Separable DeepONet: A Scalable Framework for High-Dimensional Physics-Informed Neural Operators

Somdatta Goswami

Assistant Professor, Civil and Systems Engineering

November 2, 2024

2024 SIAM New York-New Jersey-Pennsylvania Section Conference at Rochester Institute of Technology

Physics-based Models

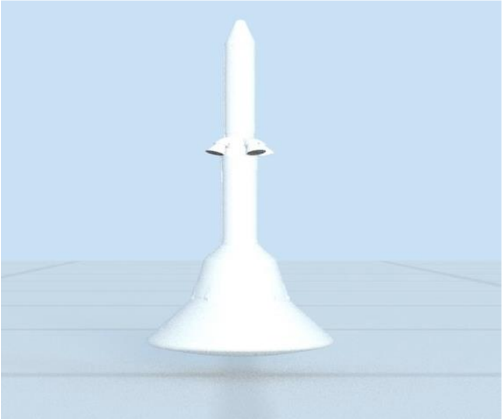
Can represent the Processes of Nature

- Physics-based models are approximated via **ODEs/PDEs**

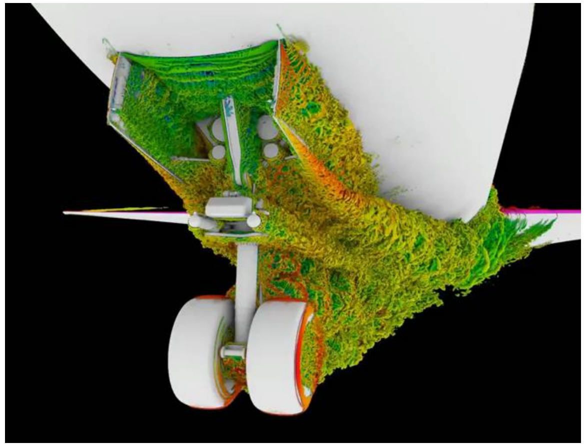
  To model earthquake: $m\dfrac{d^2u}{dt^2} + \mathrm{k}\dfrac{du}{dt} + F_0 = 0$

  To model waves: $\dfrac{\partial^2 u}{\partial t^2} - v^2\left(\dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2}\right) = 0$

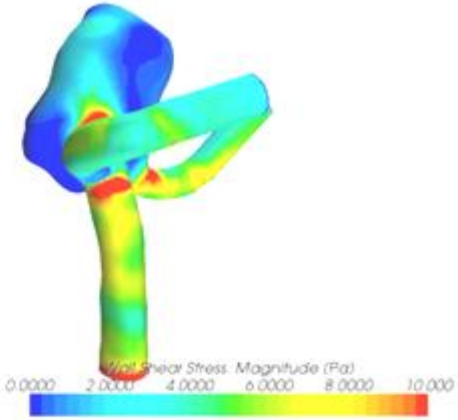
- Computational Mechanics helps us simulate these equations.



Simulation of Orion Spacecraft Launch Abort System (NASA Ames)



Detailed flow around an Aircraft's landing gear (NASA Ames)



CFD Simulation of a Patient-Specific Intracranial Aneurysm

JOHNS HOPKINS
WHITING SCHOOL of ENGINEERING

Challenges with Numerical Methods

- Require knowledge of conservation laws, and boundary conditions
- Time consuming and strenuous simulations.
- Difficulties in mesh generation.
- Solving inverse problems or discovering missing physics can be prohibitively expensive.

Develop Physics-based surrogate models for these systems to create a fast-to-evaluate alternative.

Surrogate Modeling Techniques

- Discretized Data
- Discretization dependent
- Queries on mesh
- Learning functions between vector spaces

Data-driven

- Functional Data
- Discretization Invariant
- Continuous quantities
- Learning operators between function spaces

PCA **Auto-encoders**

K-PCA **Diffusion maps**

f-PCA **DeepONet** **LNO**

FNO **WNO**

F-RKHS

Finite Dimensional

Infinite Dimensional

Physics Informed

PINNs

PI-DeepONet

PINO

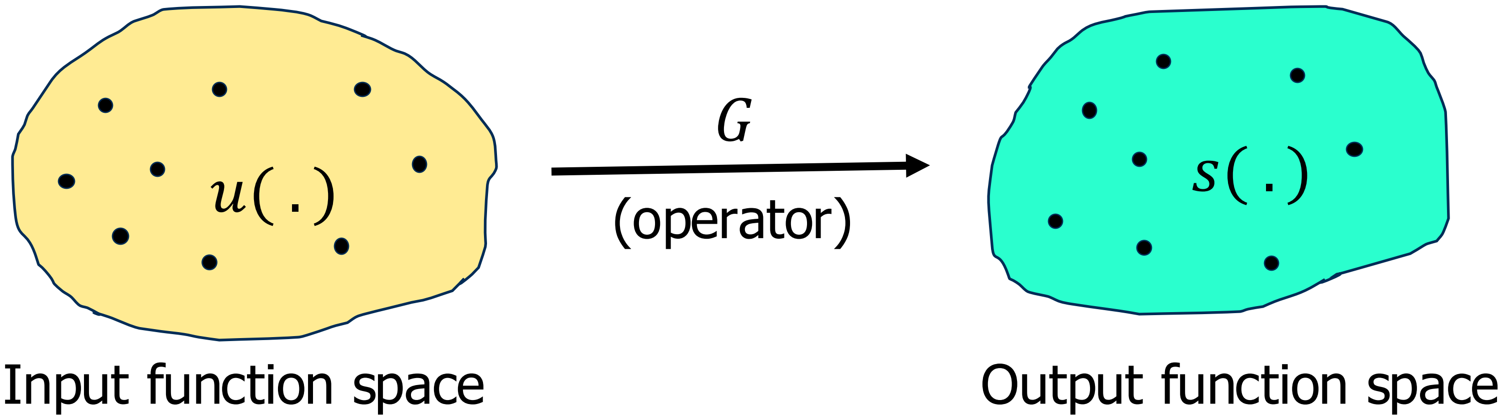
Operator Learning Framework

Input-output map

$$\Phi: \mathcal{U} \to \mathcal{S}$$

Data $\{\mathcal{U}_n, \mathcal{S}_n\}_{n=1}^{N}$and/or Physics

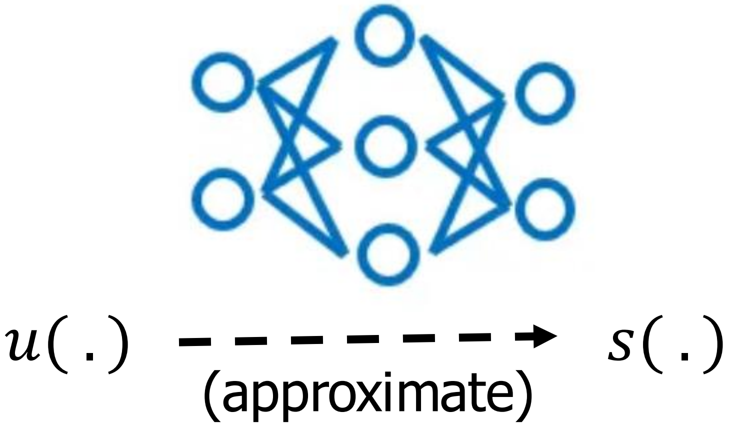
$\mathcal{S}_n = \Phi(\mathcal{F}_n)$, $\mathcal{F}_n \sim \mu \; i.i.d$



Operator learning

$$\Psi: \times\, \Theta \to \mathcal{S} \text{ such that } \Psi(.,\theta^*) \approx \Phi$$

Training $\theta^* = \mathrm{argmin}_\theta\, l(\{\mathcal{U}_n, \Psi(\mathcal{S}_n, \theta)\})$

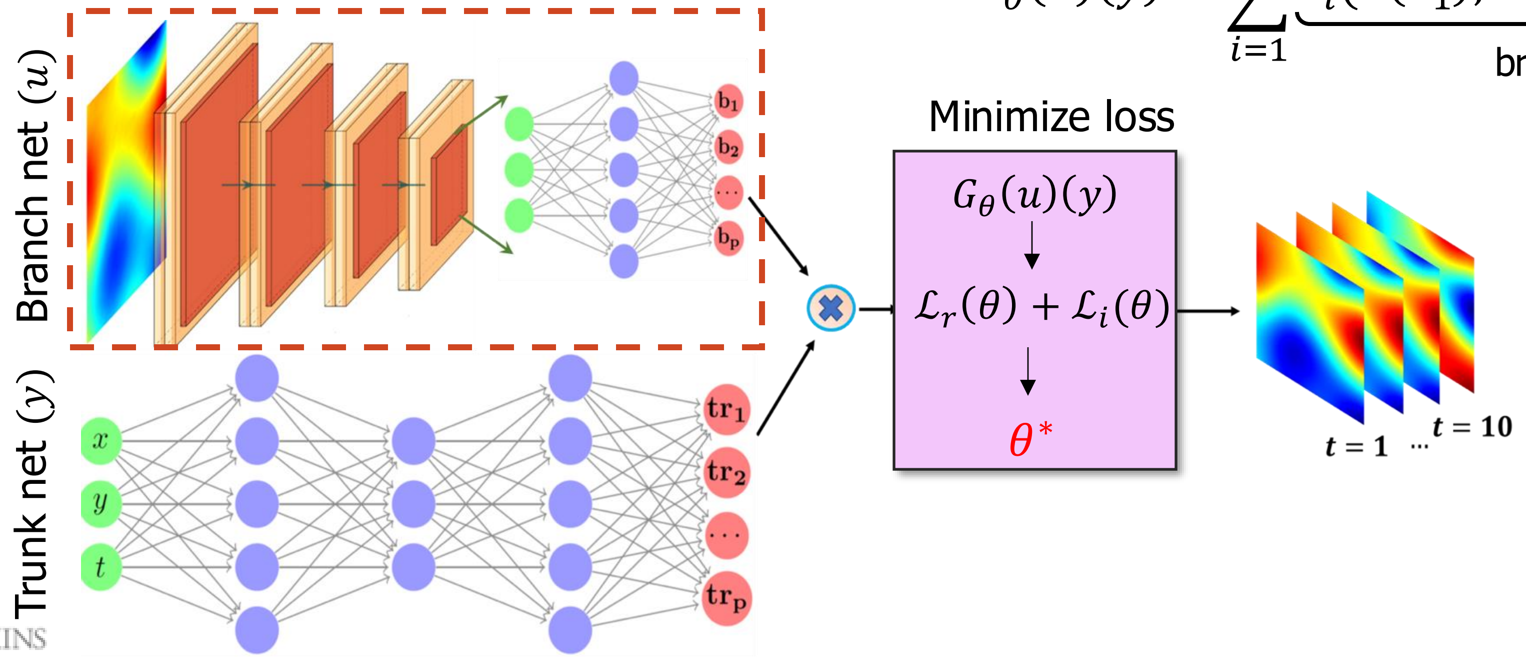


Deep Operator Network (DeepONet)

- Generalized Universal Approximation Theorem for Operator [Chen '95, Lu et al. '19]
- **Branch net**: Input $\{u(x_i)\}_{i=1}^{m}$, output: $\left[b_1, b_2, .., b_p\right]^T \in \mathbb{R}^p$
- **Trunk net**: Input y, output: $\left[t_1, t_2, .., t_p\right]^T \in \mathbb{R}^p$
- Input u is evaluated at the fixed locations $\{y_i\}_{i=1}^{m}$

$$G_\theta(u)(y) = \sum_{i=1}^{p} \underbrace{b_i(u(x_1), u\,(x_2), \ldots, u\,(x_m))}_{\text{branch net}} \cdot \underbrace{tr_i(y)}_{\text{trunk net}}$$

Can be FNN/CNN/U-Net/LSTM etc.



Data-Driven Training of DeepONet

$\nabla\big(K(\boldsymbol{x})\nabla u(\boldsymbol{x})\big) = 1 \; u(\boldsymbol{x}) = 0 \; \forall \;\; \boldsymbol{x} \in \partial\Omega$

Nonlinear operator $\mathcal{G} : \mathcal{K} \to \mathcal{U}$

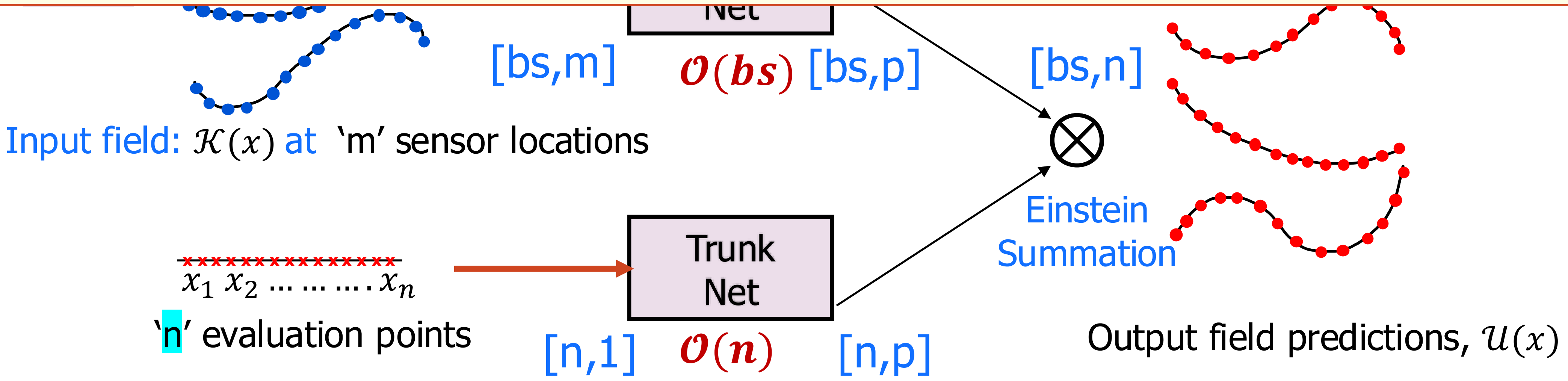
Neural operator $\mathcal{G}_\theta : \mathcal{K} \to \mathcal{U}, \;\; \theta \in \Theta$

Training data $\{\mathbf{x}_i, \boldsymbol{y}_i\}_{i=1}^{N}$

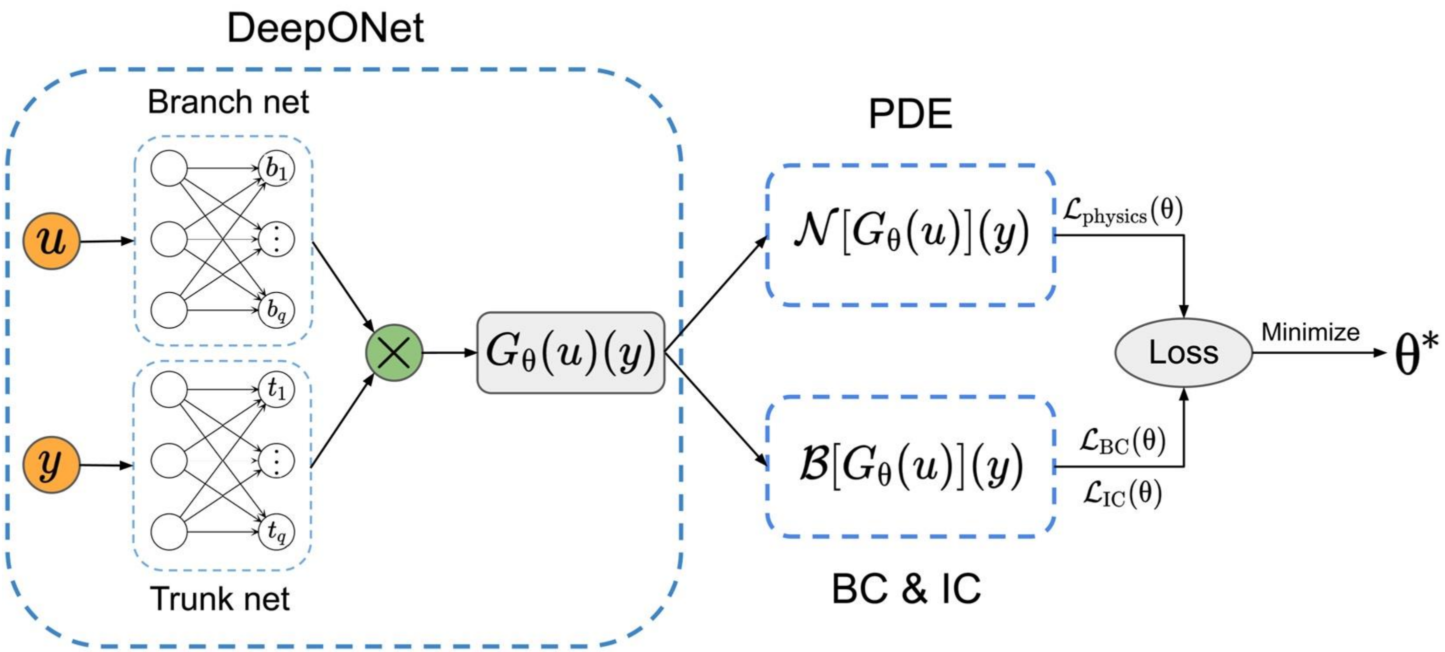
Training Dataset

S.No	Input field data	Output field data
1	$k^1(x_1), \mathrm{k}^1(x_2), \cdots\cdots\cdots, k^1(x_m)$	$\mathrm{u}^1(x_1), \mathrm{u}^1(x_2), \cdots\cdots\cdots, \mathrm{u}^1(x_n)$
2	$k^2(x_1), \mathrm{k}^2(x_2), \cdots\cdots\cdots, \mathrm{k}^2(x_m)$	$u^2(x_1), \mathrm{u}^2(x_2), \cdots\cdots\cdots, \mathrm{u}^2(x_n)$
.	.	.
N	$k^{\mathrm{N}}(x_1), \mathrm{k}^{\mathrm{N}}(x_2), \cdots\cdots\cdots, \mathrm{k}^{\mathrm{N}}(x_m)$	$u^N(x_1), u^N(x_2), \cdots\cdots\cdots, \mathrm{u}^{\mathrm{N}}(x_{\mathrm{n}})$

Extremely data-hungry.

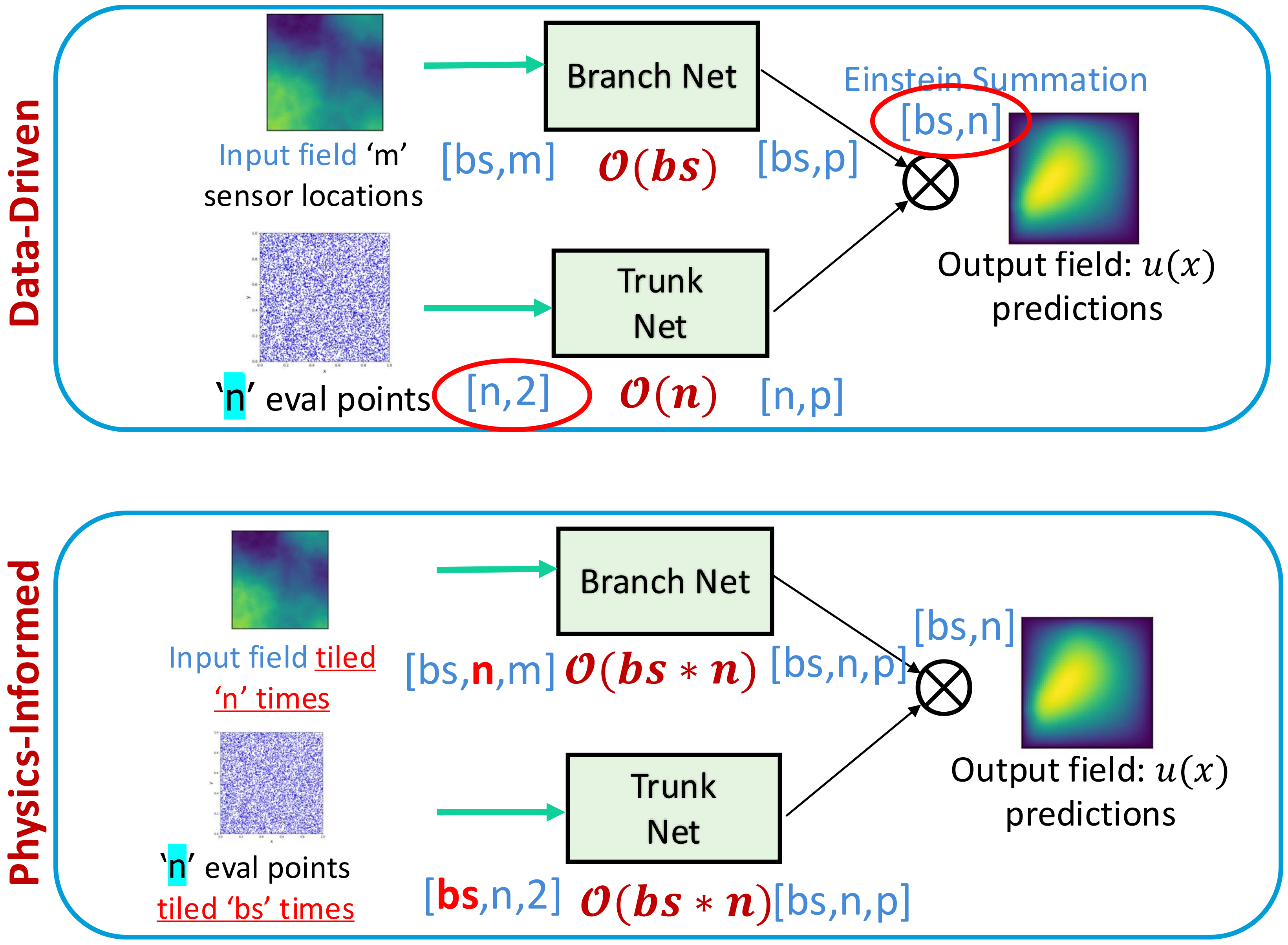


Physics-Informed DeepONet



- Wang et.al “Learning the solution operator of parametric partial differential equations with physics-informed DeepONets” Science Advances, 2021
- Goswami et al. "A physics-informed variational DeepONet for brittle fracture." *CMAME, 2022*.

Frameworks for $\nabla(K(\boldsymbol{x})\nabla u(\boldsymbol{x})) = 1\ u(\boldsymbol{x}) = 0\ \forall\ \boldsymbol{x} \in \partial\Omega$ and $\boldsymbol{x} = (x, y)$



Derivatives

$$\frac{\partial \mathcal{L}}{\partial \theta}$$

$$\frac{\partial \mathcal{L}}{\partial \theta}, \frac{\partial u}{\partial x}, \frac{\partial^2 u}{\partial x^2}, \cdots$$

Reverse-mode autodiff

$$J = [bs * n, bs * n]$$

Shortcomings

↑

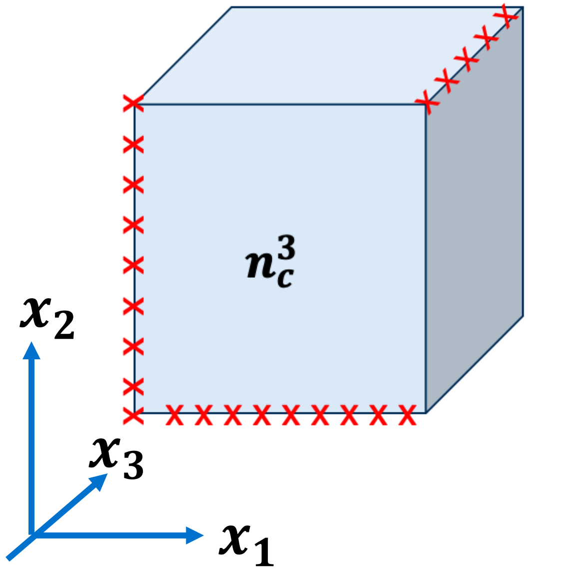
Training is extremely expensive. So, never made it to common practice.

Our Proposed framework

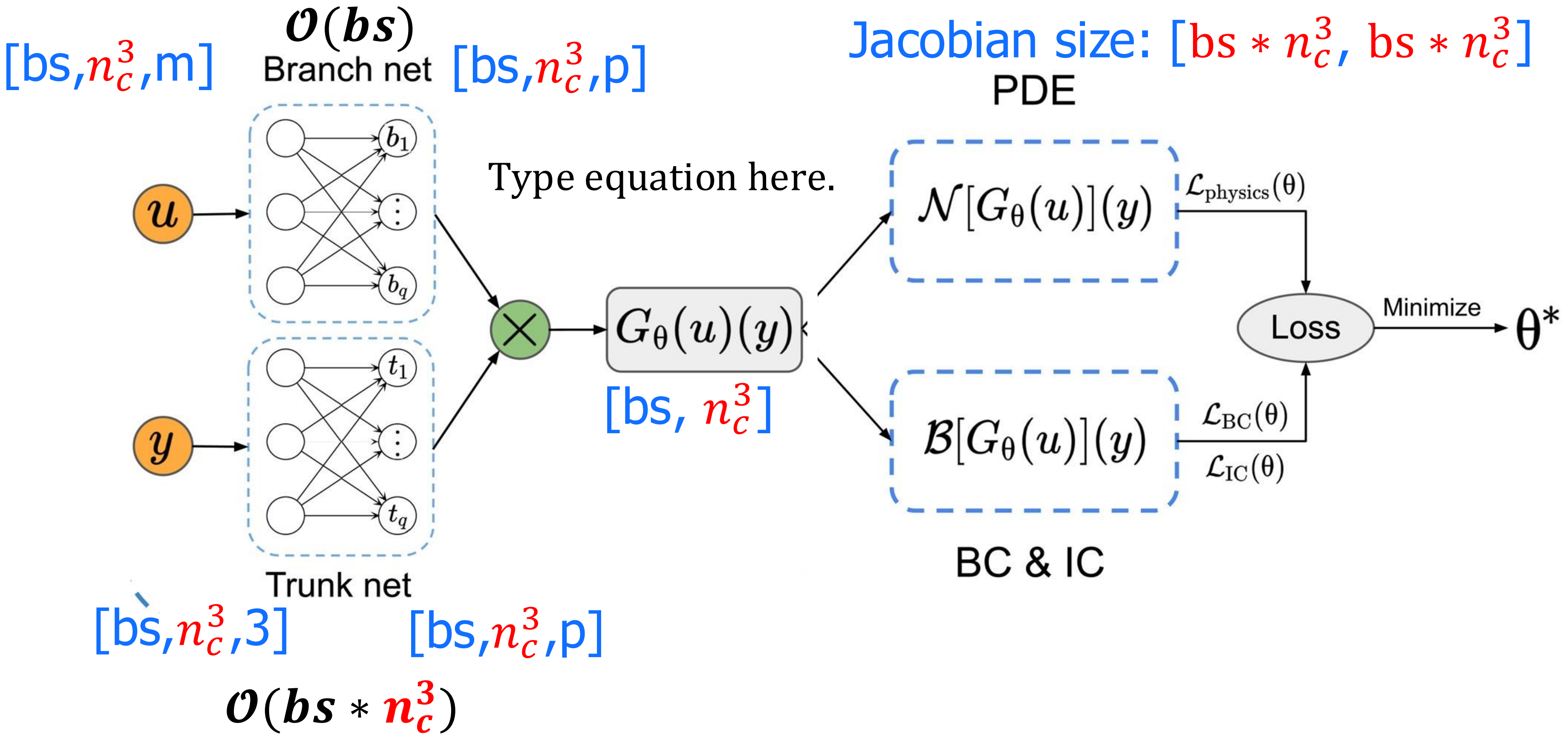
Separable DeepONet: Breaking the Curse of Dimensionality in Physics-Informed Machine Learning

Luis Mandl, Somdatta Goswami, Lena Lambers, and Tim Ricken. "Separable DeepONet: Breaking the Curse of Dimensionality in Physics-Informed Machine Learning." *arXiv preprint arXiv:2407.15887* (2024).

Vanilla – Physics Informed DeepONet



$n_{x_1} = n_{x_2} = n_{x_3} = n_c$

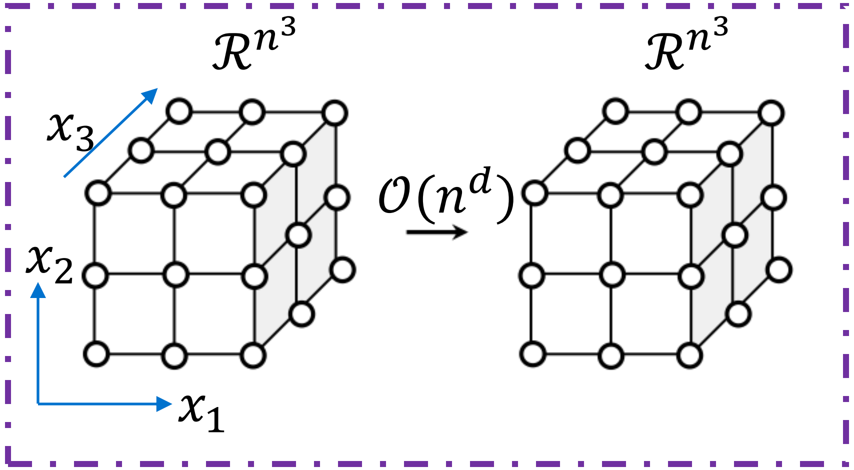


Sifan Wang, Hanwen Wang, and Paris Perdikaris. Learning the solution operator of parametric partial differential equations with physics-informed DeepONets. Science Advances, 7(40), October 2021.

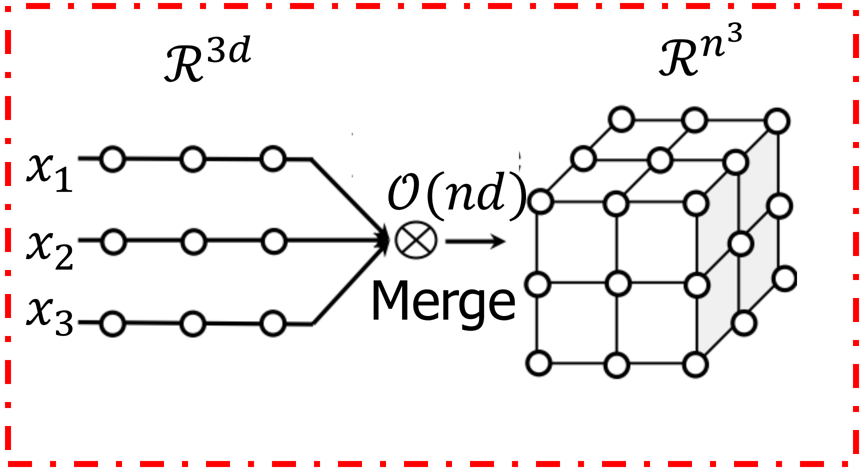
JOHNS HOPKINS WHITING SCHOOL of ENGINEERING

Introducing Separation of Variables

Vanilla Trunk network

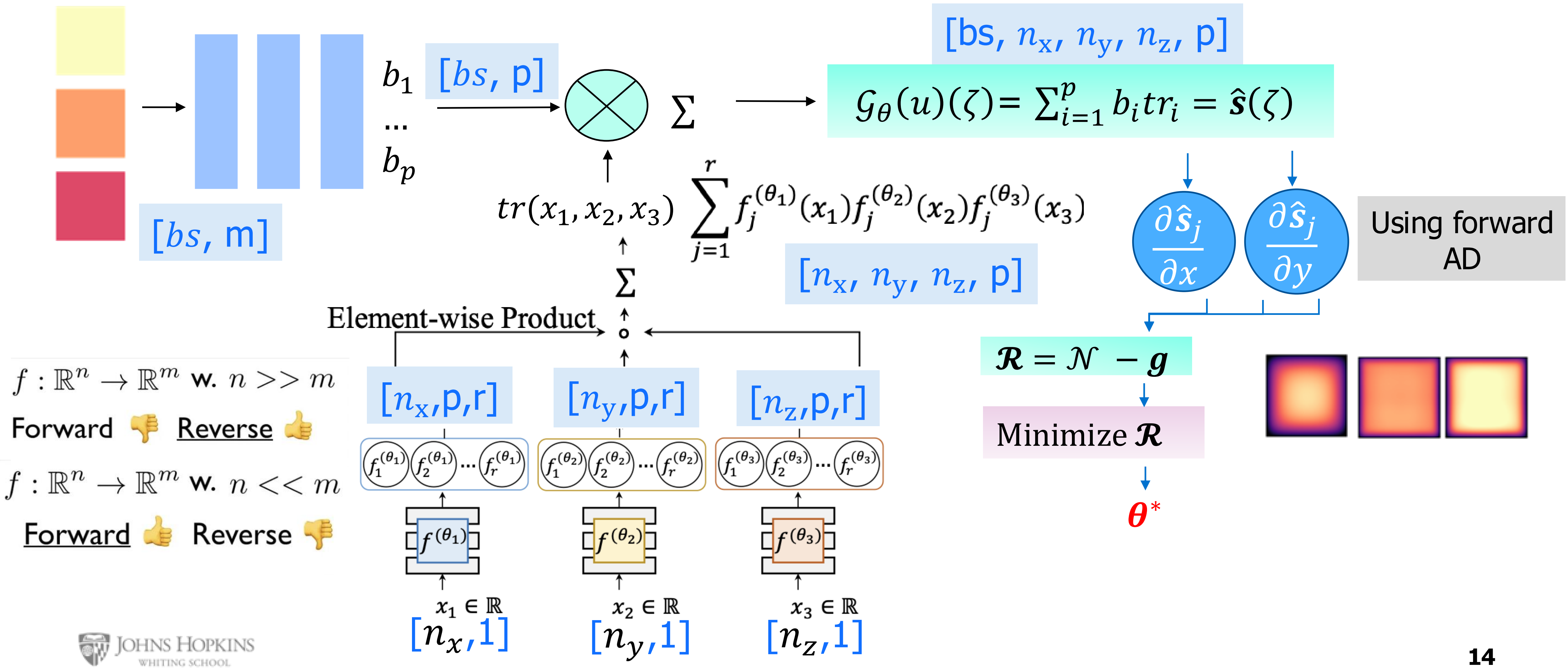


Separated Trunk network



Introduced in PINNs : Cho, J., Nam, S., Yang, H., Yun, S. B., Hong, Y., & Park, E. (2022). Separable PINN: Mitigating the curse of dimensionality in physics-informed neural networks. ArXiv preprint - 2211.08761.

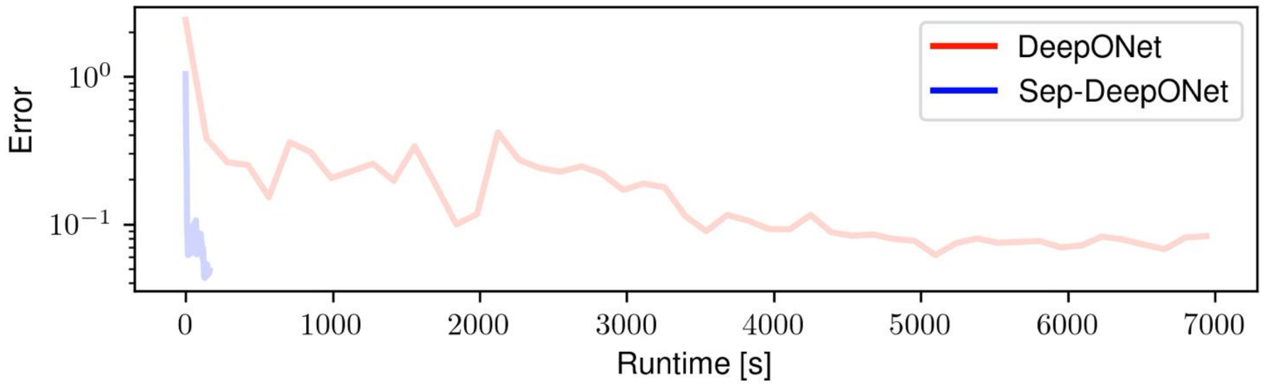
Separable DeepONet Framework



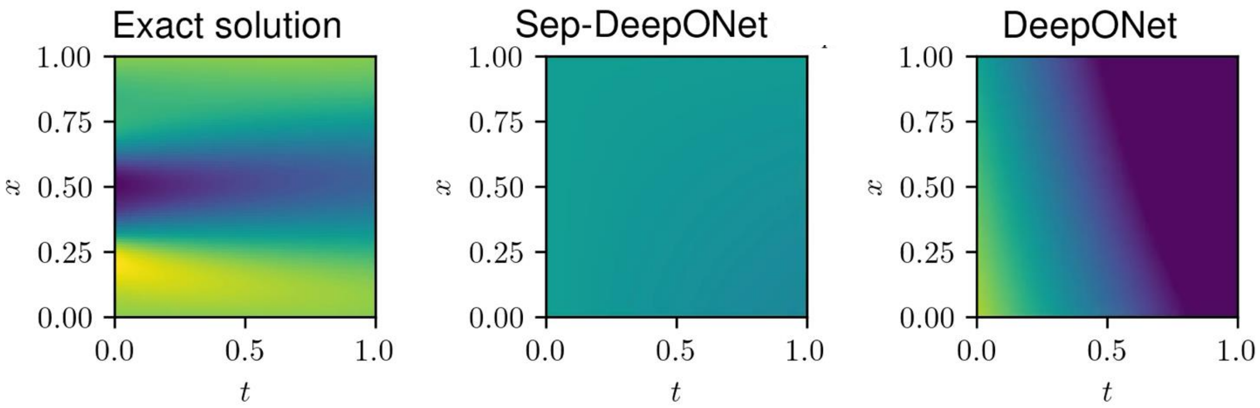
Numerical Examples

Problem	Model	d	Relative $\mathcal{L}_2$ error	Run-time (ms/iter.)
Burgers Equation	Vanilla	2	5.1e-2	136.6
	Separable (Ours)		6.2e-2	3.64
Consolidation Biot's Theory	Vanilla	2	7.7e-2	169.43
	Separable (Ours)		7.9e-2	3.68
Parameterized Heat Equation	Vanilla	4	-	10,416.7
	Separable (Ours)		7.7e-2	91.73

Burgers' Equation



$$\frac{\partial s(x,t)}{\partial t} + s\frac{\partial s(x,t)}{\partial x} - \nu\frac{\partial^2 s(x,t)}{\partial x^2} = 0,$$
$$s(0,t) = s(1,t),$$
$$\frac{\partial s(0,t)}{\partial x} = \frac{\partial s(1,t)}{\partial x},$$
$$s(x,0) = u(x), \quad x \in [0,1]$$



Model	Branch	Trunk	p	r	Parameters	$\mathcal{L}_2$ rel. err.	Runtime [s]	Runtime improvment
Vanilla PI-DeepONet	$6\times[100]$	$6\times[100]$	100	-	131,701	5.14e-2	6,829.2	-
Sep-PI-DeepONet	$6\times[100]$	$6\times[100]$	50	50	672,151	6.24e-2	182.1	97,33%
	$6\times[100]$	$6\times[100]$	20	20	244,921	6.04e-2	197.8	97,10%
	$6\times[100]$	$6\times[50]$	20	20	129,221	6.46e-2	197.0	97,12%

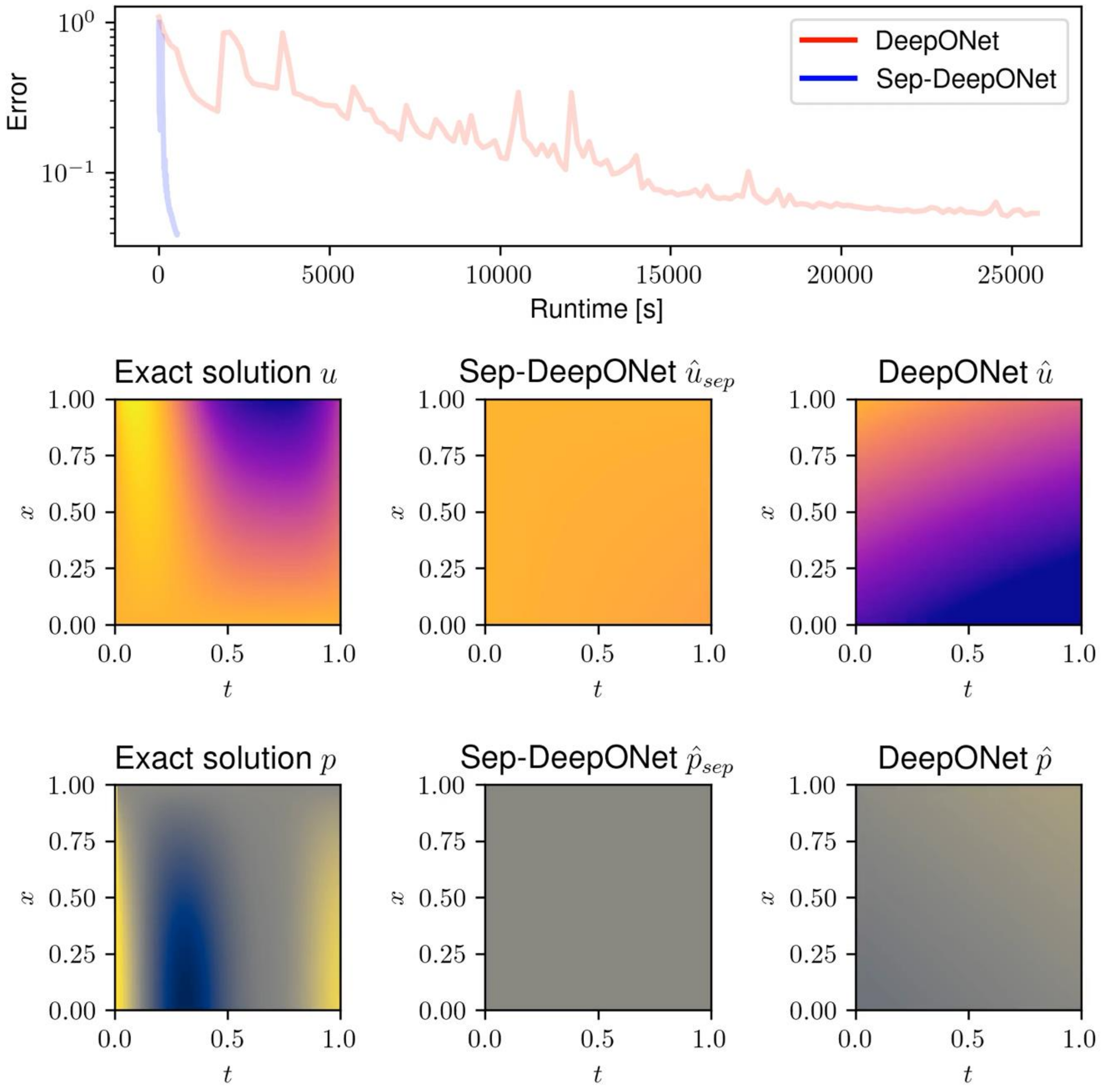
JOHNS HOPKINS
WHITING SCHOOL of ENGINEERING

Biot’s Consolidation

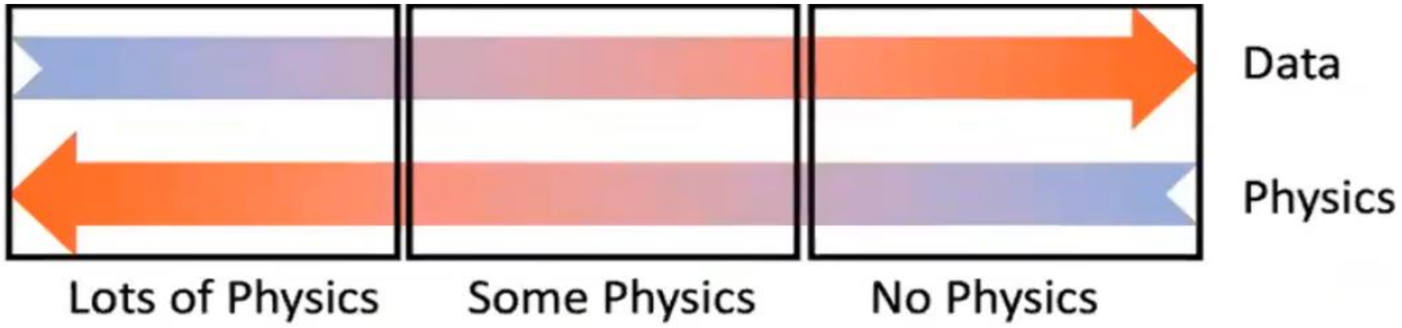
$$(\lambda + 2\mu)\frac{\partial^2 u(z,t)}{\partial z^2} - \frac{\partial p(z,t)}{\partial z} = 0$$

$$\frac{\partial^2 u(z,t)}{\partial t \partial z} - \frac{k}{\rho g}\frac{\partial^2 \tilde{p}(z,t)}{\partial z^2} = 0,$$

$$u(z,0) = 0, \qquad p(0,t) = 0,$$
$$p(z,0) = f(0), \qquad u(L,t) = 0,$$
$$\sigma(0,t) = -f(t), \qquad \frac{\partial p(L,t)}{\partial z} = 0,$$



Key Takeaways



- These methods have a niche in real world problems, where partially physics in known and some measurements of quantities of interest are available.
- Separable architecture introduces the possibility of employing physics in neural operators.
- Extending this framework for non-separable differential equation would be a part of our future work.

Acknowledgement



From U. of Stuttgart, Germany

Funding

For our other project, please visit the group

Thank you!