compared to other strategies. The Fourier Neural Operator (FNO) is a highly efficient instantiation of this concept, which parameterizes the integral kernel of the operator directly in Fourier space and leverages the Fast Fourier Transform (FFT) for quasi-linear time complexity [7]. This architecture allows FNOs to achieve prediction speed-ups of several orders of magnitude over traditional solvers while maintaining high accuracy [3, 7].

Despite being generalizable models, FNOs are known to be expensive and hard to train, with a substantial memory footprint [8]. Building upon the current state-of-the-art on FNO surrogate modeling, this paper covers the performance aspects of FNO surrogate models for porous media flow. We assess the performance requirements in terms of space complexity for training and predicting scenarios involving subsurface flow. The remainder of this paper is structured as follows. Section 2 details the governing equations for multiphase flow in porous media and describes the data generation process used to create the training and validation datasets. Section 3 provides a comprehensive overview of the Fourier Neural Operator architecture and the methodology for its implementation. Section 4 presents our proposed workflow for building the surrogate model based on FNO architectures, along with our numerical results and discussions. Finally, Section 5 concludes and outlines directions for future work.

2 Fluid Flow Through Porous Media Problem

Governing Equations

Numerical simulations of porous media flow can differ significantly depending on their degree of complexity and the resources available. For instance, compositional models are largely more complex than simpler models such as the black-oil model. The simulations in this study are performed using the Open Porous Media (OPM) Flow simulator, employing the black-oil model. This model is predicated on a system with three fluid phases (aqueous, oleic, and gaseous) and three corresponding pseudocomponents: water, oil, and gas. The oil and gas components are not single species but rather represent mixtures of hydrocarbon species that exist in liquid and vapor forms at surface conditions, respectively. The model accounts for mass transfer between the oleic and gaseous phases, tracking the amount of dissolved gas in the oleic phase and vaporized oil in the gaseous phase.

The mathematical formulation of the black-oil model is derived from the principle of mass conservation for each pseudocomponent, $\alpha \in \{w, o, g\}$. This results in a system of coupled, non-linear PDEs

$$\frac{\partial}{\partial t}(\phi_{ref} A_\alpha) + \nabla \cdot \mathbf{u}_\alpha + q_\alpha = 0, \tag{1}$$

where the first term represents mass accumulation and the second term describes the flux. The accumulation terms, A_α, are defined as:

$$A_w = m_\phi b_w s_w, \qquad \mathbf{u}_w = b_w \mathbf{v}_w, \tag{2}$$

$$A_o = m_\phi (b_o s_o + r_{og} b_g s_g), \qquad \mathbf{u}_o = b_o \mathbf{v}_o + r_{og} b_g \mathbf{v}_g, \tag{3}$$

$$A_g = m_\phi (b_g s_g + r_{go} b_o s_o), \qquad \mathbf{u}_g = b_g \mathbf{v}_g + r_{go} b_o \mathbf{v}_o. \tag{4}$$

Here, ϕ_{ref} is the reference porosity, m_ϕ is a pressure-dependent multiplier, b_α is the phase shrinkage/expansion factor, s_α is the phase saturation, and r_{go} and r_{og} are the mass ratios of dissolved gas in oil and vaporized oil in gas, respectively. The components velocities, $\mathbf{u}_\alpha$, are related to the phase fluxes, $\mathbf{v}_\alpha$, which is governed by the multiphase extension of Darcy's law,

$$\mathbf{v}_\alpha = -\lambda_\alpha \mathbf{K}(\nabla p_\alpha - \rho_\alpha \mathbf{g}) \tag{5}$$

where $\mathbf{K}$ is the absolute permeability tensor, λ_α is the phase mobility (relative permeability divided by viscosity), p_α is the phase pressure, ρ_α is the phase density, and $\mathbf{g}$ is the gravitational acceleration vector.

To close the system, two additional algebraic constraints are required. The first is that the phase saturations must sum to unity within the pore volume, and the second relates the phase pressures through capillary pressure, p_c, which is a function of saturation, that is,

$$s_w + s_o + s_g = 1, \tag{6}$$

$$p_{c,ow}(s_w) = p_o - p_w, \tag{7}$$

$$p_{c,og}(s_g) = p_g - p_o. \tag{8}$$

The OPM Flow simulator solves this system of equations using a fully implicit numerical scheme. The spatial domain is discretized with an upwind finite-volume method, while time is discretized using an implicit (backward) Euler scheme. The resulting large system of non-linear algebraic equations is solved simultaneously at each time step using a Newton-Raphson linearization method coupled with a preconditioned iterative linear solver. More details on how OPM Flow approximates the black-oil model can be found in [2].

Data Generation

To assess the surrogate models' performance, we choose the 10th Comparative Solution Project (CSP) [9] of the Society of Petroleum Engineers (SPE), more commonly known as SPE10, as our benchmark problem. SPE10 comprises two problems: a smaller 2D model (SPE10 Model 1) and a larger 3D model (SPE10 Model 2). Although SPE10 was proposed to test competitors' abilities regarding upscaling and pseudoization on reservoir models, we use the fine-scale input decks available for both models. In this study, our results focus solely on SPE10 Model 1.

SPE10 Model 1 is a two-phase (oil and gas) model with no dipping or faults. The problem contains two wells: injection and production wells are positioned in the first and last cell columns of the domain (from left to right), respectively. The dimensions of the model are 2500ft long $\times$ 25ft wide $\times$ 50ft thick on a $100 \times 1 \times 20$ mesh. Figure 1 shows the permeability field provided for the problem. For all parameters used in the SPE10 Model 1, we refer to [9]. SPE10 input decks for OPM Flow, although outdated, are available at the OPM Data repository[1].

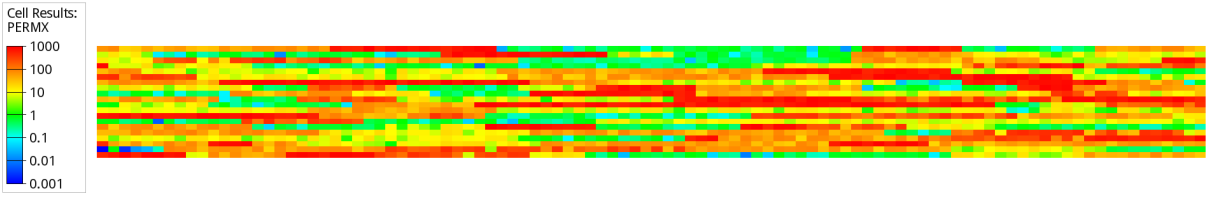


Figure 1. Correlated permeability field in the x direction. Notice that, for this benchmark, the logarithmic scale reveals a variation of six orders of magnitude. The z-axis (vertical) has been exaggerated 5 times.

The original input deck for SPE10 Model 1 provides an injection rate of 0.30 Mscf/day of gas injection. To test the performance of our surrogate model, we generate 100 samples of different injection rates from a uniform distribution $\mathcal{U}(0.23, 0.37)$. Each sample is a different configuration for a numerical simulation, which takes around 5 minutes to run. Figure 2 shows the oil production curves for a portion of the dataset containing 10 samples.

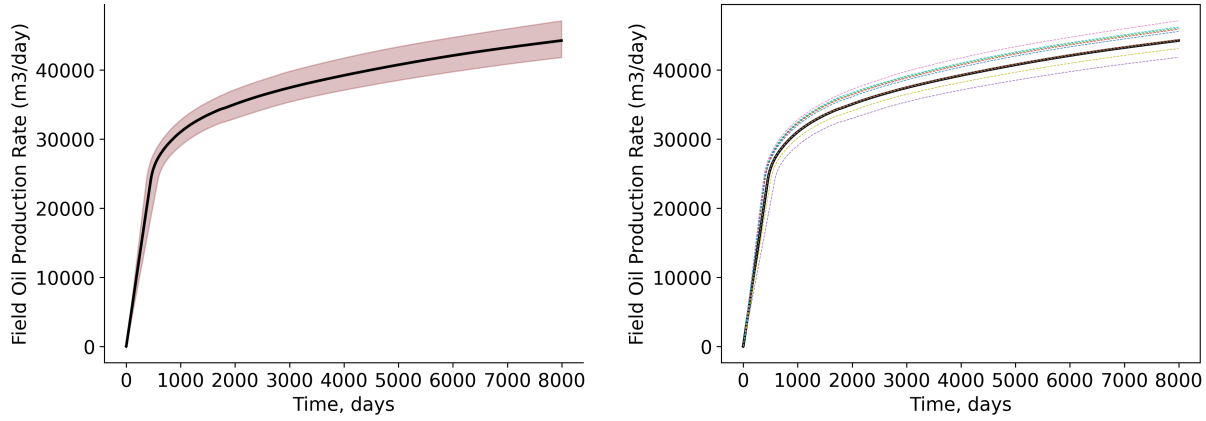


Figure 2. Field oil production for ten simulations on SPE10. The left figure shows the envelope of the generated curve, and the right shows the curves themselves. The black line denotes the original SPE10 Model 1 solution in both cases. The injection rates are generated using a uniform distribution between 0.23 and 0.37.

[1] https://github.com/OPM/opm-data/

3 Fourier Neural Operators and the Current Workflow

We choose Fourier Neural Operators (FNOs) as the primary architecture for our data-driven surrogate model due to their mesh-free formulation, strong performance across various PDEs (including Navier–Stokes and multiphase flow problems [10, 11]) and their efficient implementation via Fast Fourier Transforms (FFTs), particularly on uniform grids and regular domains [7]. Furthermore, FNOs have been theoretically proven to be universal approximators for continuous operators, capable of approximating any given PDE to arbitrary accuracy [6].

The field of neural operators for fluid dynamics is evolving rapidly, building upon the foundational work of Li et al. [7] who first demonstrated the FNO's efficacy on canonical PDE problems. Subsequent research has focused on architectural enhancements to address the limitations of the original FNO when faced with the complex physics of multiphase flow. For instance, the U-FNO incorporates a U-Net architecture to more effectively capture high-frequency information and sharp saturation fronts, resulting in superior accuracy [11]. Similarly, the Wavelet Neural Operator (WNO) employs wavelet transforms to more effectively model the sharp CO_2 saturation fronts found in large-scale, complex geomodels [3]. These developments highlight a crucial trade-off in operator design: the global, spectral nature of the FNO is often augmented with local-feature extractors to handle non-smooth solutions, sometimes at the expense of computational performance or theoretical properties, such as mesh-invariance. Concurrently, the frontier of generalization is expanding. Recent work by Badawi and Gildin [5] has demonstrated a single FNO surrogate capable of predicting solutions across not only varying permeability fields but also dynamic operational conditions, including changing well locations, controls, and even the number of wells.

FNOs essentially truncate the frequency spectrum of the data in the hidden layers by subsampling the number of frequency modes, thus acting as a low-pass filter [3]. Figure 3 illustrates the FNO architecture. The input field $a(x)$ is lifted to a higher dimension channel space by a neural network P. Then, the lifted input is sequentially submitted to the Fourier layers, where the spectral convolution acts as the kernel operation in the Neural Operator architecture. After T layers, the output is projected to the target dimension by the neural network Q. For this study, we use the neuralop[2] library, designed by the FNO idealizers.

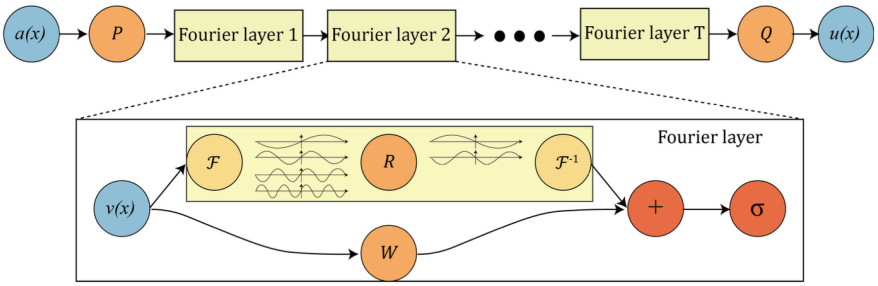


Figure 3. FNO architecture, from [7].

That said, we need to build a proper workflow to handle the data transformations required for fitting the FNO model. The pipeline consists of reading the generated dataset, applying various pre-processing strategies to it, fitting the FNO model, and validating the results on unseen data. Regarding the data structures used in this process, the scheme depicted in Figure 4 provides a more detailed overview of the subject. We split the input and output channels into grid and scalar quantities. Grid quantities refer to quantities described by the cells in the computational grid, such as porosity, permeability, and cell coordinates. Scalar quantities refer to post-processed and consolidated information regarding the field or wells, such as well oil production rates, injection rates, water cuts, and other relevant data. These quantities do not refer to cells in the domain, but rather contain information about the field itself or a particular well. To incorporate this information into our model, we map the injection and production wells and apply these scalar quantities to the well cells, following the approach outlined in [5]. That is, a bit mask is applied to the domain, tagging the cells that belong to a well whose quantity is being considered as a channel. The scalar value at each time step is stamped into the domain on the tagged cells, transforming a scalar quantity into a grid quantity. The same approach is considered on scalar predictions: channels submitted to this procedure get averaged through the bit mask and brought back to a scalar value.

[2] `https://github.com/neuraloperator/neuraloperator`

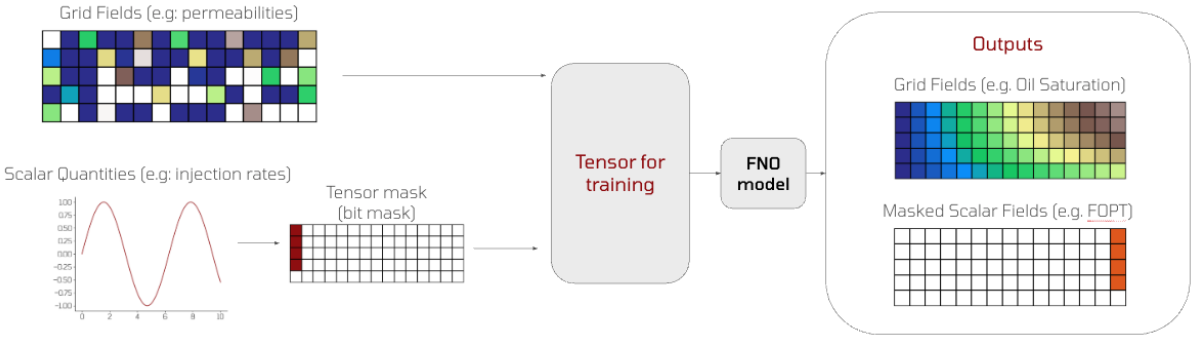


Figure 4. Scheme showing inputs and outputs of the surrogate model for the proposed experiment. The FNO model box refers to the Vanilla FNO and tested variants we have already explored, namely UNO and T-FNO. The models are implemented at the original repository.

4 Numerical Results and Discussion

In this section, we describe our first numerical experiments using Neural Operators. We train the model for several distinct configurations to evaluate its performance. Our preliminary studies examine the model's sensitivity to various parameters, including the number of samples (simulations performed), number of epochs, different FNO variants, and number of output channels. The choice for the latter is not yet unanimous in the literature, given that some authors prefer to develop FNO models that only output one output field. In contrast, others argue that predicting multiple channels simultaneously can be advantageous. After some ablation studies, we proceed with the following hyperparameters:

Table 1. Hyperparameters used in the FNO model

Hyperparameter	Value
Learning rate	0.0008 (adaptive with schedule)
Epochs	200
Modes	32 for all dimensions (1 for the y dimension in the SPE10M1 problem)
Layers	4
Hidden Channels	64
FNO variant	TFNO with Tucker Factorization with 0.05 rank
Number of samples	100
Batch size	4
Train/Test Split	80/20
Input channels	Permeability in $x-$ direction (grid) and Well Gas Injection Rate (scalar)
Output channels	Gas Saturation (grid) and Well Oil Production Rate (scalar)
Comparison metric	Test set L2 error norm

Considering the aforementioned parameters, after the tensor assembly, we can estimate the dimensions of the tensor. We should expect input and output tensors of dimensions $(N_s, N_c, N_x, N_y, N_z, N_t)$, where N_s is the number of numerical simulations spanning the parameter space, N_c is the number of input or output channels, $N_x = 100$, $N_y = 1$ and $N_z = 20$ are the number of grid cells in x, y and z directions (in the case of SPE10 Model 1), and $N_t = 800$ is the number of snapshots in time for each simulated case (one solution per 10 days, during a total simulated time of 8000 days. For both input and output tensors, we have $N_c = 2$. That is, for 100 samples, the expected amount of memory required to store all datasets in the VRAM is $N_s \times N_c \times N_x \times N_y \times N_z \times N_t$, that is, $100 \times (2+2) \times 100 \times 1 \times 20 \times 800 = 6.4 \times 10^8$ components. The model itself contains around 44×10^6 parameters using this setup. Assuming double precision for both parameters and data, the total amount of memory required for this configuration is approximately 40.76 GB, excluding any computational overhead or additional data structures required for training (such as preprocessors, for instance).

Model training is performed for approximately 80 minutes on an NVIDIA H100 with 94 GB of VRAM,

yielding an average loss of 8.70×10^{-3} and a test set L2 error of 6.66×10^{-3}. We present results for a snapshot of a sample from the test set in Figures 5, which compare the ground truth and predictions for grid and scalar quantities, respectively. For grid quantities, we notice good agreement between the ground truth data and the prediction, with spatial errors being negligible in the domain except for a few specific cells. Regarding scalar plots, we observe good agreement between the oil production rate curves.

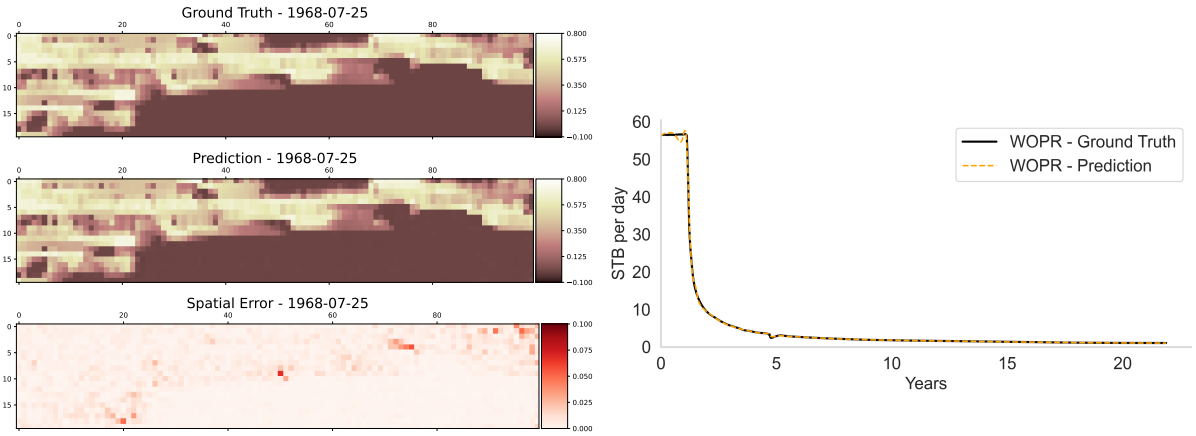


Figure 5. Grid results (left) show the ground truth, prediction and absolute spatial error for the gas saturation (SGAS) channel after 540 simulation days. Scalar results (right) show how the oil production rate curve for the surrogate model approaches the ground truth.

Regarding efficiency, we observe how the GPU is utilized in this problem, as shown in Figure 6. We use two tools for this purpose: Weights and Biases (WandB)[3] and PyTorch's memory profiling tool called Memory Viz[4]. WandB provides a higher-level view, displaying the maximum cumulative memory during training. PyTorch's Memory Viz provides a more detailed visualization of the low-level operations, covering how tensors are allocated during both forward and backward propagation, as well as other important steps such as validation. Figure 6b shows the memory allocation on the GPU for a single epoch. Looking at the time horizon at the bottom of the figure, we notice several blue peaks. Each peak represents a batch undergoing training for the same epoch. That is, for 100 samples on an $80/20$ split, with a batch size of 4, we notice 20 peaks during training. The smaller plateau seen on the time horizon represents the validation step after training one epoch. We notice that both metrics cover approximately the amount of VRAM required for training, which is around 50 GB and 60 GB.

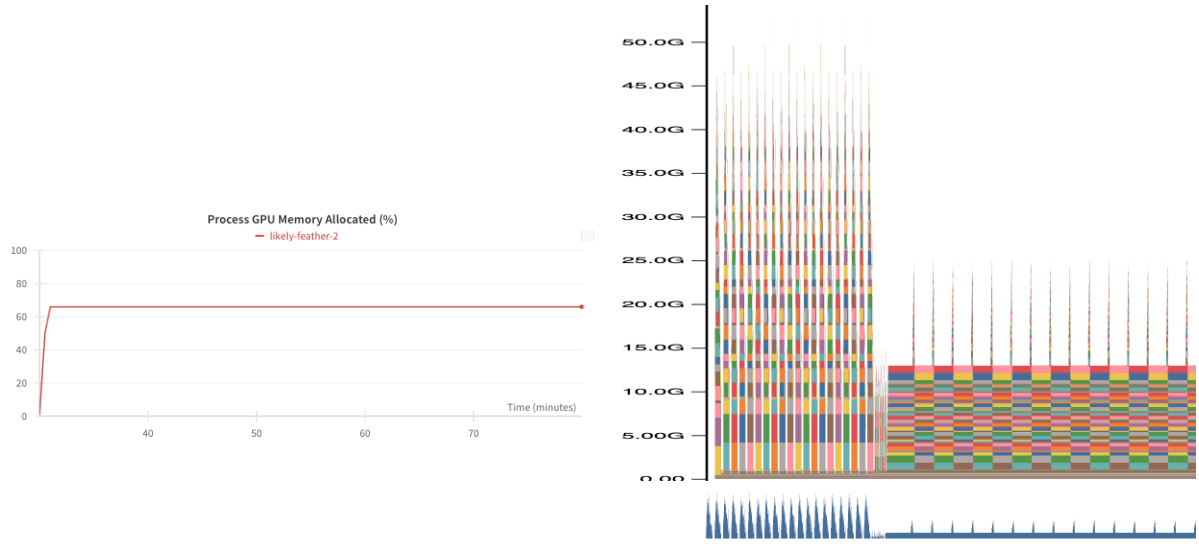
5 Conclusions

Fourier Neural Operators are known for their good generalization in operator learning, making them promising candidates as surrogate models for classical numerical simulations. It is known that FNOs are expensive to train due to their data structure. In this study, we analyze the performance and efficiency of an FNO surrogate for two-phase flow in porous media. We use OPM Flow to generate 100 samples with different injection rates and train an FNO-based surrogate model to generalize for different injection regimes. We noticed that the results are in agreement with the ground truth for the test set, indicating good generalization capabilities for the FNO, for both grid and scalar quantities. Regarding efficiency, we observe that the memory footprint for training an FNO model is substantial, necessitating special attention for scalability purposes.

Acknowledgements. This study was financed in part by CAPES, Brazil - Finance Code 001. This work is also partially supported by FAPERJ, CNPq, and ExxonMobil.

Authorship statement. The authors hereby confirm that they are the sole liable persons responsible for the authorship of this work, and that all material that has been herein included as part of the present paper is either the property (and authorship) of the authors, or has the permission of the owners to be included here.

[3] `https://wandb.ai/`
[4] `https://docs.pytorch.org/memory_viz`



(a) Memory utilization via Weights and Biases for the whole training phase.

(b) Memory utilization via PyTorch Memory Viz for one epoch.

Figure 6. Memory profiling during training.

References

[1] A. Samnioti and V. Gaganis. Applications of machine learning in subsurface reservoir simulation—a review—part i. *Energies*, vol. 16, n. 16, 2023.

[2] A. F. Rasmussen, T. H. Sandve, K. Bao, A. Lauser, J. Hove, B. Skaflestad, R. Klöfkorn, M. Blatt, A. B. Rustad, O. Sævareid, K. A. Lie, and A. Thune. The Open Porous Media Flow reservoir simulator. *Computers and Mathematics with Applications*, vol. 81, pp. 159–185, 2021.

[3] P. A. Witte, T. Konuk, E. Skjetne, and R. Chandra. Fast CO2 saturation simulations on large-scale geomodels with artificial intelligence-based Wavelet Neural Operators. *International Journal of Greenhouse Gas Control*, vol. 126, n. April, pp. 103880, 2023.

[4] A. O. Bocoum and M. R. Rasaei. Multi-objective optimization of WAG injection using machine learning and data-driven Proxy models. *Applied Energy*, vol. 349, 2023.

[5] D. Badawi and E. Gildin. Neural operator-based proxy for reservoir simulations considering varying well settings, locations, and permeability fields. *Computers and Geosciences*, vol. 196, 2025.

[6] N. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. Stuart, and A. Anandkumar. Neural operator: Learning maps between function spaces with applications to pdes. *Journal of Machine Learning Research*, vol. 24, n. 89, pp. 1–97, 2023.

[7] Z. Li, N. B. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. M. Stuart, and A. Anandkumar. Fourier neural operator for parametric partial differential equations. *CoRR*, vol. abs/2010.08895, 2020.

[8] J. Liu, H. Jing, and H. Pan. New fast simulation of 4d (x, y, z, t) co2 eor by fourier neural operator based deep learning method. In *Society of Petroleum Engineers - SPE Reservoir Simulation Conference, RSC 2023*, 2023.

[9] M. A. Christie and M. J. Blunt. Tenth spe comparative solution project: A comparison of upscaling techniques, 2001.

[10] B. Yan, B. Chen, D. Robert Harp, W. Jia, and R. J. Pawar. A robust deep learning workflow to predict multiphase flow behavior during geological CO2 sequestration injection and Post-Injection periods. *Journal of Hydrology*, vol. 607, 2022.

[11] G. Wen, Z. Li, K. Azizzadenesheli, A. Anandkumar, and S. M. Benson. U-FNO—An enhanced Fourier neural operator-based deep-learning model for multiphase flow. *Advances in Water Resources*, vol. 163, n. April, pp. 104180, 2022.