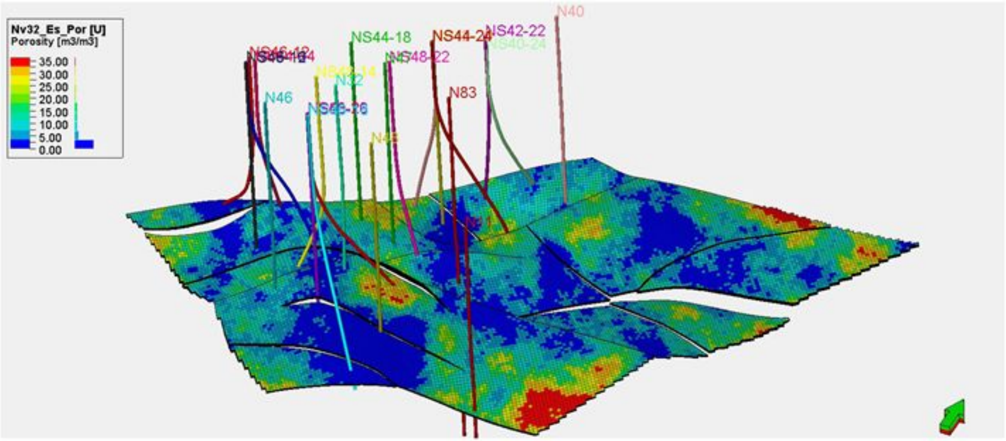
Physics-informed well surrogate model

Avantyristy

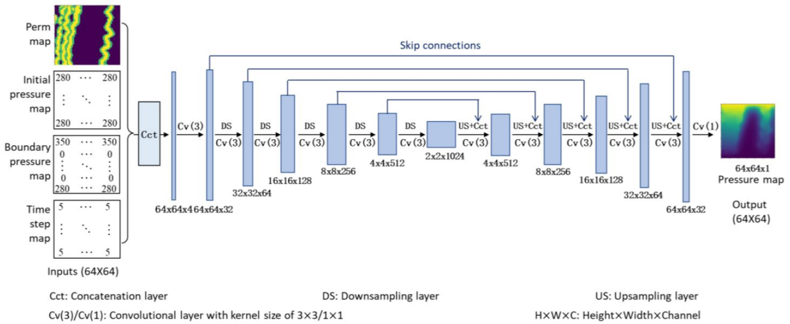
Damir Akhmetov
Daniil Sherki
Egor Cherepanov
Egor Malkershin

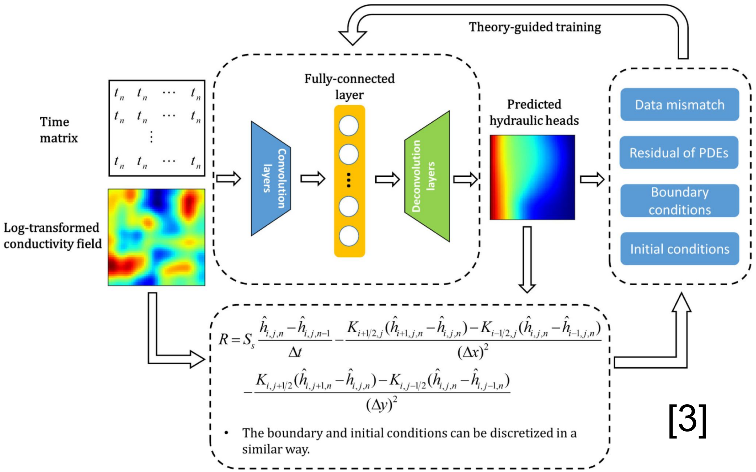
Relevance

An important and urgent task of Petroleum engineering is the modeling of oil and gas reservoirs. Using the diffusivity equation, for example, we can calculate the pressure distribution in the reservoir. However, such simulation is time-consuming and resource-intensive, so there is a reasonable question "How can we speed up the process of obtaining results?”. One approach to solve such a problem is to use surrogate modeling with physically-informed neural networks.

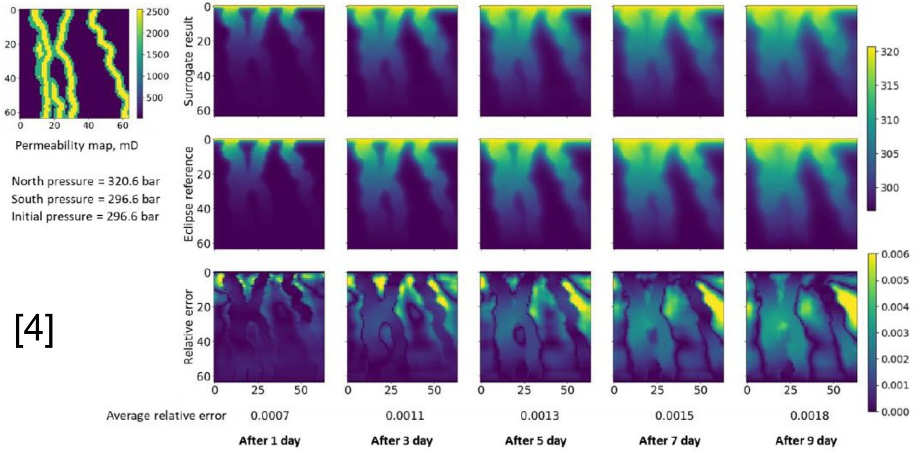


Related work

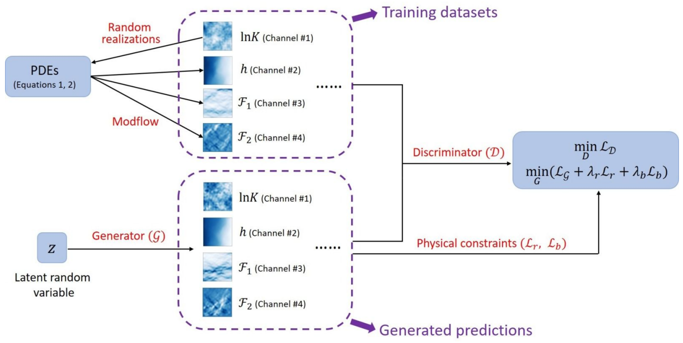




[3]



[4]

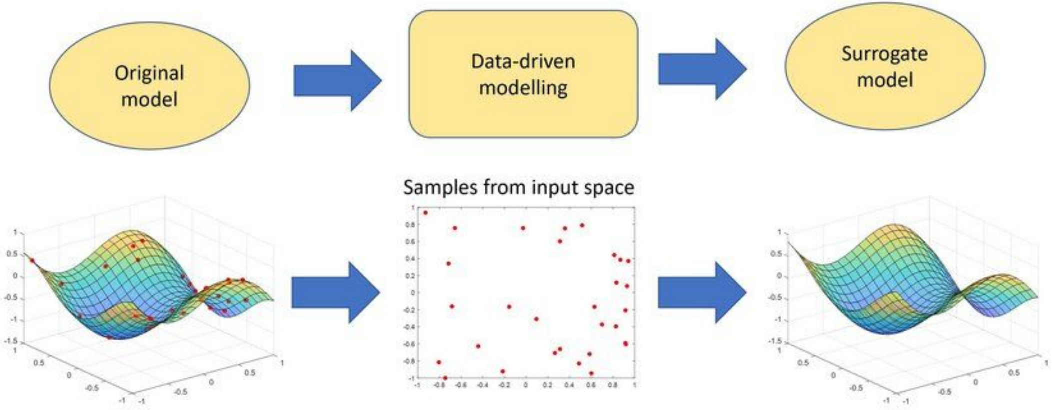


[1]

A few definitions before we start

A **surrogate model** is an engineering method used when an outcome of interest cannot be easily measured or computed, so an **approximate** mathematical model of the outcome is used instead.

Physics-informed neural networks (**PINNs**) are a type of universal function approximators that can embed the knowledge of any physical laws that govern a given data-set in the learning process, and can be described by partial differential equations (PDEs).



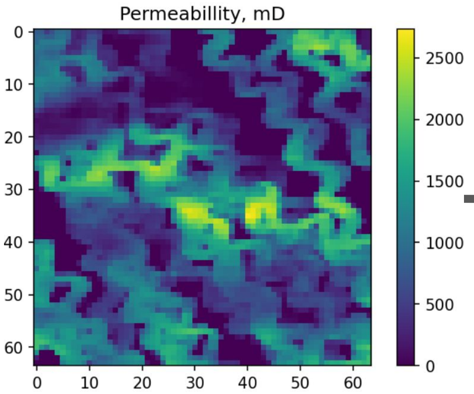
Problem statement

Problem statement: using PINN predict pressure maps (by solving pressure diffusivity equation) after a certain period of time based on Permeability maps for three cases:

- steady pressure diffusivity equation;
- unsteady pressure diffusivity equation;
- steady pressure diffusivity equation with fracture in the well.

Input

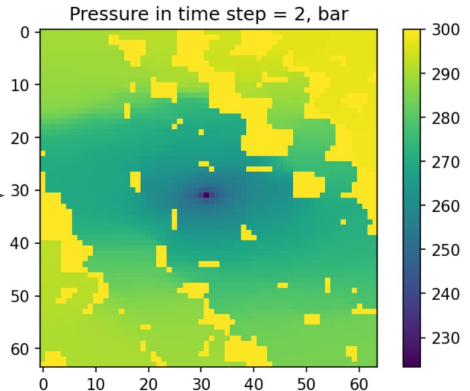
Permeability map



Physics Informed Neural Network

Output

Pressure map



Steady problem

Steady-state problem

Equation:

Hydraulic conductivity Hydraulic head

$$\nabla\cdot[K(\mathbf{x})\nabla h(\mathbf{x})]+q(\mathbf{x})=0, \quad \mathbf{x}\in\mathcal{X}$$

Sink/source term Coordinates

Boundary conditions:

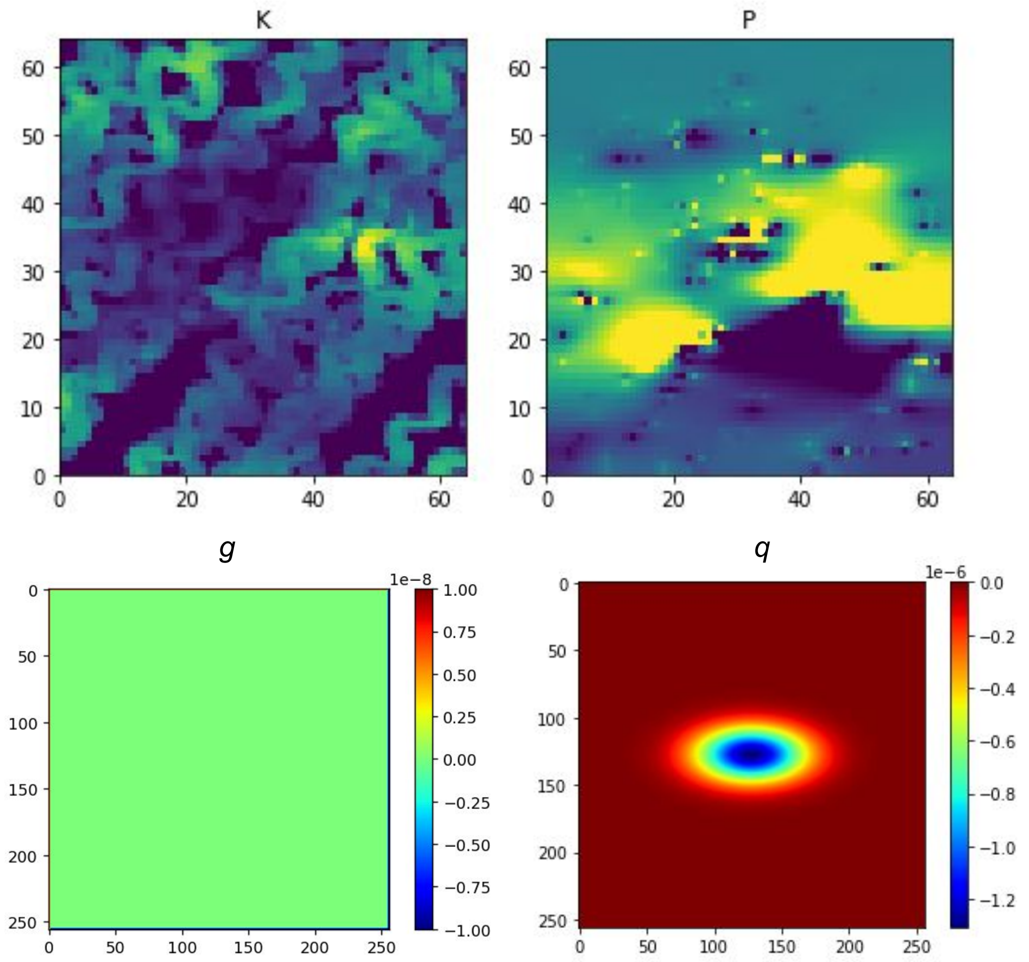
$$h(\mathbf{x}) = h_D(\mathbf{x}),\ \mathbf{x}\in\Gamma_D,$$
$$\nabla h(\mathbf{x})\cdot n = g(\mathbf{x}), \quad \mathbf{x}\in\Gamma_N,$$

In related work:

$$q(\mathbf{x}) = 0$$

In our case it's more complicated:

$$q(\mathbf{x}) = q\delta(\mathbf{x}-\mathbf{x}_w)$$



Dataset generation (steady-state)

Main model properties	Value
Number of cells	64× 64 × 1
Reservoir dimensions, m	12800 × 6400 × 0.25
Permeability, mD (from Brugge dataset)	0 – 2500
Methods	Finite differences + Sparse Matrix Inversion
Total Number of K-P pairs	3643

$$\nabla\left[K(\mathbf{x})\nabla p(\mathbf{x})\right] = q(\mathbf{x})$$

$$\nabla[K\nabla p] = \frac{d}{dx}[K\frac{d}{dx}p] + \frac{d}{dy}[K\frac{d}{dy}p] = \frac{d}{dx}K\frac{d}{dx}p + K\frac{d^2}{dx^2}p + \frac{d}{dy}K\frac{d}{dy}p + K\frac{d^2}{dy^2}p$$

$$\nabla[K\nabla p] = \frac{(K_{i+1,j} - K_{i-1,j})(p_{i+1,j} - p_{i-1,j})}{4dx^2} + K_{i,j}\frac{p_{i+1,j} - 2p_{i,j} + p_{i-1,j}}{dx^2} + \\ + \frac{(K_{i,j+1} - K_{i,j-1})(p_{i,j+1} - p_{i,j-1})}{4dy^2} + K_{i,j}\frac{p_{i,j+1} - 2p_{i,j} + p_{i,j-1}}{dy^2}$$

Determining PI-Loss function

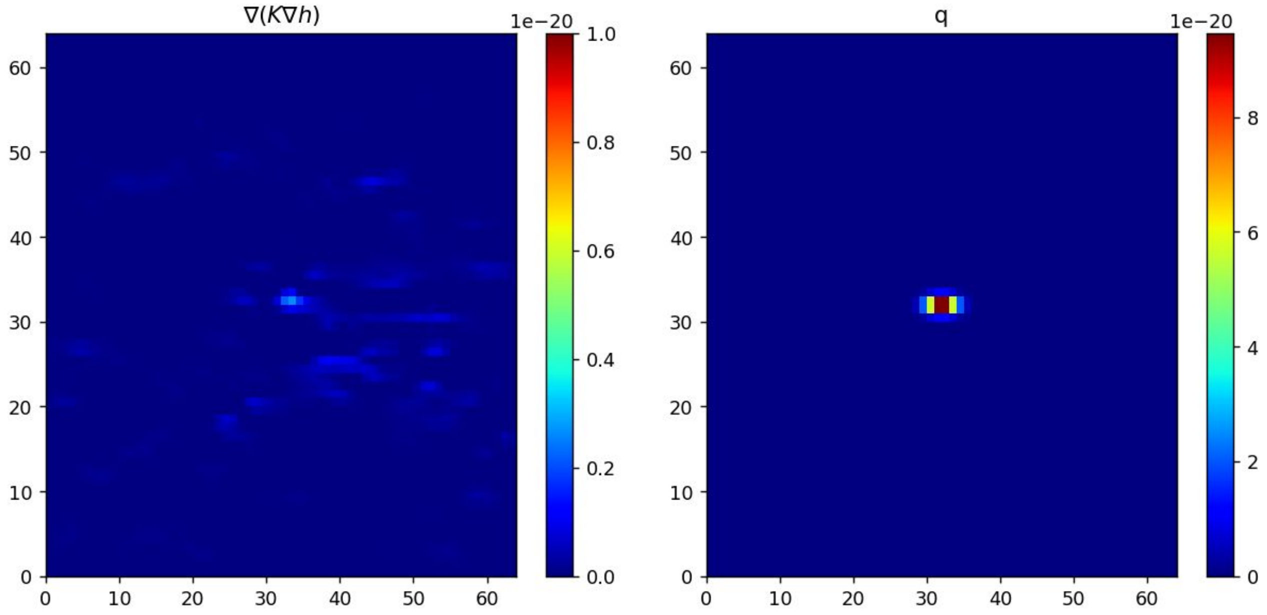
$$\mathcal{L}_{\text{data}}(\theta) = ||\hat{h}(\theta) - h||_2^2$$

$$\mathcal{L}_{\text{PDE}}(\theta) = ||\nabla(K\nabla\hat{h}(\theta)) + q||_2^2$$

$$\mathcal{L}_{\text{NB}}(\theta) = ||K\nabla\hat{h}(\theta)) - g||_2^2$$

$$\mathcal{L}_{\text{total}}(\theta) = \alpha_1 \cdot \mathcal{L}_{\text{data}}(\theta) + \alpha_2 \cdot \mathcal{L}_{\text{PDE}}(\theta) + \alpha_3 \cdot \mathcal{L}_{\text{NB}}(\theta)$$

PDE - loss



-1	0	+1
-2	0	+2
-1	0	+1

Gx

+1	+2	+1
0	0	0
-1	-2	-1

Gy

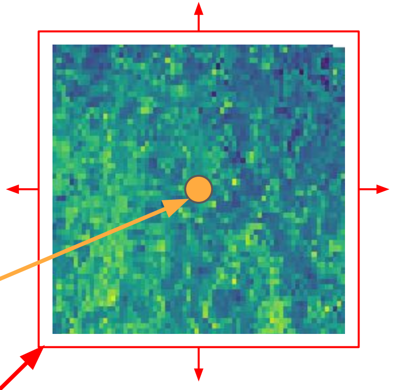
Unsteady-state problem

Problem statement

$$S_s \frac{\partial h(\mathbf{x},t)}{\partial t} + \nabla \left[K(\mathbf{x}) \nabla h(\mathbf{x},t) \right] = q(\mathbf{x})$$

$$\Gamma_D : h(\mathbf{x}) = h_D, \mathbf{x} \in \Gamma_D$$

$$\Gamma_N : K \nabla h(\mathbf{x}) \cdot n(\mathbf{x}) = g(\mathbf{x}), \mathbf{x} \in \Gamma_N$$



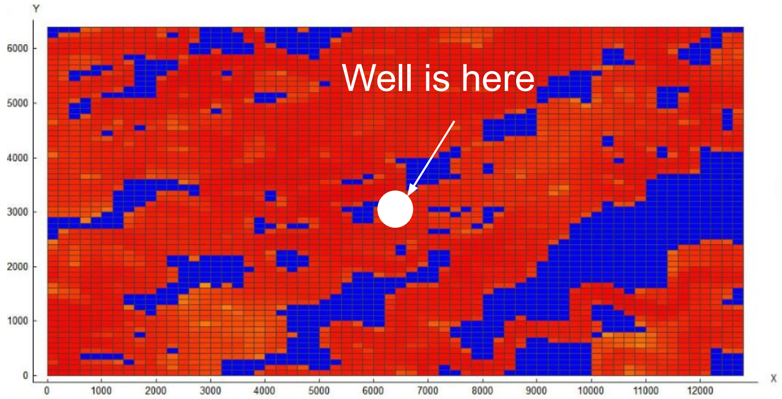
Γ_D: p = const in the well

Γ_N: g(x) = 0 on boundary

Dataset generation (unsteady-state)

Main tNav model properties	Value
Number of cells	64× 64 × 1
Reservoir dimensions, m	12800 × 6400 × 0.25
Initial pore pressure, bar.	300
Bottomhole pressure, bar.	50
Permeability, mD (from Brugge dataset)	0 – 2900
Simulation time, months	10
Number of time steps	10

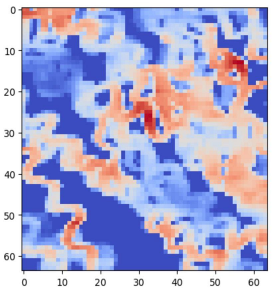
Number of unique tNav models: 3643



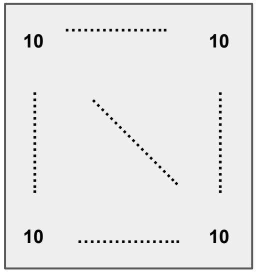
Permeability map tNavigator

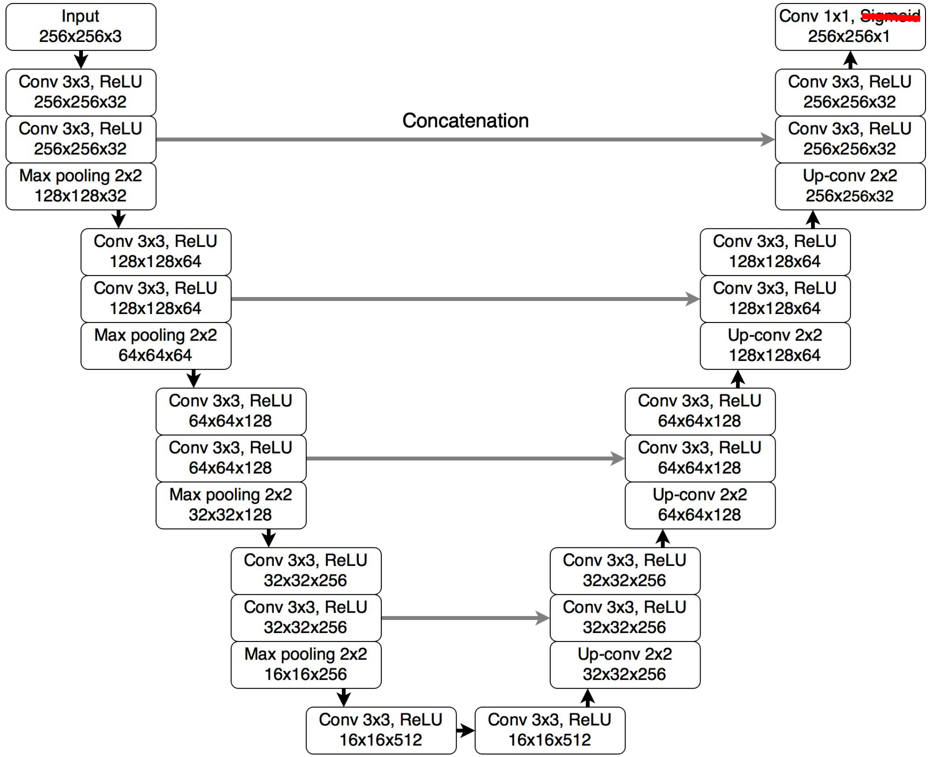
Unet for unsteady-state problem

Perm

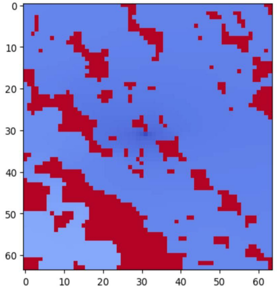


Time step





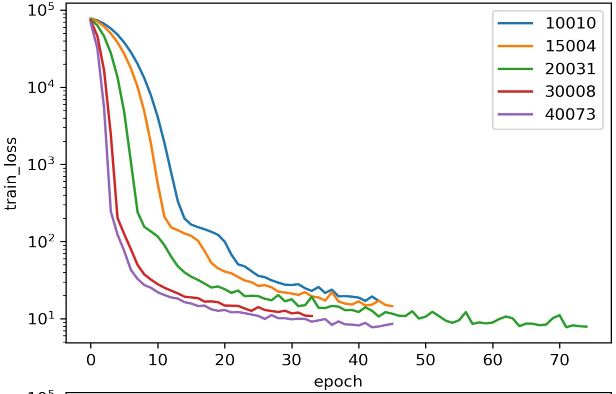
Pressure

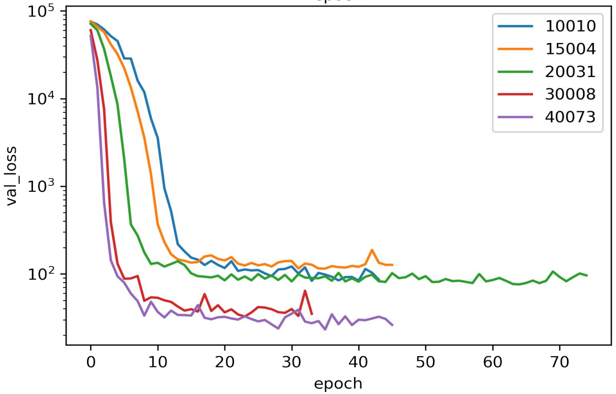


Standard boundary conditions for nonflow at the outer boundary of the reservoir (Neman conditions)

According to the work [4] we can drop activation sigmoid function from the last layer for pressure map generating

Experiments

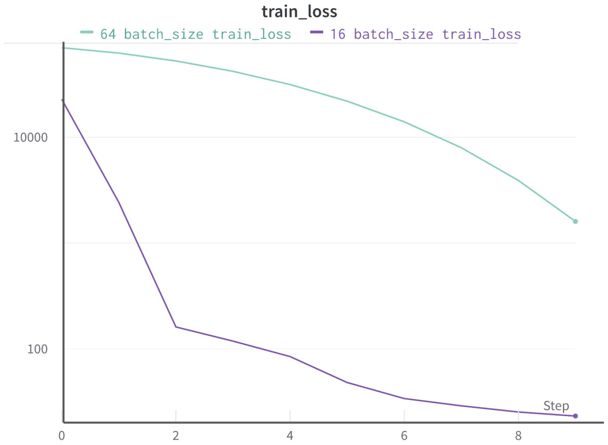




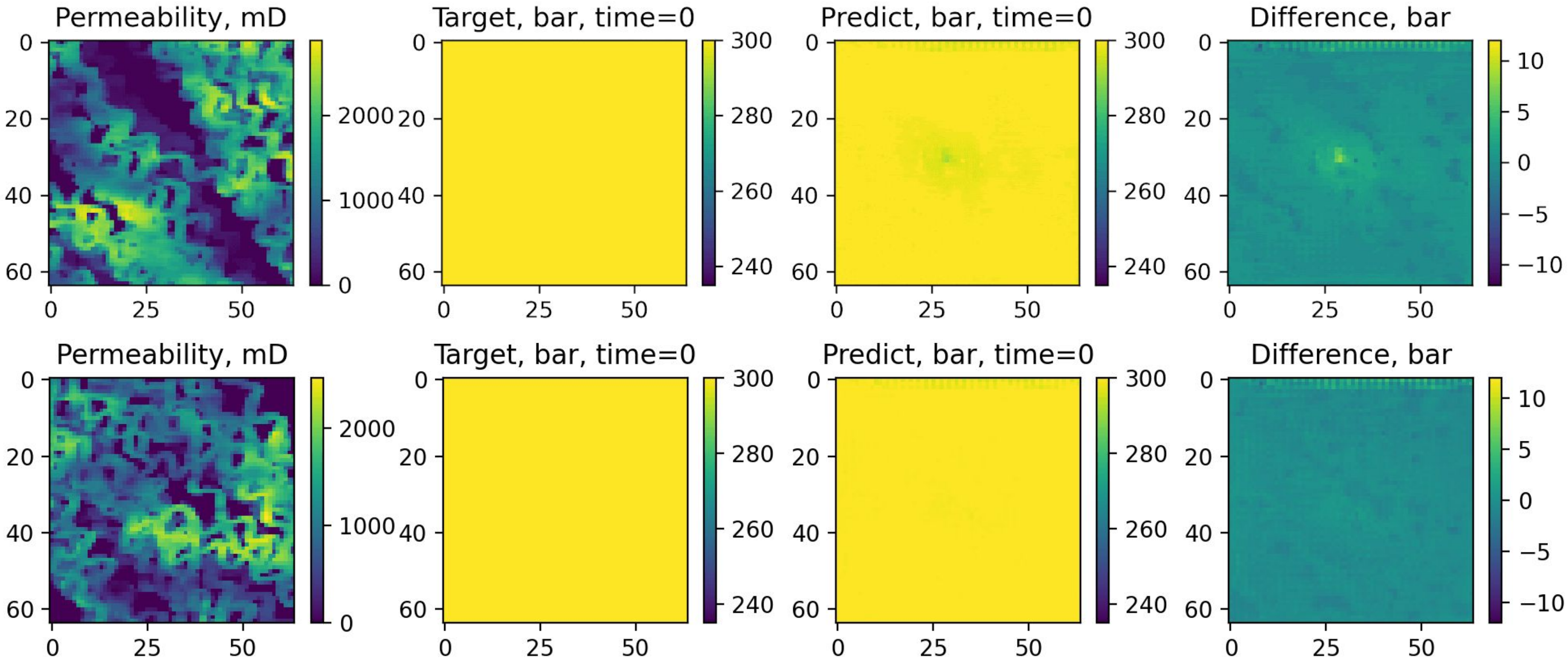
Model training on the different dataset size

Dataset size, sample	RMSE on test dataset
10010	10.27
15004	10.43
20031	9.16
30008	6.45
40073	6.15

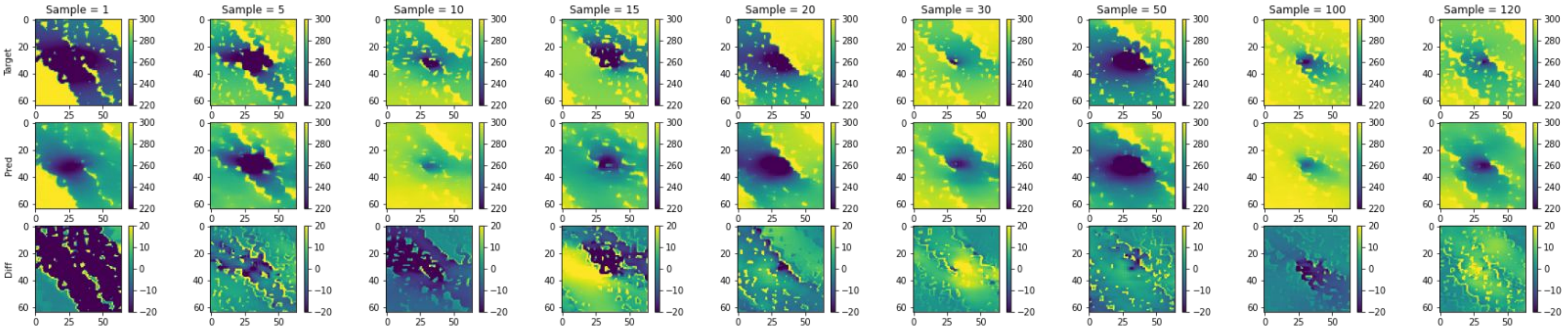
Model training on the different batch size

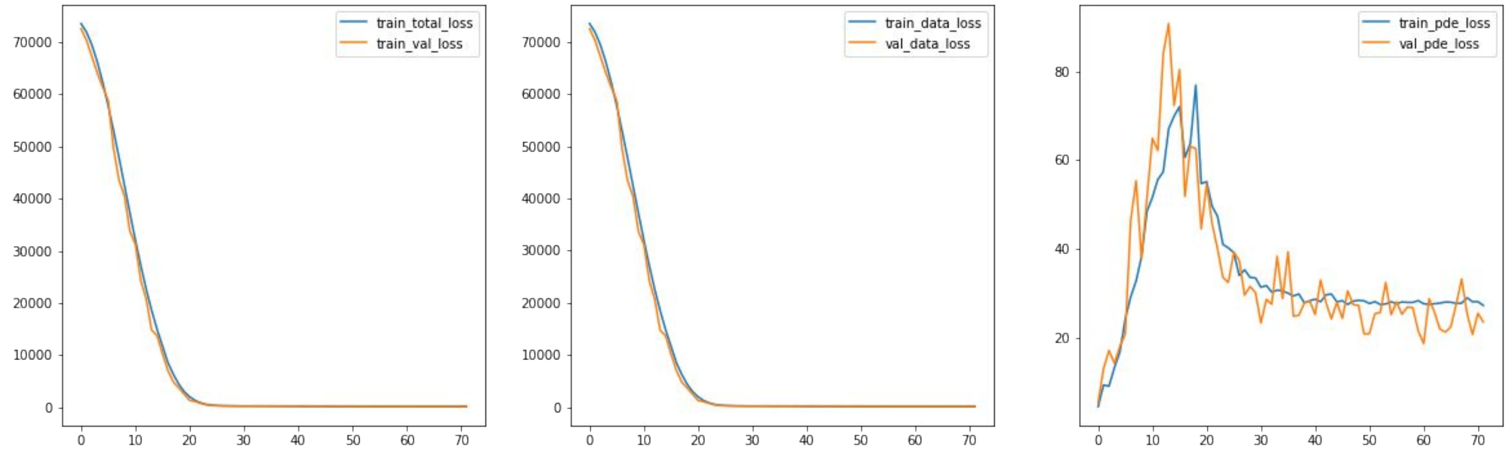


Results

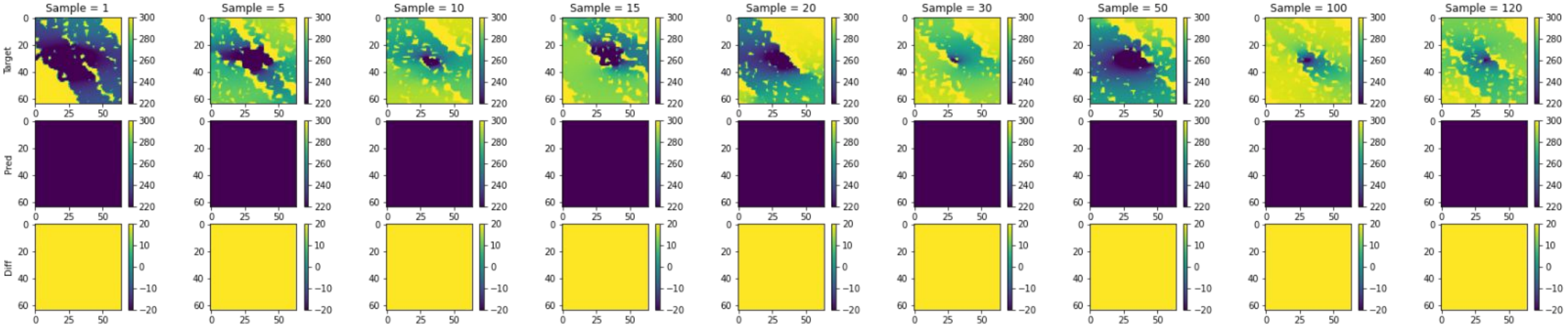

Permeability, mD
Target, bar, time=0
Predict, bar, time=0
Difference, bar
0
20
40
60
0
25
50
2000
1000
300
280
260
240
10
5
0
−5
−10

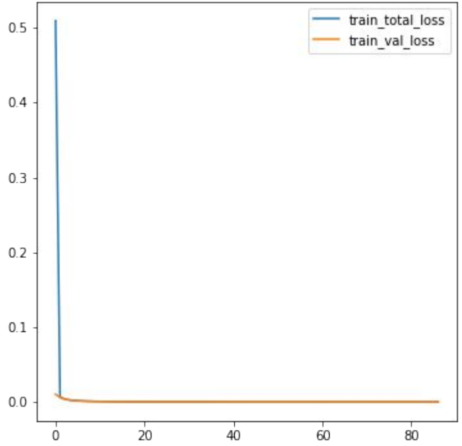
Training on PDE+Data loss

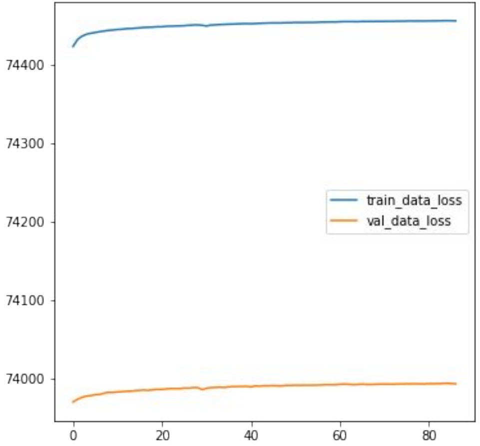


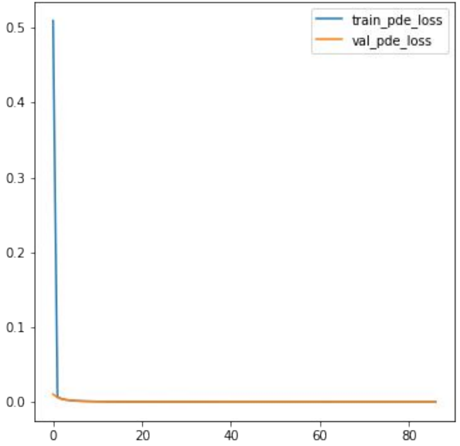


Training on PDE loss



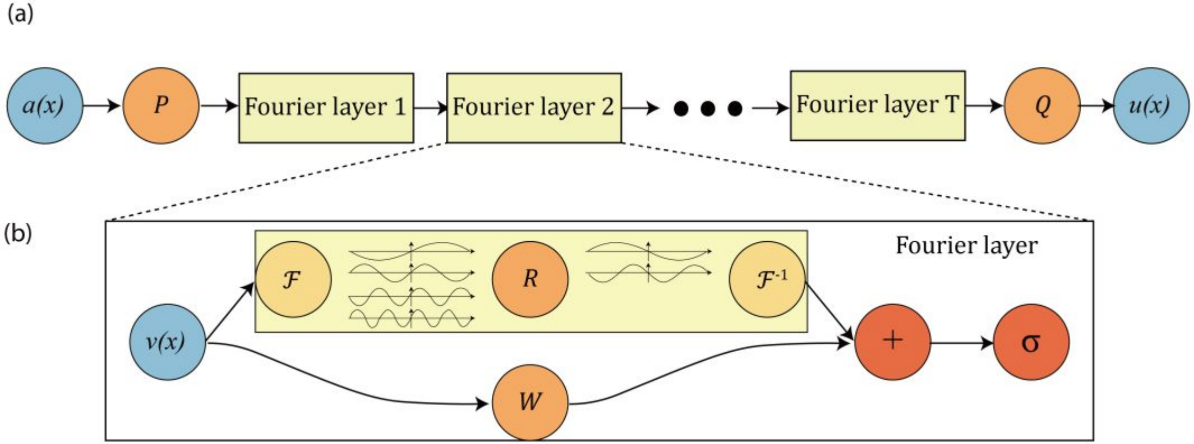


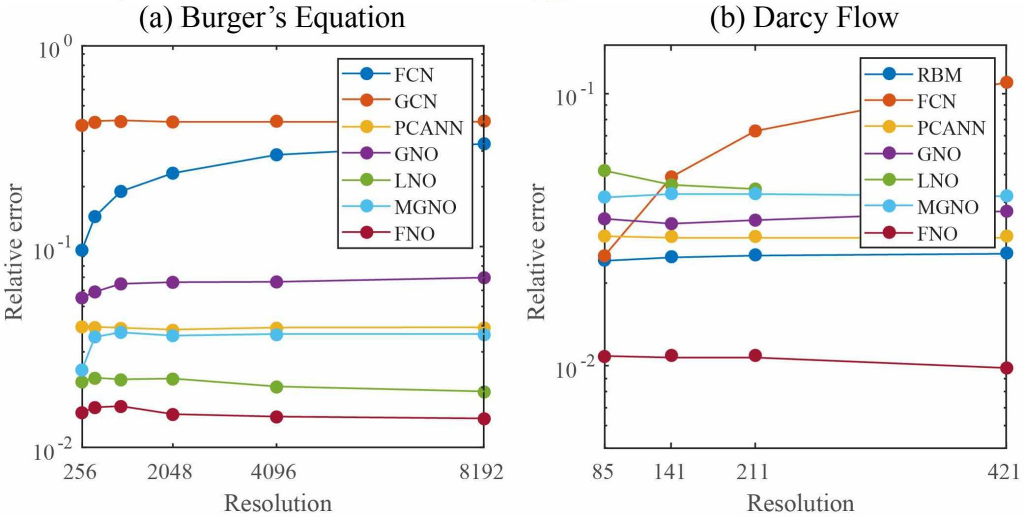




FNO

FNO



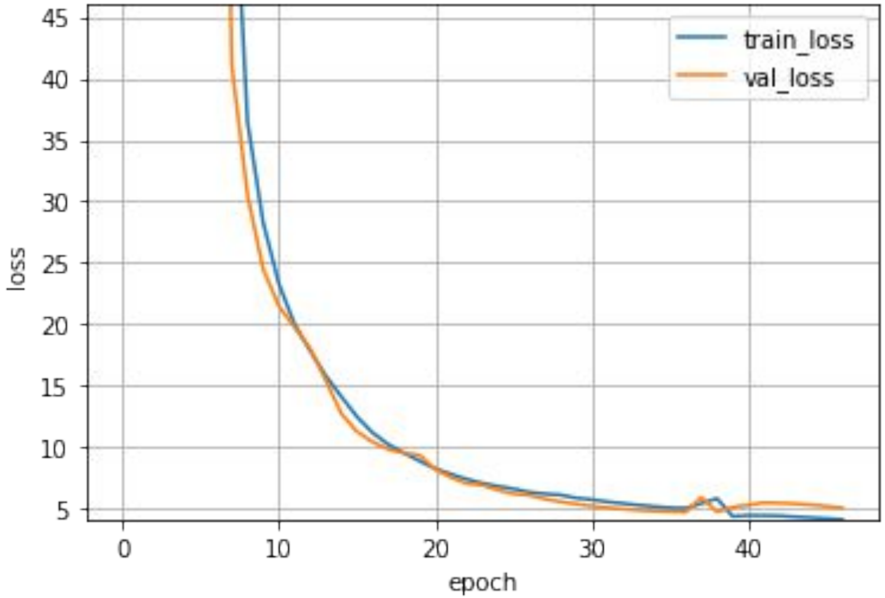


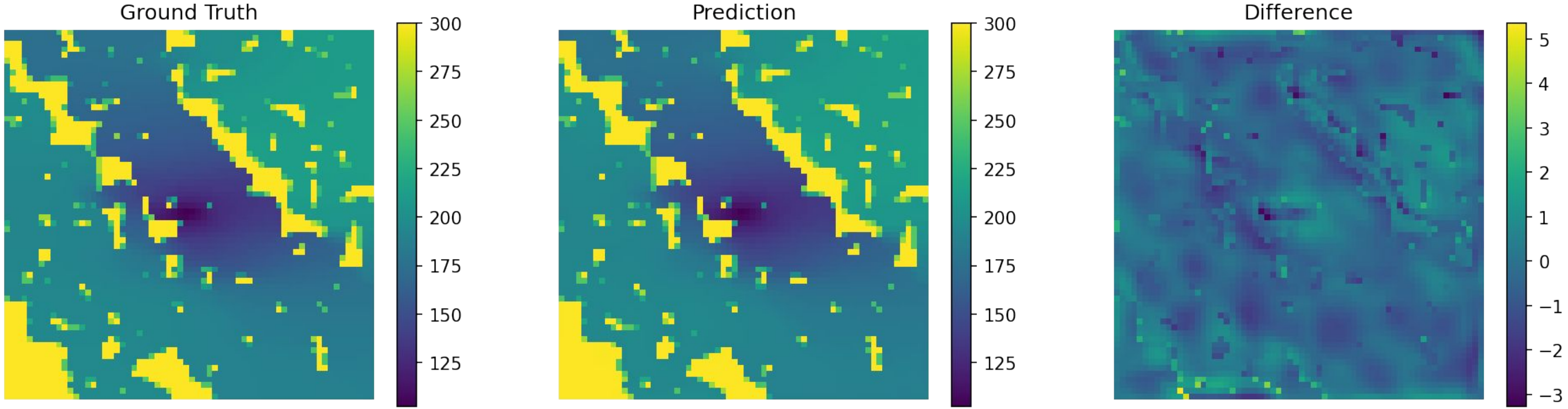
The Fourier layer consists of three main steps:

1. Fourier transform;
2. Linear transform on the lower Fourier modes (low-pass filter);
3. Inverse Fourier transform;

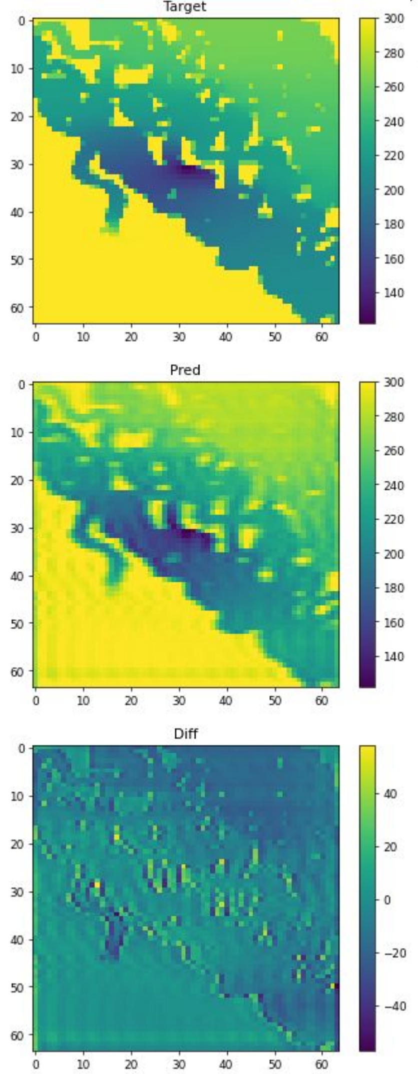
FNO results

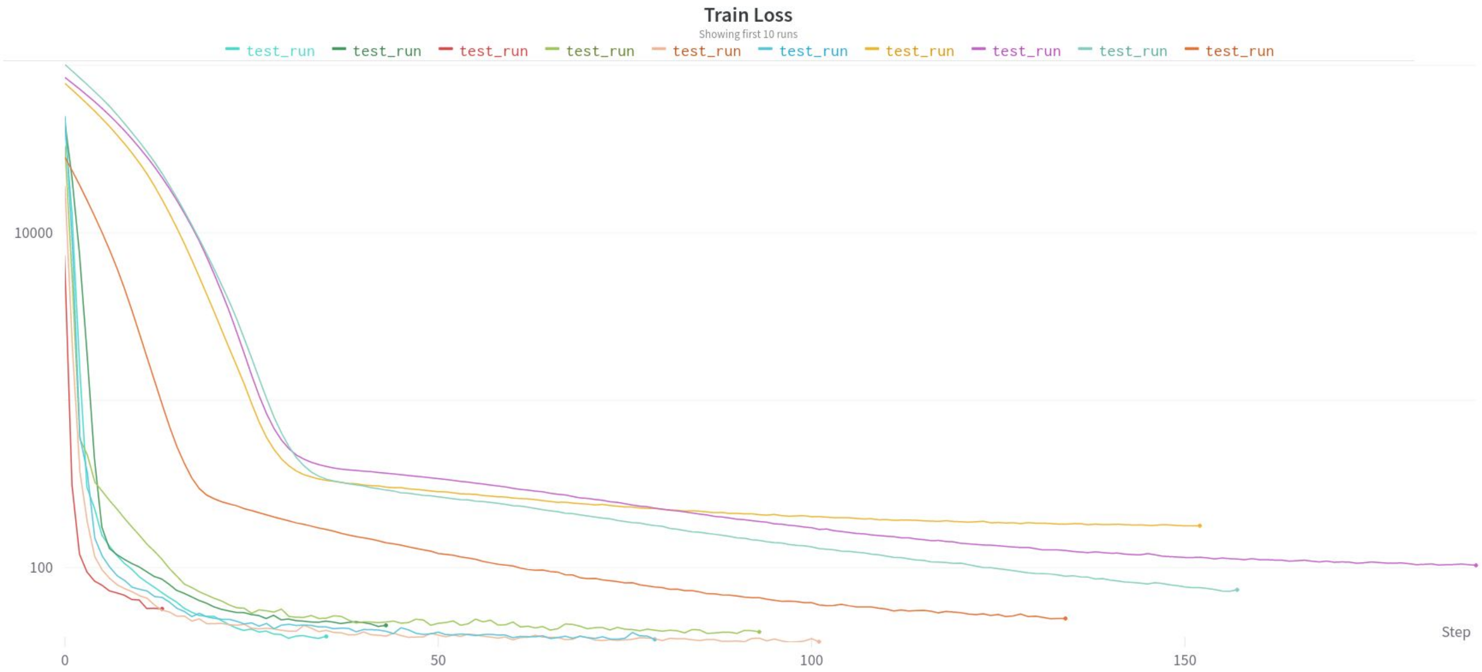
Input data	Output data	RMSE, bar
$p(t=1)$	$p(t=2,3,4)$	3.45
$p(t=1)$	$p(t=2,\ldots,20)$	20.19
$p(t=1)$	$p(t=7,14,20)$	15.71
$k, p(t=1)$	$p(t=7,14,20)$	15.17
$p(t=1,\ldots 10)$	$p(t=10,\ldots,20)$	1.31





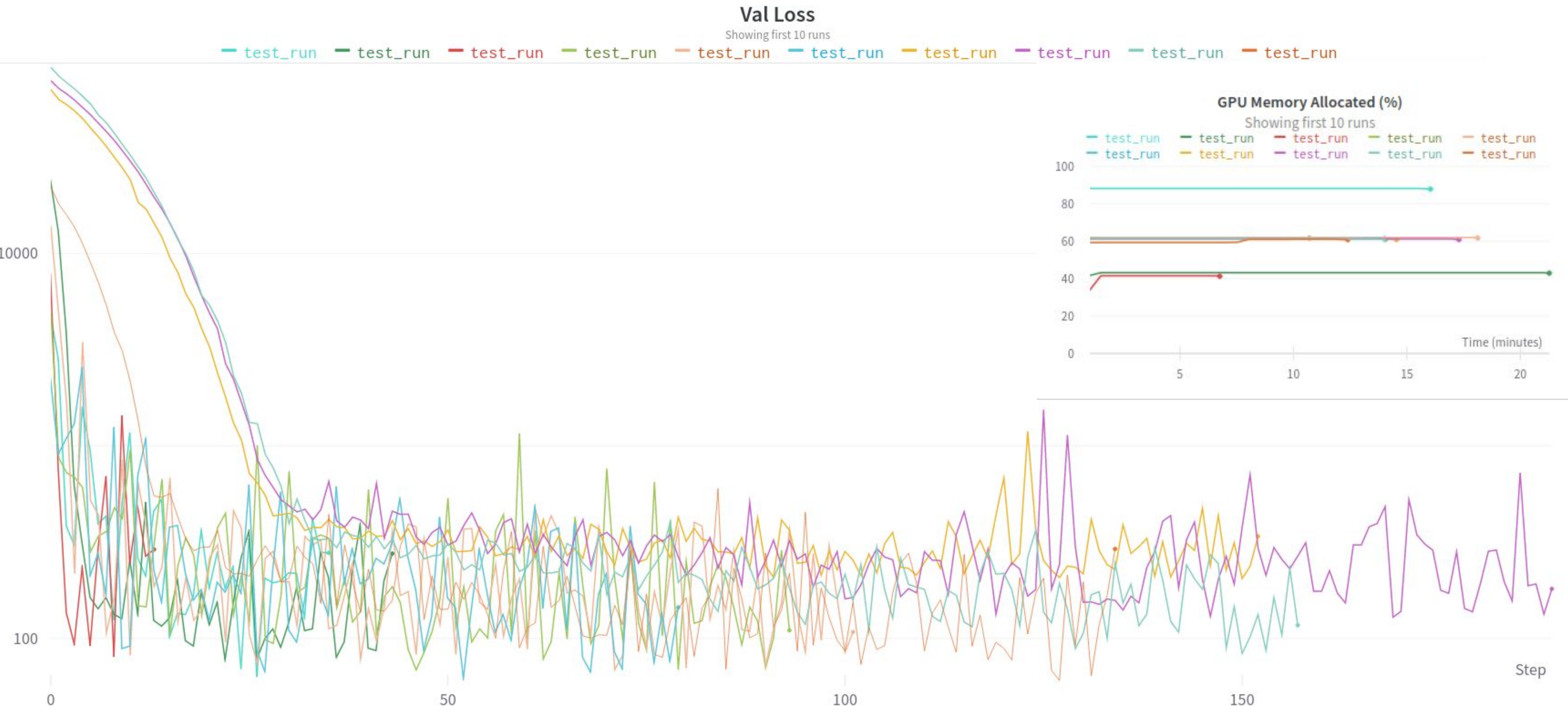
Custom FNO





Test RMSE: 6 ~ 9 bar.

Best architecture (red): 16 modes, depth = 4, width = 256, lr = 1e-3, w.d. = 1e-4


Val Loss
Showing first 10 runs
test_run test_run test_run test_run test_run test_run test_run test_run test_run test_run
10000
100
0
50
100
150
Step
GPU Memory Allocated (%)
Showing first 10 runs
test_run test_run test_run test_run test_run
test_run test_run test_run test_run test_run
100
80
60
40
20
0
Time (minutes)
5
10
15
20

Spectral neural operator (SNO)

Mapping input coefficients d_i (permeability) to the output coefficients b_i (pressure)

$$\sum_i g_i(x) d_i = f_{\text{in}}(x) \xrightarrow{\text{N}} f_{\text{out}}(x) = \sum_i g_i(x) b_i,$$

$g_i(x)$ is Chebyshev polynomial

Advantages:

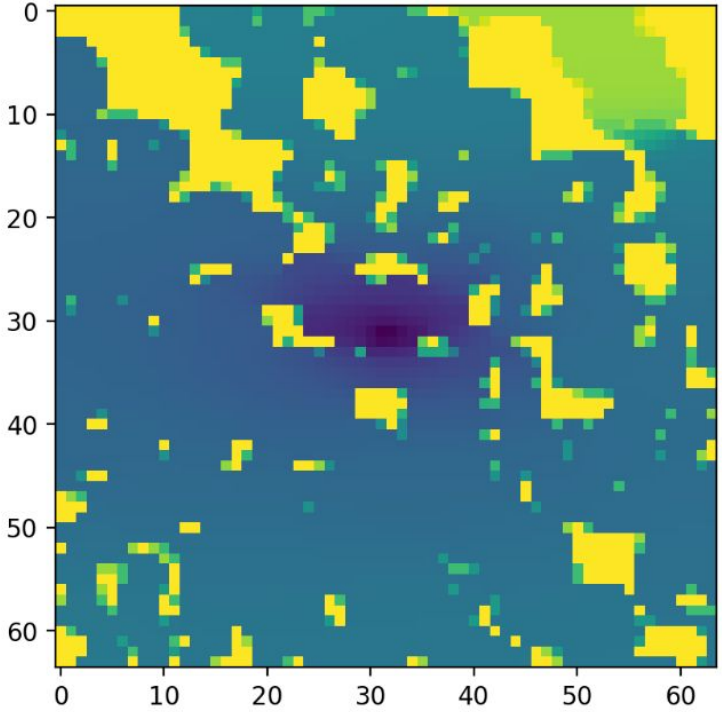
- Is not subject to aliasing errors
- Transparent output
- May include additional operations on functions

Disadvantages:

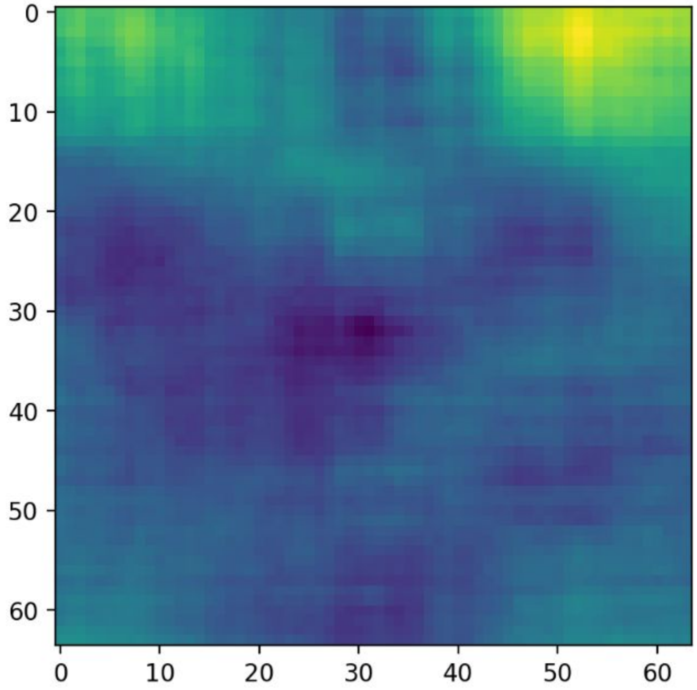
- Requires smooth function

Source: V. Fanaskov, I. Oseledets "Spectral Neural Operators"

Obtained result is poor by now

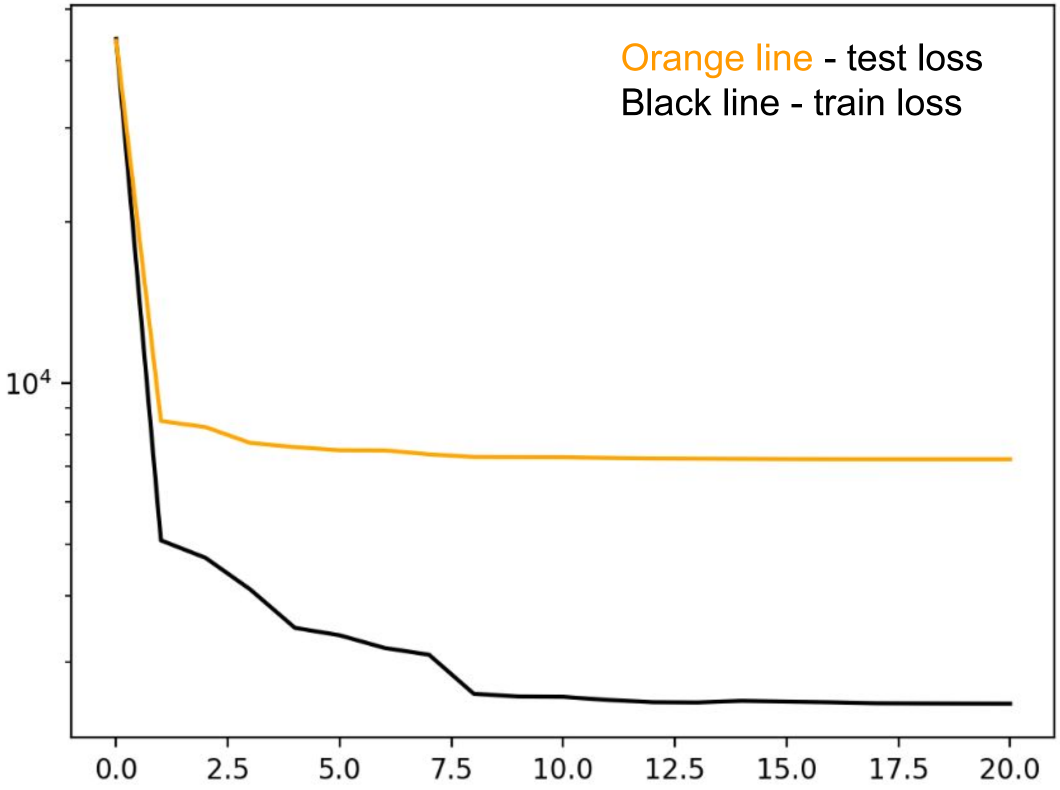


Calculated pressure map



Predicted pressure map

Loss function plot



Results are extremely poor either because of methodology mistakes or implementation mistake

Further research

- Complete SNO in Chebyshev basis implementation
- Implement other types of SNO
- Create a multi-well surrogate model
- Learn how respond the influence one production well on each other

Conclusions

1. PI-losses significantly increase the accuracy of PDEs and make the solution more physical.
2. FNO applicability for pressure distribution prediction was observed.
3. FNO give us pretty good results for prediction future steps results using time steps above.
4. Potentially SNO could be a great architecture to solve the problem.

References

1. Karniadakis, G.E., Kevrekidis, I.G., Lu, L. et al. “Physics-informed machine learning”. Nat Rev Phys 3, 422–440 (2021) - https://www.nature.com/articles/s42254-021-00314-5#citeas
2. Qiang Zheng, Lingzao Zeng, George Em Karniadakis. “Physics-informed semantic inpainting: Application to geostatistical modeling”. Journal of Computational Physics, Volume 419, 2020, 109676, ISSN 0021-9991 - https://www.sciencedirect.com/science/article/pii/S0021999120304502
3. Nanzhe Wang, Haibin Chang, Dongxiao Zhang. “Theory-guided Auto-Encoder for surrogate construction and inverse modeling”. Computer Methods in Applied Mechanics and Engineering, Volume 385, 2021, 114037, ISSN 0045-7825 - https://www.sciencedirect.com/science/article/pii/S0045782521003686
4. Suihong Song, Dongxiao Zhang, Tapan Mukerji, and Nanzhe Wang. “GANSim-surrogate: An Integrated Framework for conditional geomodelling and uncertainty analysis” - https://eartharxiv.org/repository/view/4594/
5. Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, Anima Anandkumar. “Fourier Neural Operator for Parametric Partial Differential Equations”. - https://arxiv.org/abs/2010.08895
6. Gege Wen, Zongyi Li, Kamyar Azizzadenesheli, Anima Anandkumar, Sally M. Benson “An enhanced Fourier neural operator-based deep-learning model for multiphase flow” - https://arxiv.org/abs/2109.03697

Questions?

Work distribution among team members

Daniil Sherki:
1. Unet approach architecture with data-loss
2. FNO implementation & experiments

Egor Cherepanov:
1. Custom FNO architecture implementation
2. PI - loss implementation & experiments

Damir Akhmetov:
1. Dataset generation
2. SNO approach implementation

Egor Malkershin:
1. SNO approach implementation
2. PI - loss implementation

GitHub link

https://github.com/PhysicsInformedWellSurrogareModel/PIWSM_dl_course