On Spectral Coefficient Learning Operator Network for NSEs

Choi, Junho at KAIST

Jan. 8, 2026

Spectral Coefficient Learning Operator Network (SpecONet) is a neural network

which learns (mimics) spectral coefficients of incompressible Navier-Stokes Equations (NSEs).

SpecONet's features

1. the sole novel method capable of quickly generating multiple inference solutions for 3D NSEs,

2. learning NSE structures without reference solutions for training,

3. superior accurate and robust with fewer nodal points,

4. flexible to various types of inputs.

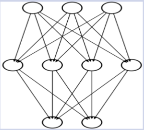
How it works

Input: PDE data

Initial condition : u_0
Boundary condition : g
Forcing fucntion: f

↓

Neural Network



Update parameters ←

Loss : weak forms

$$loss(\hat{u}) = \sum_{k=0}^{N-1} \left| \int_0^T \int_\Omega \hat{u}_t \phi_k + \mathcal{N}(\hat{u}, \nabla \hat{u}) \phi_k - f \phi_k dx\, dt \right|^2$$

↓

Output

Spectral coefficients : $\{\hat{\alpha}_k\}_{k=0}^{N-1}$

Basis functions
$\{\phi_k\}_{k=0}^{N-1} \subset \mathcal{H}$

→

Inference (Linear combination)

$\hat{u} = \sum_{k=0}^{N-1} \hat{\alpha}_k \phi_k$

↑

Incompressible Navier-Stokes equations

Here are the incompressible Navier-Stokes equations:

$$
\begin{aligned}
\mathbf{u}_t + \mathbf{u}\cdot\nabla\mathbf{u} - \nu\Delta\mathbf{u} + \nabla p &= \mathbf{f}, \quad \text{for} \quad t>0, x\in\Omega, \\
\nabla\cdot\mathbf{u} &= 0 \\
\mathbf{u} &= \mathbf{u}_0, \quad \text{for} \quad t=0, \quad x\in\Omega \\
\mathbf{u} &= \mathbf{g}, \quad \text{for} \quad t\geq 0, \quad x\in\partial\Omega
\end{aligned}
\tag{1.1}
$$

Spatial Scheme: Spectral element method

Let $\tilde{u} \approx \sum_{n=0}^{N-1} \alpha_n \psi_n$ and $\Phi \approx \sum_{n=0}^{N-1} \alpha_n \phi_n$ in

$$V_N := span\{\psi_0 \cdots \psi_{N-1}\} \subset H_0^1,$$
$$W_N := span\{\phi_0 \cdots \phi_{N-1}\} \subset H^1,$$

where $\psi_n = 0$ and $\frac{\partial \phi_n}{\partial \mathbf{n}} = 0$ on $\partial\Omega$.

Temporal Scheme: Rotational Pressure-correction Method

The temporal scheme breaks down NSEs into two kinds of Helmholtz equations:

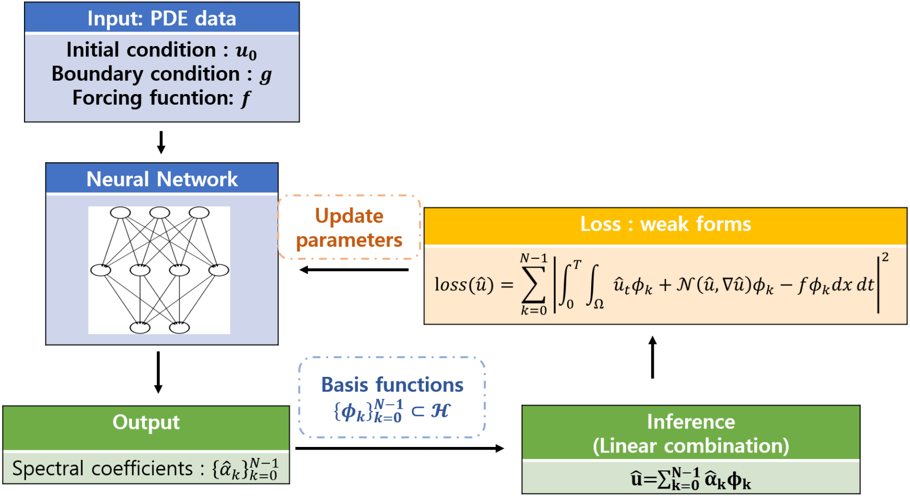
(1.2)
$$\frac{1}{2\Delta t}(3\widetilde{\mathbf{u}}^{k+1} - 4\mathbf{u}^k + \mathbf{u}^{k-1}) - \nu\Delta\widetilde{\mathbf{u}}^{k+1} + \nabla p^k = \mathbf{g}(t^{k+1}), \quad \widetilde{\mathbf{u}}^{k+1}|_{\partial\Omega} = 0$$

(1.3)
$$\Delta\Phi^{k+1} = \frac{3}{2\Delta t}\nabla \cdot \widetilde{\mathbf{u}}^{k+1}, \quad \left.\frac{\partial\Phi}{\partial\mathbf{n}}\right|_{\partial\Omega} = 0.$$

After solving the Helmholtz equations, $\mathbf{u}^{k+1}, p^{k+1}$ for the next time steps are obtained as

(1.4)
$$\mathbf{u}^{k+1} = \widetilde{\mathbf{u}}^{k+1} + \frac{2\Delta t}{3}\nabla\Phi^{k+1}$$

(1.5)
$$p^{k+1} = p^k + \Phi^{k+1} - \nu\nabla \cdot \widetilde{\mathbf{u}}^{k+1}.$$


Input: PDE data
Initial condition : u_0
Boundary condition : g
Forcing fucntion: f
Neural Network
Update parameters
Loss : weak forms
$loss(\hat{u}) = \sum_{k=0}^{N-1} \left| \int_0^T \int_\Omega \hat{u}_t \phi_k + \mathcal{N}(\hat{u}, \nabla\hat{u})\phi_k - f\phi_k dx\, dt \right|^2$
Output
Spectral coefficients : $\{\hat{\alpha}_k\}_{k=0}^{N-1}$
Basis functions
$\{\phi_k\}_{k=0}^{N-1} \subset \mathcal{H}$
Inference
(Linear combination)
$\hat{u} = \sum_{k=0}^{N-1} \hat{\alpha}_k \phi_k$

Definition of loss

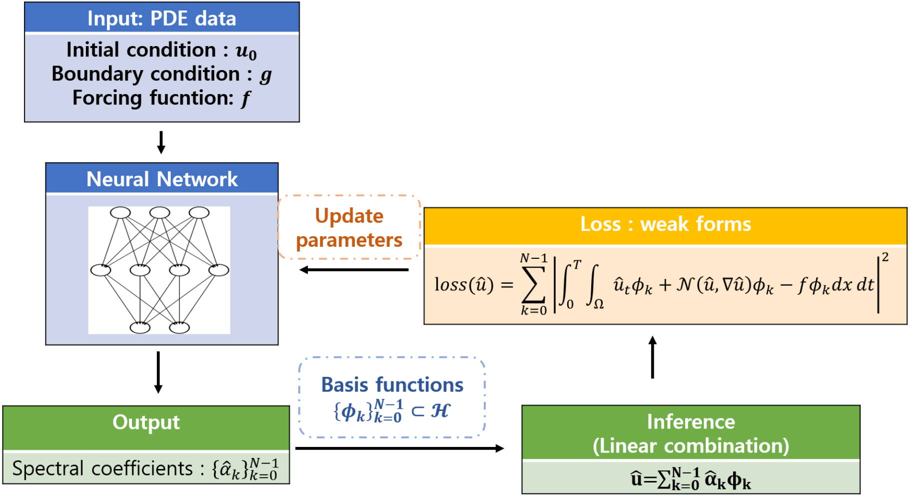
$$\mathcal{L}_{\widetilde{u}}^{(k+1)} = \sum_{l,m,n=0}^{N-1} \left| \int_\Omega \frac{1}{2\Delta t}(3\widehat{\widetilde{\mathbf{u}}}_N^{k+1} - 4\widehat{\mathbf{u}}^k + \widehat{\mathbf{u}}^{k-1})\overline{\Psi}_{lmn} d\mathbf{x} d\mathbf{x} \right.$$

$$\left. + \nu \int_\Omega \nabla \widehat{\widetilde{\mathbf{u}}}_N^{k+1} \cdot \nabla \overline{\Psi}_{lmn} + \int_\Omega \nabla \widehat{p}^k \overline{\Psi}_{lmn} d\mathbf{x} - \int_\Omega \mathbf{g}^{k+1} \overline{\Psi}_{lmn} d\mathbf{x} \right|^2, \tag{1.6}$$

for $\overline{\Psi}_{lmn} \in V_N$.

$$\mathcal{L}_{\Phi}^{(k+1)} = \sum_{l,m,n=0}^{N-1} \left| \int_\Omega \nabla \widehat{\Phi}_N^k \cdot \nabla \overline{\phi}_{lmn} + \frac{3}{2\Delta t} \nabla \cdot \widehat{\widetilde{\mathbf{u}}}_N^k \overline{\phi}_{lmn} d\mathbf{x} \right|^2. \tag{1.7}$$

for $\overline{\phi}_{lmn} \in W_N$.


Input: PDE data
Initial condition : u_0
Boundary condition : g
Forcing fucntion: f
Neural Network
Update parameters
Loss : weak forms
$loss(\hat{u}) = \sum_{k=0}^{N-1} \left| \int_0^T \int_\Omega \hat{u}_t \phi_k + \mathcal{N}(\hat{u}, \nabla \hat{u}) \phi_k - f \phi_k dx\, dt \right|^2$
Output
Spectral coefficients : $\{\hat{\alpha}_k\}_{k=0}^{N-1}$
Basis functions
$\{\phi_k\}_{k=0}^{N-1} \subset \mathcal{H}$
Inference
(Linear combination)
$\hat{u} = \sum_{k=0}^{N-1} \hat{\alpha}_k \phi_k$

After training networks, $\widehat{\mathbf{u}}^{k+1}, \widehat{p}^{k+1}$ for the next time steps are obtained as

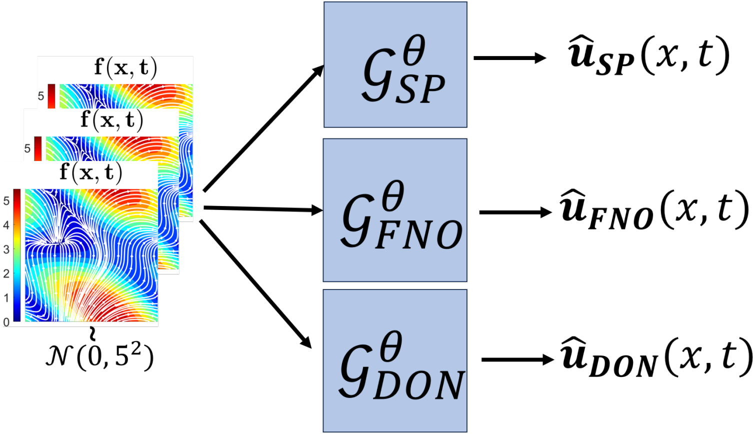
$$\widehat{\mathbf{u}}^{k+1} = \widehat{\widetilde{\mathbf{u}}}^{k+1} + \frac{2\Delta t}{3}\nabla\widehat{\Phi}^{k+1} \tag{1.8}$$

$$\widehat{p}^{k+1} = \widehat{p}^{k} + \widehat{\Phi}^{k+1} - \nu\nabla\cdot\widehat{\widetilde{\mathbf{u}}}^{k+1}. \tag{1.9}$$

Numerical experiments

1. Flexible modality
2. Robustness for perturbed inputs and high-variance-distribution inputs
3. Inference time

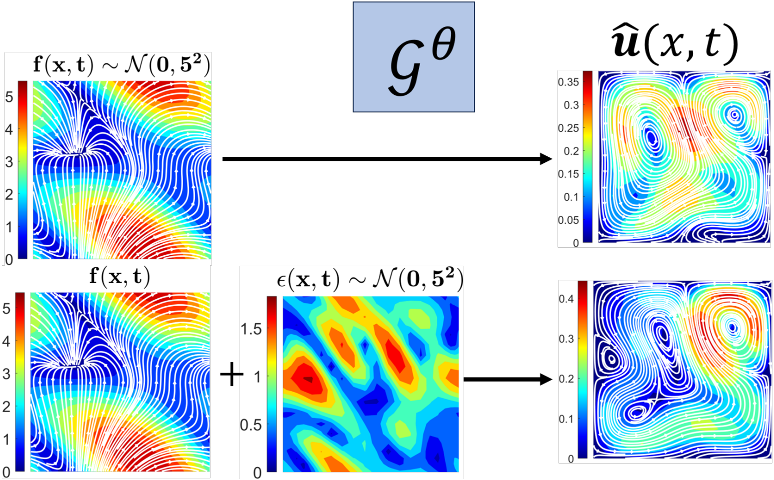
Robustness 1. perturbed inputs



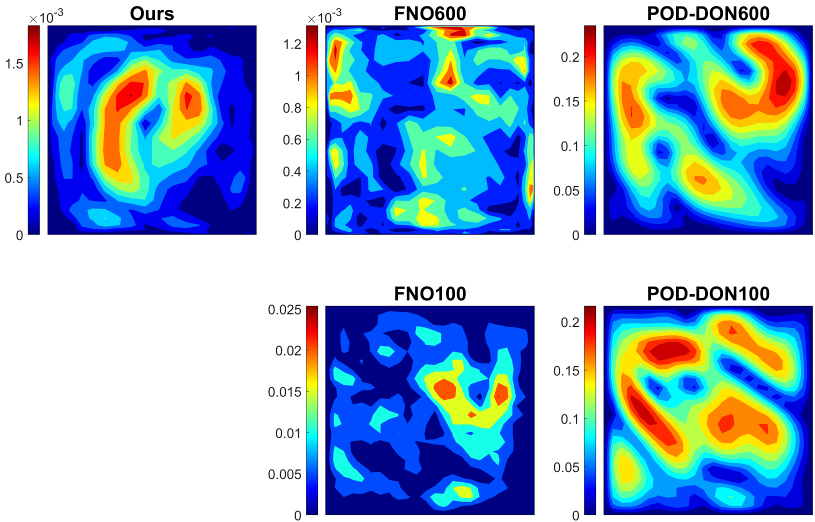
[1] Li, Zongyi, et al. *Fourier neural operator for parametric partial differential equations.* arXiv preprint arXiv:2010.08895 (2020).

[2] Lu, Lu, et al. *A comprehensive and fair comparison of two neural operators (with practical extensions) based on fair data*, Computer Methods in Applied Mechanics and Engineering 393 (2022): 114778.

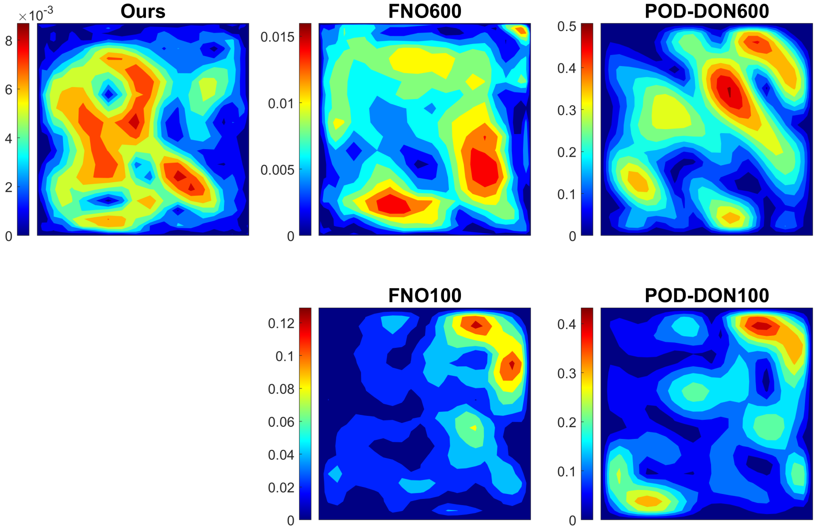
Test networks with perturbed inputs



Given $\mathbf{f} \sim \mathcal{N}(\mathbf{0}, \mathbf{5}^2)$, $|\mathbf{u} - \widehat{\mathbf{u}}|$ at $t = 1$:

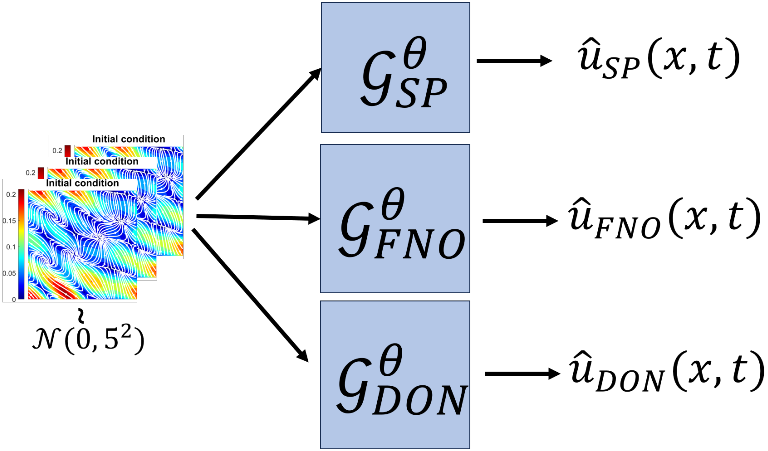


Given $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{5}^2)$, $|\mathbf{u} - \widehat{\mathbf{u}}|$ at $t = 1$:

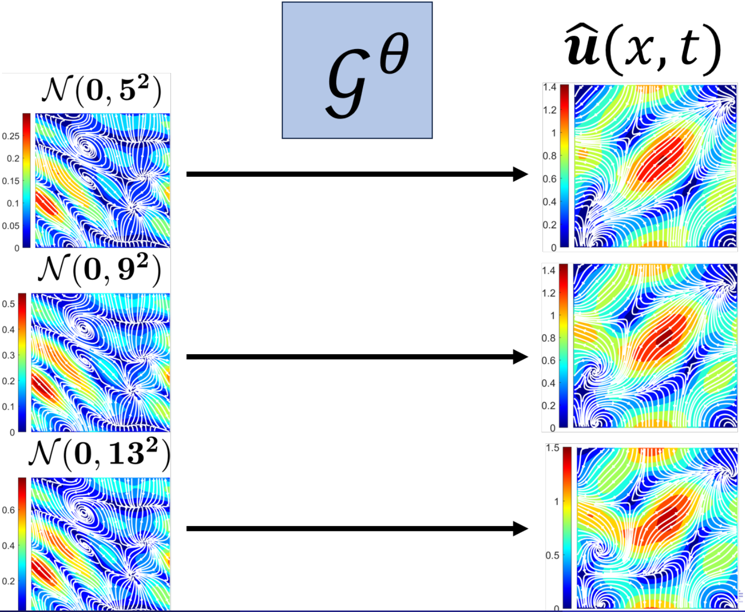


Robustness 2. higher-variance-distribution inputs

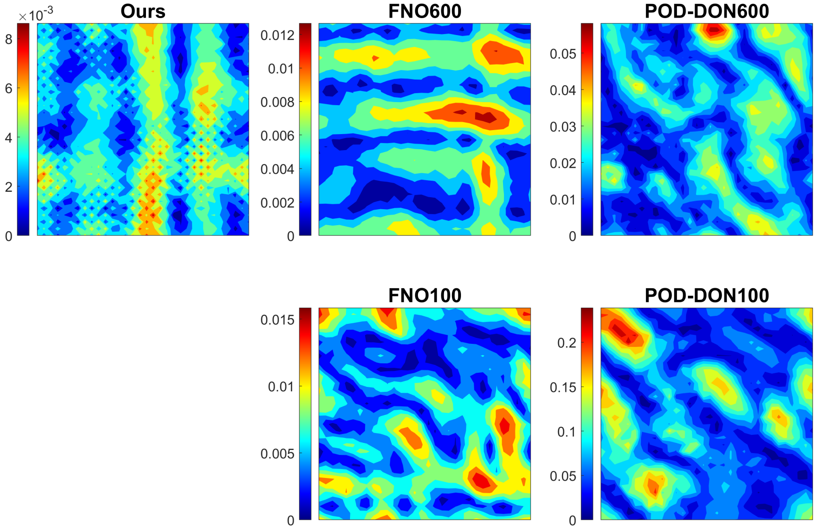
Train SpecONet, FNO, POD-DON with input datasets of initial conditions.



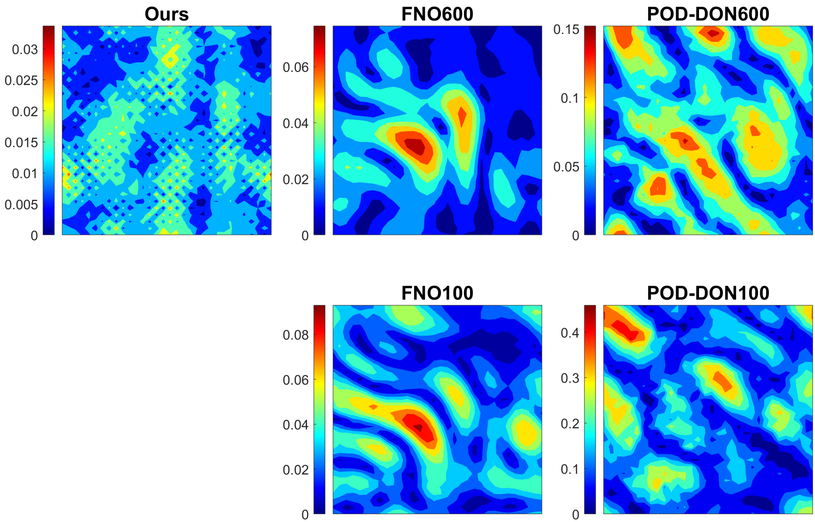
Test networks with inputs drawn from various distributions



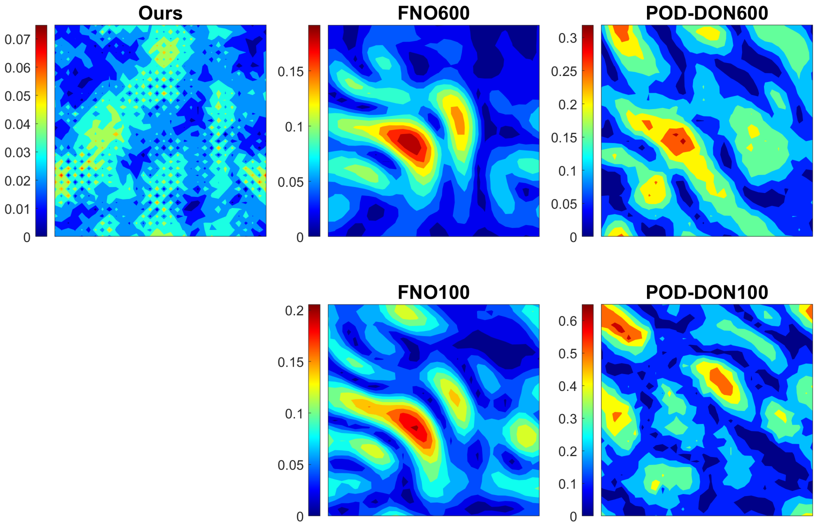
Given $\mathbf{u_0} \sim \mathcal{N}(\mathbf{0}, \mathbf{5^2})$, $|\mathbf{u} - \widehat{\mathbf{u}}|$ at $t = 1$:



Given $\mathbf{u_0} \sim \mathcal{N}(\mathbf{0}, \mathbf{9^2})$, $|\mathbf{u} - \widehat{\mathbf{u}}|$ at $t = 1$:

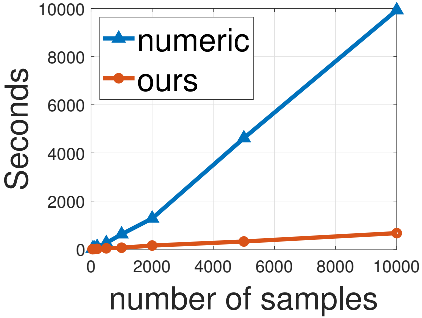


Given $\mathbf{u_0} \sim \mathcal{N}(\mathbf{0}, \mathbf{13^2})$, $|\mathbf{u} - \widehat{\mathbf{u}}|$ at $t = 1$:



Quick inference

3D NSEs with forcing functions as input data:



number of samples	50	200	500	2000	5000	10000
Inference(sec)	3.3	15.9	37.6	152.6	321.2	668.6
Computing(sec)	23.4	101.6	259.4	1279.4	4613.7	9927.9
ratio	7.1	6.4	6.9	10.0	14.4	14.8

Thank you.

junho_choi@kaist.ac.kr

Choi, J., Chang, T. Y., Kim, N., and Hong, Y. (2025).
A data free neural operator enabling fast inference of 2D and 3D Navier Stokes equations.
arXiv preprint arXiv:2510.23936.