NEURAL OPERATORS FOR SCIENTIFIC MACHINE LEARNING

Presented by Handi Zhang

Functions map data, Operators map functions

- Function: $\mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$

  e.g., image classification:  $\mapsto 5$

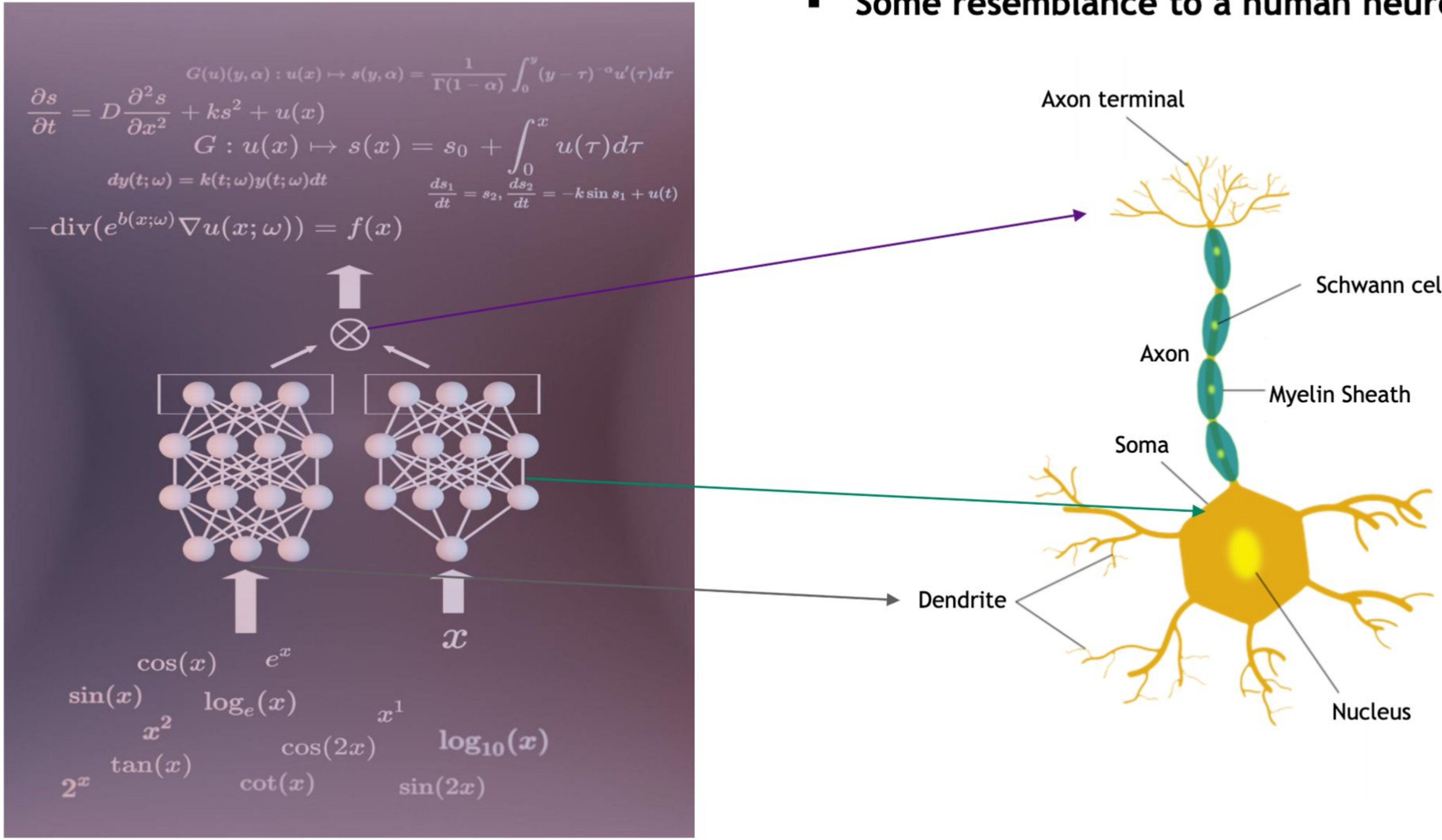
- Operator: function (∞-dim) $\mapsto$ function (∞-dim)
  e.g., derivative (local): $x(t) \mapsto x'(t)$
  e.g., integral (global): $x(t) \mapsto \int K(s,t)x(s)ds$

  e.g., dynamic system: $x(t)$  $y(t)$

  e.g., biological system
  e.g., social system

Deep Operator Network (DeepONet)

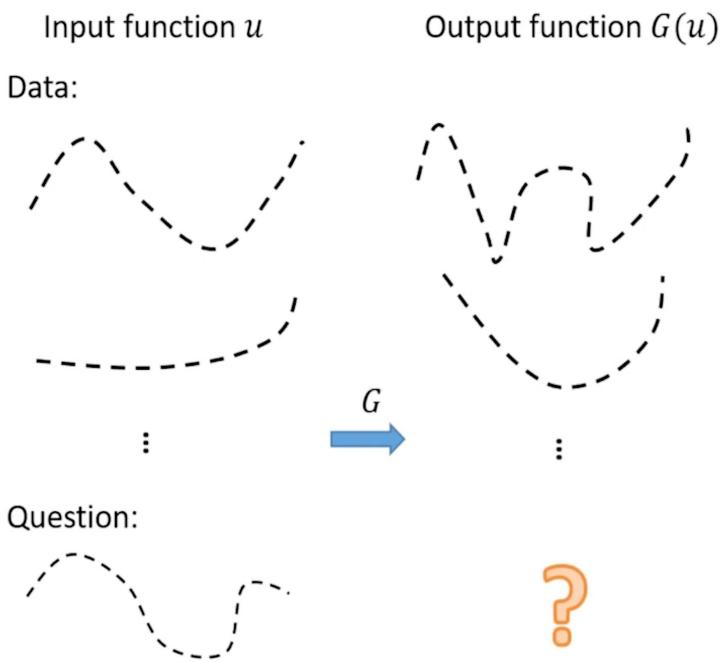
- **Some resemblance to a human neuron**

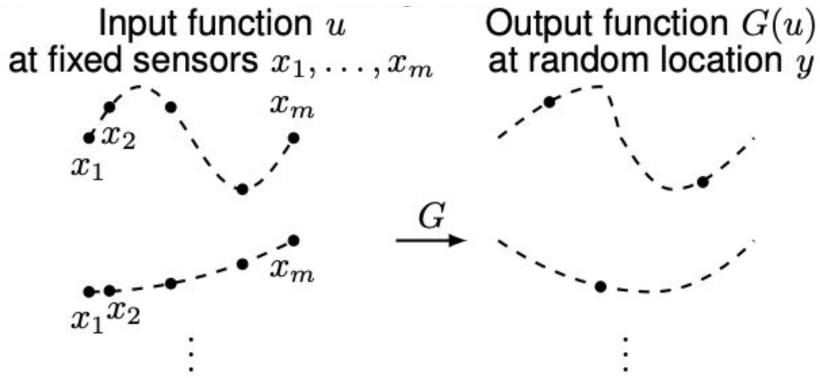


Problem Setup

$G : u \mapsto G(u)$

$G(u) : y \in \mathbb{R}^d \mapsto G(u)(y) \in \mathbb{R}$





$G(u)(y) \approx \sum_{k=1}^{p} b_k t_k$

- Inputs: u at sensors $\{x_1, x_2, \ldots, x_m\}, y$
- Outputs: $G(u)(y)$
- $G(u)(y)$: a function of y conditioning on u
  - $t_k(y)$: basis functions of y
  - $b_k(u)$: u-dependent coefficient

Universal Approximation Theorem for Operator

Theorem 1 (Universal Approximation Theorem for Operator). *Suppose that σ is a continuous non-polynomial function, X is a Banach space, $K_1 \subset X$, $K_2 \subset \mathbb{R}^d$ are two compact sets in X and $\mathbb{R}^d$, respectively, V is a compact set in $C(K_1)$, G is a nonlinear continuous operator, which maps V into $C(K_2)$. Then for any $\epsilon > 0$, there are positive integers n, p and m, constants c_i^k, ξ_{ij}^k, θ_i^k, $\zeta_k \in \mathbb{R}$, $w_k \in \mathbb{R}^d$, $x_j \in K_1$, $i = 1, \dots, n$, $k = 1, \dots, p$ and $j = 1, \dots, m$, such that*

$$\left| G(u)(y) - \sum_{k=1}^{p} \underbrace{\sum_{i=1}^{n} c_i^k \sigma \left(\sum_{j=1}^{m} \xi_{ij}^k u(x_j) + \theta_i^k \right)}_{\text{branch}} \underbrace{\sigma(w_k \cdot y + \zeta_k)}_{\text{trunk}} \right| < \epsilon \tag{1}$$

holds for all $u \in V$ and $y \in K_2$. Here, $C(K)$ is the Banach space of all continuous functions defined on K with norm $\| f \|_{C(K)} = \max_{x \in K} |f(x)|$.

Generalized Universal Approximation Theorem for Operator

Theorem 2 (Generalized Universal Approximation Theorem for Operator). *Suppose that X is a Banach space, $K_1 \subset X$, $K_2 \subset \mathbb{R}^d$ are two compact sets in X and $\mathbb{R}^d$, respectively, V is a compact set in $C(K_1)$. Assume that $G: V \to C(K_2)$ is a nonlinear continuous operator. Then, for any $\epsilon > 0$, there exist positive integers m, p, continuous vector functions $\mathbf{g} : \mathbb{R}^m \to \mathbb{R}^p$, $\mathbf{f} : \mathbb{R}^d \to \mathbb{R}^p$, and $x_1, x_2, \ldots, x_m \in K_1$, such that*

$$\left| G(u)(y) - \langle \underbrace{\mathbf{g}(u(x_1), u(x_2), \cdots, u(x_m))}_{\text{branch}}, \underbrace{\mathbf{f}(y)}_{\text{trunk}} \rangle \right| < \epsilon$$

holds for all $u \in V$ and $y \in K_2$, where $\langle \cdot, \cdot \rangle$ denotes the dot product in $\mathbb{R}^p$. Furthermore, the functions $\mathbf{g}$ and $\mathbf{f}$ can be chosen as diverse classes of neural networks, which satisfy the classical universal approximation theorem of functions, for example, (stacked/unstacked) fully connected neural networks, residual neural networks and convolutional neural networks.

Sketch of Proof

Continuously extend $\mathcal{G}(u)(y), y \in K_2$ to $\mathcal{G}(u)(y), y \in D$.

$$
\begin{aligned}
\mathcal{G}(u) &\approx \mathcal{G}(\mathcal{I}^0_{m,x} u) \quad \text{(piecewise constant interpolation)} \\
&\approx \sum_{k=1}^{p} \int_D \mathcal{G}(\mathcal{I}^0_{m,x} u) e_k(y) dy \; e_k(y), \quad (K_2 \subset D, \text{spectral expansion}) \\
&\approx \sum_{k=1}^{p} \int_D \mathcal{I}^0_{n,y}(\mathcal{G}(\mathcal{I}^0_{m,x} u)(y)) e_k(y) dy e_k(y) \quad \text{(interpolation)} \\
&= \sum_{k=1}^{p} (\sum_{i=1}^{n} \mathcal{G}(\mathcal{I}^0_{m,x} u)(y_i) \underbrace{\int_D e_k(y) \chi_{D_i}(y) dy}_{c_i^k}) e_k(y)
\end{aligned}
$$

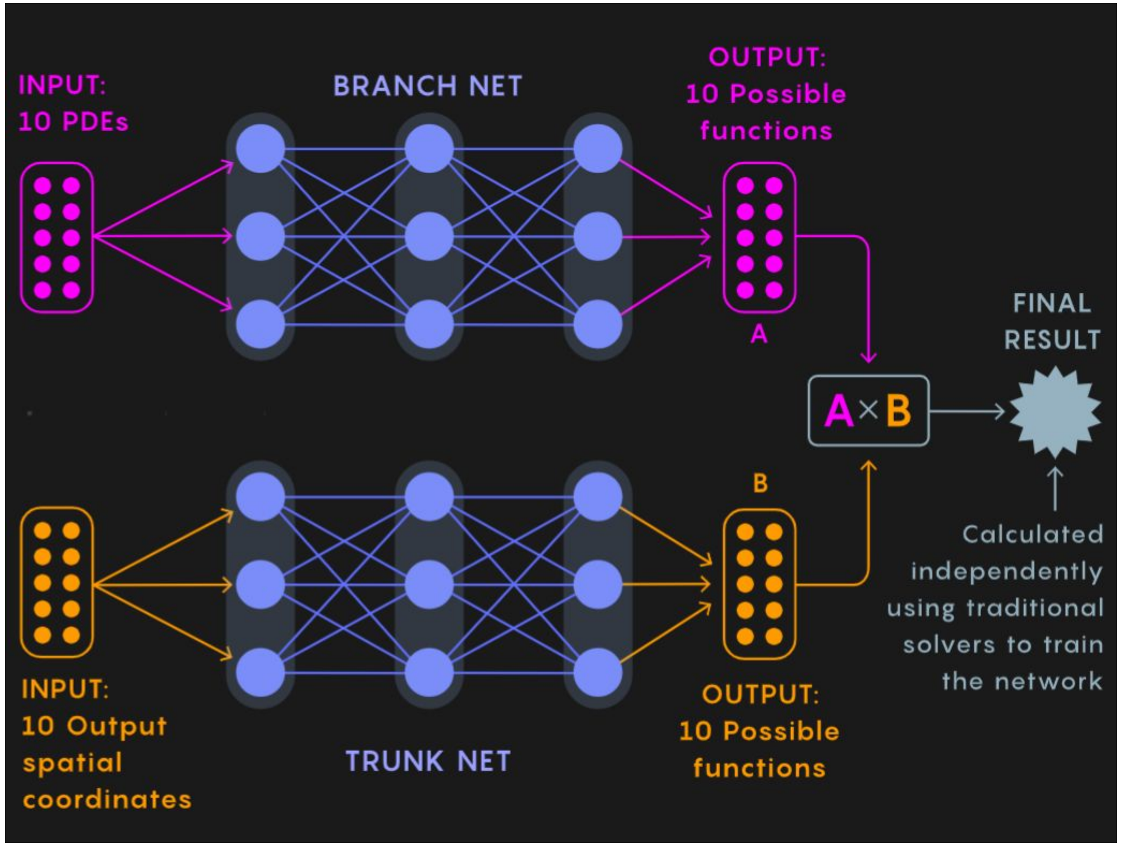
- $\mathcal{G}\left(\mathcal{I}^0_m u\right)(y_i)$ is continuous in u_m, on $[-M, M]^m$, $M = \max_{1 \le i \le m} |u(x_i)|$.
- As $\mathcal{G}: V \to C(D)$ is continuous, we have uniform approximation

$$
\sup_{u \in V} \sup_{u_m \in [-M,M]^m} \left| \mathcal{G}\left(\mathcal{I}^0_m u\right)(y_i) - g^{\mathcal{N}}\left(u_m; \Theta^{(k,i)}\right) \right| < \epsilon
$$

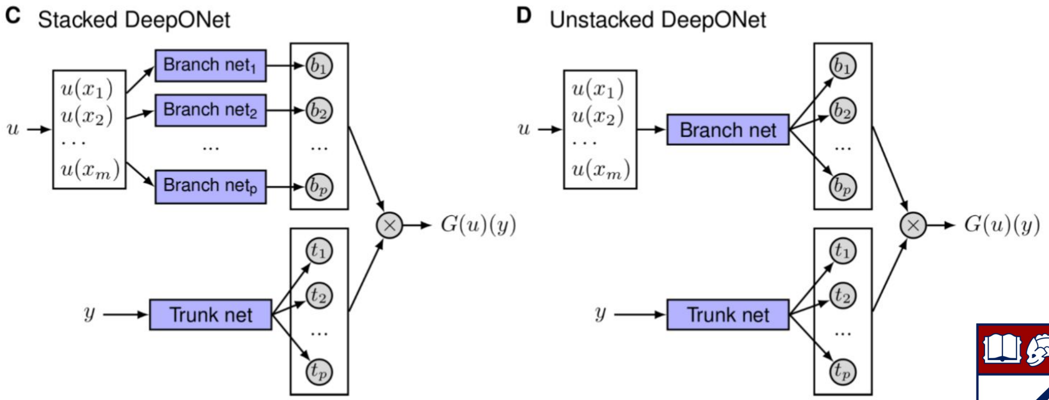
- $e_k(y) \approx f^{\mathcal{N}}\left(y; \theta^{(k)}\right)$

B. Deng, Y. Shin, L. Lu, Z. Zhang, & G. E. Karniadakis. Convergence rate of DeepONets for learning operators arising from advection-diffusion equations. arXiv preprint arXiv:2102.10621, 2021.

DeepONet



$$G(u)(y) \approx \sum_{k=1}^{p} \underbrace{\sum_{i=1}^{n} c_i^k \sigma\left(\sum_{j=1}^{m} \xi_{ij}^k u(x_j) + \theta_i^k\right)}_{branch} \underbrace{\sigma(w_k \cdot y + \zeta_k)}_{trunk}$$



Error Estimates

For all the explicit and implicit operators in our examples, the operators G are Hölder continuous.

$$\|G(f) - G(g)\|_X \leq C\,\|f - g\|_Y^\alpha, \quad 0 < \alpha \leq 1.$$

Here $C > 0$ depends on f and g and the operator G. Here X and Y are Banach spaces and they refer to the space of continuous functions on a compact set unless otherwise stated.

Let $G_{\mathbb{N}}$ be the approximated G using DeepONet. Let f_h be an approximation of f in Y, possibly by collocation or neural networks. Then

$$\|G(f) - G_{\mathbb{N}}(f_h)\|_X \leq \|G(f) - G(f_h)\|_X + \|G(f_h) - G_{\mathbb{N}}(f_h)\|_X \leq C\,\|f - f_h\|_Y^\alpha + \varepsilon,$$

where ε is the user-defined accuracy as in the universal approximation theorem by neural networks and thus the key is to verify the operator G is Hölder continuous.

The explicit operators and their Lipschitz continuity ($\alpha = 1$) are presented below.

1. (Simple ODE, **Problem 1.A**) $G(u)(x) = s_0 + \int_0^x u(s)\,ds$.

$$\max_{x\in[0,b]} |G(u)(x) - G(v)(x)| \leq b \max_{x\in[0,b]} |u - v|\,.$$

2. (Caputo derivative, **Problem 2**) The operator is Lipschitz continuous with respect to its argument in weighted Sobolev norms; see e.g. [14, Theorem 2.4].

3. (Integral fractional Laplacian, **Problem 3**) The operator is Lipschitz continuous with respect to its argument in weighted Sobolev norms, see e.g. [6, Theorem 3.3].

4. (Legendre transform, **Problem 7** in Equation (S1)). The Lipschitz continuity of the operator $G(u)(n) = \int_{-1}^1 P_n(x)u(x)\,dx$ can be seen as follows. For any non-egative integer,

$$\max_n |G(u)(n) - G(v)(n)| \leq \max_n \int_{-1}^1 |P_n(x)|\,dx \max_{x\in[-1,1]} |u - v| \leq C \max_{x\in[-1,1]} |u - v|$$

   where $C = \max_n \left(\int_{-1}^1 dx\right)^{1/2}\left(\int_{-1}^1 |P_n(x)|^2\,dx\right)^{1/2} = \max_n (2\frac{2}{2n+1})^{1/2} \leq 2$.

5. The linear operator from **Problem 9** in (S2) in Lipschitz continuous with respect to the initial condition in the norm in the space of continuous functions, from he classical theory for linear parabolic equations [9, Chapter IV],

The implicit operator from **Problem 6**. The operator can be written as $u = G(b)$ and $u_i = G(b_i)$, $i = 1, 2$ satisfying the following equations:

$$-\mathrm{div}(e^{b_i(x)}\nabla u_i) = f(x), \quad x \in D = (0,1), \qquad u_i(x) = 0, \quad x \in \partial D,$$

Then $\|u_i(\omega)\|_{H^1} \leq (\min_{x\in D} e^{-b_i})^{-1}\,\|f\|_{H^{-1}}$, where H^1 and H^{-1} are standard Sobolev-Hilbert spaces. The difference $u_1 - u_2$ satisfies the following

$$-\mathrm{div}(e^{b_1(x)}\nabla(u_1 - u_2)) = \mathrm{div}((e^{b_1(x)} - e^{b_2(x)})\nabla u_2), \quad x \in D, \qquad u_1 - u_2 = 0, \quad x \in \partial D.$$

Then by the stability of the elliptic equation , we have

$$\begin{aligned}
\|u_1(\omega) - u_2(\omega)\|_{H^1} &\leq (\min_{x\in D} e^{b_i(x)})^{-1} \left\|(e^{b_1} - e^{b_2})\nabla u_2\right\|_{L^2} \\
&= (\min_{x\in D} e^{b_1(x)})^{-1} \left\|e^{b_1} - e^{b_2}\right\|_{C(D)} \|u_2\|_{H^1} \\
&\leq (\min_{x\in D} e^{b_1(x)})^{-1}(\min_{x\in D} e^{b_2(x)})^{-1} \left\|e^{b_1} - e^{b_2}\right\|_{C(D)} \|f\|_{H^{-1}}\,.
\end{aligned}$$

Then by the mean value theorem, $|e^x - e^y| \leq |x - y|\,(e^x + e^y)$ holds for all $x, y \in \mathbb{R}$. Thus,

$$\|u_1(\omega) - u_2(\omega)\|_{H_0^1} \leq (\min_{x\in D} e^{b_1(x)})^{-1}(\min_{x\in D} e^{b_2(x)})^{-1}(\left\|e^{b_1}\right\|_{C(D)} + \left\|e^{b_2}\right\|_{C(D)})\,\|b_1 - b_2\|_{C(D)}\,\|f\|_{H^{-1}}\,.$$

According to [3, Proposition 2.3], all the random variables $\min_{x\in D} e^{b_i(x)})^{-1}$ and $\left\|e^{b_i}\right\|_{C(D)}$, $i = 1, 2$ have any moments of finite order. Then, we obtain the pathwise Lipschitz continuity of the operator G

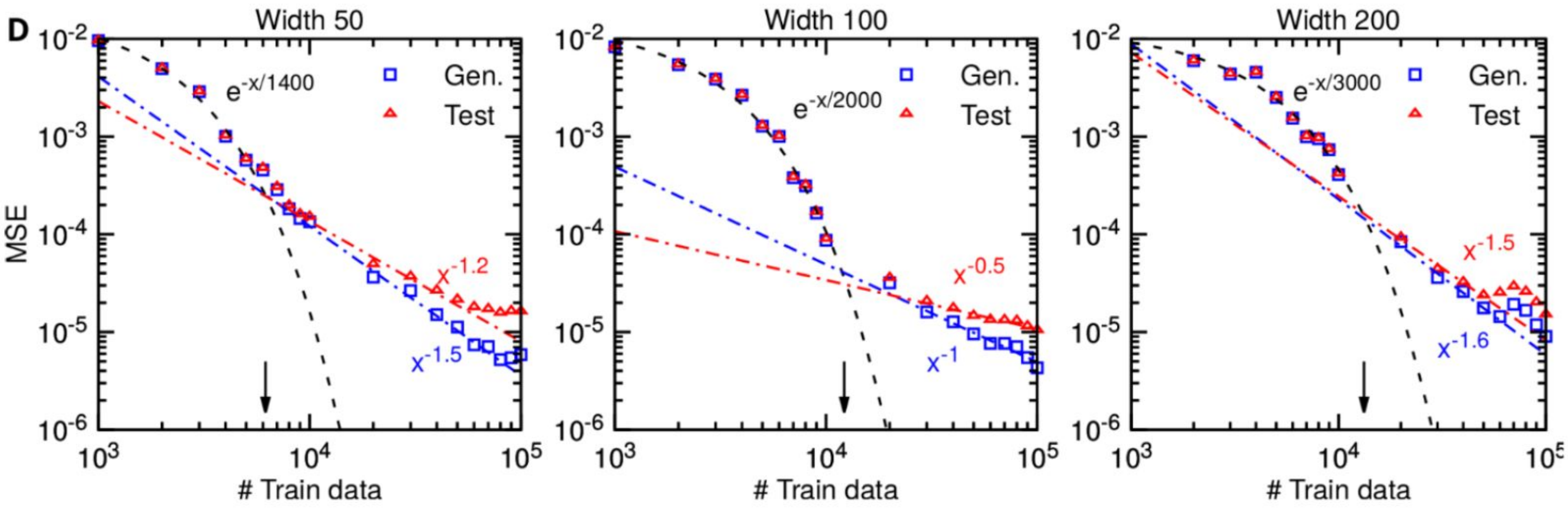
$$\|\mathcal{G}(b_1)(\omega) - \mathcal{G}(b_2)(\omega)\|_{H^1} \leq C(\omega)\,\|b_1 - b_2\|_{C(D)}\,.$$

Here $C(\omega) = (\min_{x\in D} e^{b_1(x)})^{-1}(\min_{x\in D} e^{b_2(x)})^{-1}(\left\|e^{b_1}\right\|_{C(D)} + \left\|e^{b_2}\right\|_{C(D)})\,\|f\|_{H^{-1}}$.

Lu L, Jin P, Pang G, Zhang Z, Karniadakis GE. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. Nature Machine Intelligence. 2021 Mar;3(3):218-29

Experiment Results: Gravity Pendulum

$$\frac{ds_1}{dt} = s_2, \quad \frac{ds_2}{dt} = -k \sin s_1 + u(t)$$



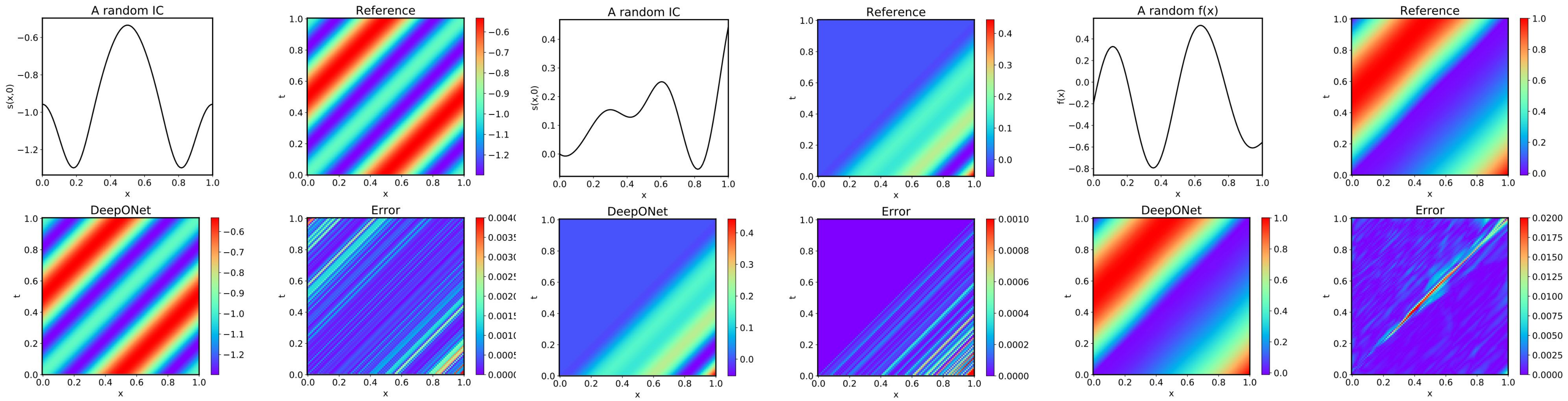
Test/generalization error:

- Small dataset: exponential convergence
- Large dataset: polynomial rates
- Smaller network has earlier transition point

Lu L, Jin P, Pang G, Zhang Z, Karniadakis GE. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. Nature Machine Intelligence. 2021 Mar;3(3):218-29

Experiment Results: Advection Equation

$$\frac{\partial s}{\partial t} + a(x)\frac{\partial s}{\partial x} = 0, \quad x \in [0,1], t \in [0,1]$$



Case	$a(x)$	IC $s(x,0)$	BC	Input	Length scale l	Test MSE
I	1	$f(\sin^2(\pi x))$	Periodic	$s(x,0)$	0.5	$1.8 \times 10^{-6} \pm 4.2 \times 10^{-7}$
II	1	$xf(x)$	$s(0,t)=0$	$s(x,0)$	0.2	$3.7 \times 10^{-7} \pm 1.8 \times 10^{-7}$
III	$1 + 0.1 \cdot f(x)$	x^2	$s(0,t)=\sin(\pi t)$	$f(x)$	0.2	$1.6 \times 10^{-5} \pm 6.8 \times 10^{-6}$
IV	$1 + 0.1 \cdot \frac{f(x)+f(1-x)}{2}$	$\sin(2\pi x)$	Periodic	$f(x)$	0.2	$3.2 \times 10^{-5} \pm 1.4 \times 10^{-5}$

Lu L, Jin P, Pang G, Zhang Z, Karniadakis GE. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. Nature Machine Intelligence. 2021 Mar;3(3):218-29

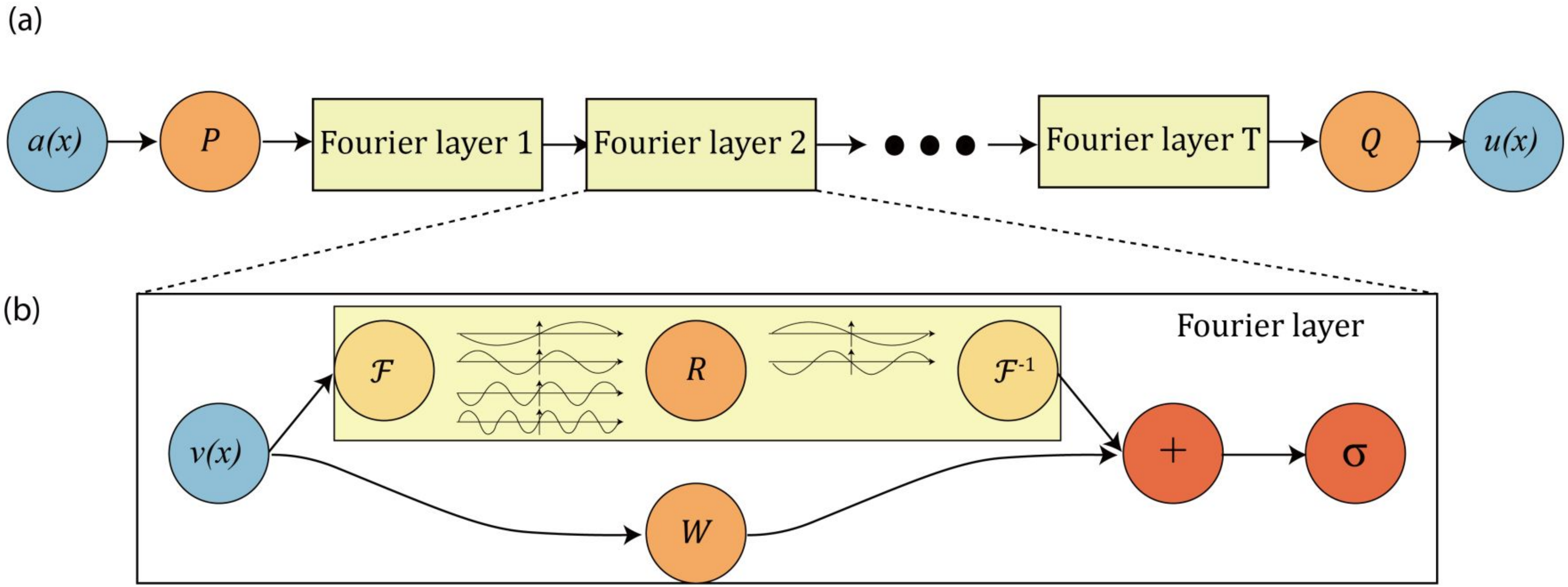
DeepONets for learning operator

- DeepONet
  - 16 ODEs/PDEs (nonlinear, fractional & stochastic) (*Lu et al., Nature Mach Intell, 2021*)
  - Bubble growth dynamics (*Lin, et al., J Chem Phys, 2021; Lin, et al., J Fluid Mech, 2021*)
  - Linear instability waves in high-speed boundary layers (*Di Leoni, et al., arXiv:2105.08697*)

- DeepM&Mnet
  - Electroconvection (*Cai, et al., J Comput Phys, 2021*)
  - Hypersonics (*Mao, et al., J Comput Phys, 2021*)

- Extensions of DeepONet, e.g., POD-DeepONet, MIO-Net, PI-DeepONet, V-DeepONet…

Fourier Neural Operator



Li Z, Kovachki N, Azizzadenesheli K, Liu B, Bhattacharya K, Stuart A, Anandkumar A. Fourier neural operator for parametric partial differential equations. arXiv preprint arXiv:2010.08895. 2020 Oct 18.

Structure of FNO

Step 1: Function value $v(x)$ is lifted to a higher dimensional representation $z_0(x)$ by

$$z_0(x) = P(v(x)) \in \mathbb{R}^{d_z}$$

Transformation $P : \mathbb{R} \to \mathbb{R}^{d_z}$ is a shallow fully-connected NN or simply a linear layer. d_z is like the channel size in CNN.

Step 2: L Fourier layers are applied iteratively to z_0. z_L is the output of the last Fourier layer, and the dimension of $z_L(x)$ is d_z.

Step 3: Transformation $Q : \mathbb{R}^{d_z} \to \mathbb{R}$ is applied to project $z_L(x)$ to the output by

$$u(x) = Q(z_L(x))$$

Q is parameterized by a fully-connected NN.

Li Z, Kovachki N, Azizzadenesheli K, Liu B, Bhattacharya K, Stuart A, Anandkumar A. Fourier neural operator for parametric partial differential equations. arXiv preprint arXiv:2010.08895. 2020 Oct 18.

Fourier Layer using FFT

For the output of the lth Fourier layer z_l with d_v channels:

Step 1: Compute the transform by FFT $\mathcal{F}$ and inverse FFT $\mathcal{F}^{-1}$:

$$\mathcal{F}^{-1}\left(R_l \cdot \mathcal{F}(z_l)\right)$$

$\mathcal{F}$ is applied to each channel of z_l separately Truncate the higher modes of $\mathcal{F}(z_l)$, keeping only the first k Fourier modes in each channel. So $\mathcal{F}(z_l)$ has the shape $d_v \times k$.

Step 2: Apply a different (complex-number) weight matrix of shape $d_v \times d_v$ for each mode index of $\mathcal{F}(z_l)$ Have k trainable matrices, which form a weight tensor $R_l \in \mathbb{C}^{d_v \times d_v \times k}$. $R_l \cdot \mathcal{F}(z_l)$ has the same shape of $d_v \times k$ as $\mathcal{F}(z_l)$.

Step 3: Inverse FFT Need to append zeros to $R_l \cdot \mathcal{F}(z_l)$ to fill in the truncated modes. Moreover, in each Fourier layer, a residual connection with a weight matrix $W_l \in \mathbb{R}^{d_v \times d_v}$. The output of the $(l+1)$th Fourier layer z_{l+1} is $\boldsymbol{z}_{l+1} = \sigma\left(\mathcal{F}^{-1}\left(R_l \cdot \mathcal{F}\left(\boldsymbol{z}_l\right)\right) + W_l \cdot \boldsymbol{z}_l + \boldsymbol{b}_l\right)$

Li Z, Kovachki N, Azizzadenesheli K, Liu B, Bhattacharya K, Stuart A, Anandkumar A. Fourier neural operator for parametric partial differential equations. arXiv preprint arXiv:2010.08895. 2020 Oct 18.

DeepONet and FNO

Then $u(x,t)$ is a rational function in $\mathbf{V}_m := (V_0, V_1, \cdots, V_{m-1})^\top$. By [Telgarsky 2017], there exists $\tilde{g}^{\mathcal{N}}(\mathbf{V}_m; \theta_{x,t})$ of size $O(m^2 \ln(\epsilon^{-1}))$ for fixed x, t s.t.

$$\sup_{u_0 \in S} \mid \tilde{g}^{\mathcal{N}}(\mathbf{V}_m; \theta_{x,t})\Big) - G(u_0) \mid \leq C\epsilon$$
$$S := \{v : \|v\|_{L^\infty} \leq M_0, \|\partial_x v\|_{L^\infty} \leq M_1\}$$

- $\tilde{g}^{\mathcal{N}}(\mathbf{V}_m; \theta_{x,t}))$ can be further approximated by a ReLU network
- $g^{\mathcal{N}}(\mathbf{u}_{0,m}; \Theta_{x,t})$ with input $\mathbf{u}_{0,m}$ (initial values).
- $g^{\mathcal{N}}(\mathbf{u}_{0,m}; \Theta_{x,t})$ can be viewed as FNO.
- $g^{\mathcal{N}}(\mathbf{u}_{0,m}; \Theta_{x,t})$ can serve as the branch of the DeepONet

$\sum_{k=1}^{p} g^{\mathcal{N}}(\mathbf{u}_{0,m}; \Theta_{x_k,t}) L_k$, where x_k 's are Fourier collocation points and $L_k(x)'s$ are the Lagrange basis.

Conclusion

- Take $\epsilon = m^{-1}$ (accuracy), the size of FNO is $O(m^3 \ln(m))$ while the size of DeepONet is $O(m^3 \ln(m))$ (branch) $+O(m)$ (trunk).
- In general, we connect FNO $F^{\mathcal{N}}(v)$ and DeepONet by

$$G^v(v)(x) = \sum_{k=1}^{m} F^N(v)(x_k) L_k(x)$$

Consider the Burgers' equation $u_t + uu_x = \kappa u_{xx}$ with periodic boundary condition $u(x-\pi, t) = u(x+\pi, t), x \in R$. Then, by the Cole-Hopf transformation,

$$\begin{aligned} u(x,t) =: G(u_0) &= \frac{-2\kappa \cdot \int_{\mathbb{R}} \partial_x \mathcal{K}(x,y,t) v_0(y) dy}{\int_{\mathbb{R}} \mathcal{K}(x,y,t) v_0(y) dy} \\ &\approx \frac{-2\kappa \int_{\mathbb{R}} \mathcal{K}(x,y,t) v_0(y) dy}{\int_{\mathbb{R}} \mathcal{K}(x,y,t) (\mathcal{I}_m v_0)(y) dy} \\ &= \frac{V_0 c_0^1 + V_1 c_1^1 + \cdots + V_{m-1} c_{m-1}^1}{V_0 c_0^2 + V_1 c_1^2 + \cdots + V_{m-1} c_{m-1}^2} \end{aligned}$$

where $\mathcal{K}$ is the heat kernel, $\mathcal{I}_m$ is the Fourier interpolation operator and

$$V_0 = 1, V_j = \exp\left(-\sum_k \int_0^{x_j} L_k(y) dy \frac{u_0(y_k)}{2\kappa}\right), j = 1, \cdots, m-1$$

$$c_j^1(x,t) = -2\kappa_i \int_0^x \left(\sum_l \partial_x \mathcal{K}(x, y + 2\pi l, t)\right) L_j(y) dy$$

$$c_j^2(x,t) = \int_0^x \left(\sum_j \mathcal{K}(x, y + 2\pi l, t)\right) L_j(y) dy$$

Conclusion

- Take $\epsilon = m^{-1}$ (accuracy), the size of FNO is $O(m^3 \ln(m))$ while the size of DeepONet is $O(m^3 \ln(m))$ (branch) $+O(m)$ (trunk).

Applications of Neural Operators

DeepONet for Approximating Functionals: Predicting unsteady pressure and lift/drag-force coefficients

- Simulation of NACA0012 airfoil (Nektar by Z. Wang)

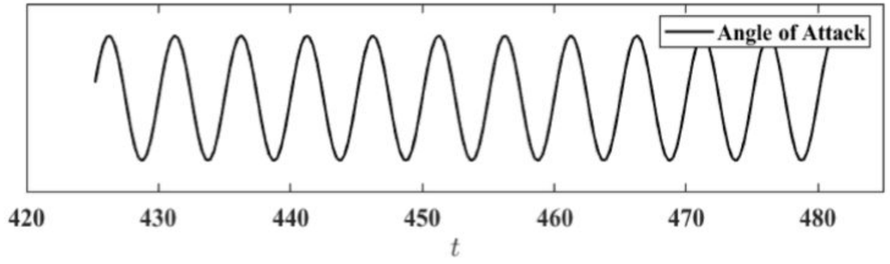
inflow

$$u = U_\infty \cos\left(\frac{\alpha_0 \pi}{180} \times \frac{\sin(2f\pi t) + 1.0}{2}\right),$$

$$v = U_\infty \sin\left(\frac{\alpha_0 \pi}{180} \times \frac{\sin(2f\pi t) + 1.0}{2}\right),$$

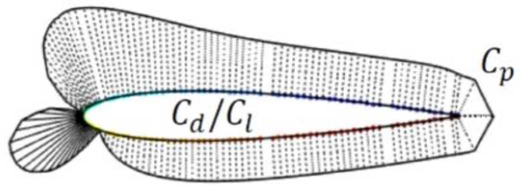
$$\alpha = 15^o \quad f = 0.2 \quad U_\infty = 1 \quad Re = 2500$$

- Generate time-dependent AOA



predict

- Time-dependent coefficients of drag, lift, pressure



Applications of Neural Operators

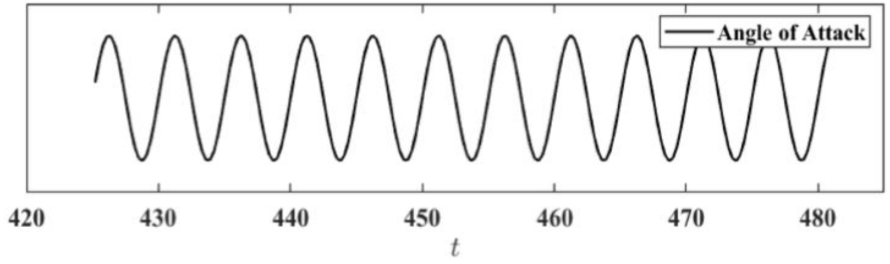
DeepONet for Approximating Functionals: Predicting unsteady pressure and lift/drag-force coefficients

- Simulation of NACA0012 airfoil (Nektar by Z. Wang)

$$u = U_\infty \cos\left(\frac{\alpha_0 \pi}{180} \times \frac{\sin(2 f \pi t) + 1.0}{2}\right),$$
$$v = U_\infty \sin\left(\frac{\alpha_0 \pi}{180} \times \frac{\sin(2 f \pi t) + 1.0}{2}\right),$$

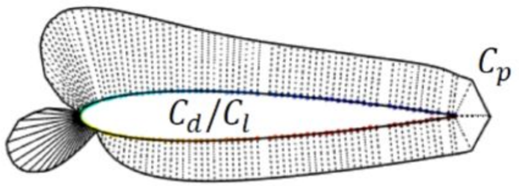
$\alpha = 15^o$
$f = 0.2$
$U_\infty = 1$
$Re = 2500$

- Generate time-dependent AOA



predict

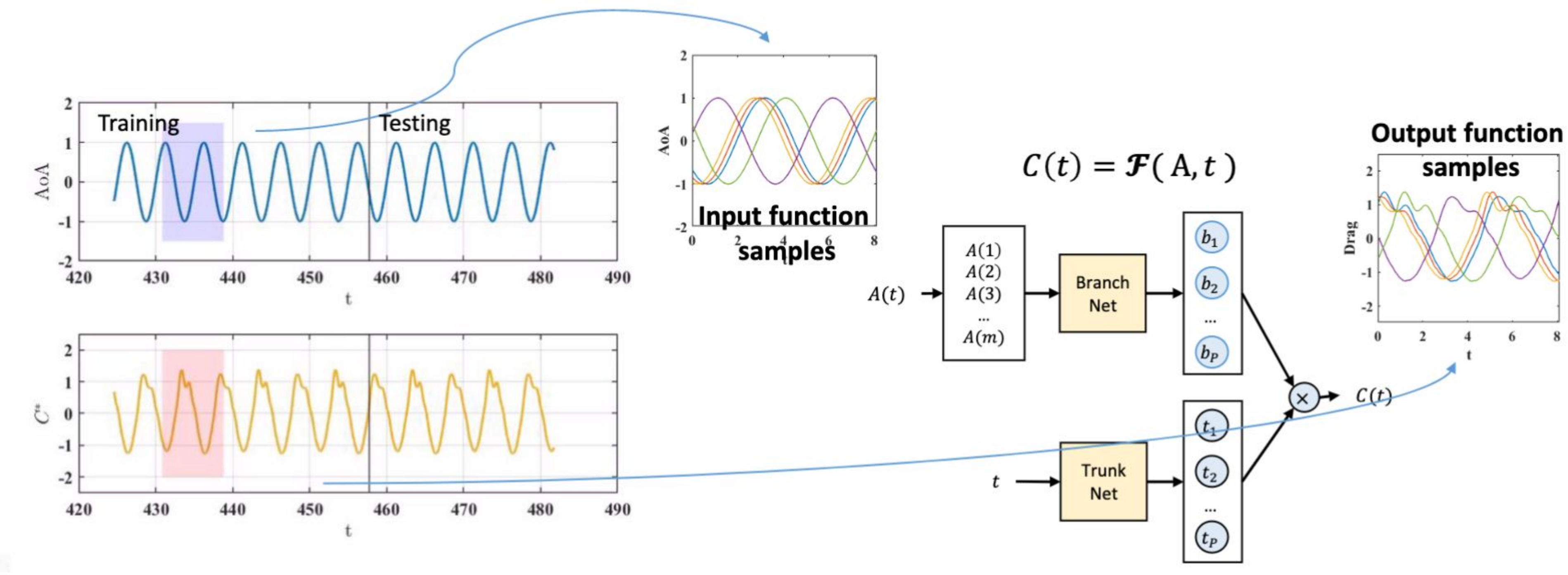
- Time-dependent coefficients of drag, lift, pressure



Applications of Neural Operators

DeepONet for Approximating Functionals: Predicting unsteady pressure and lift/drag-force coefficients

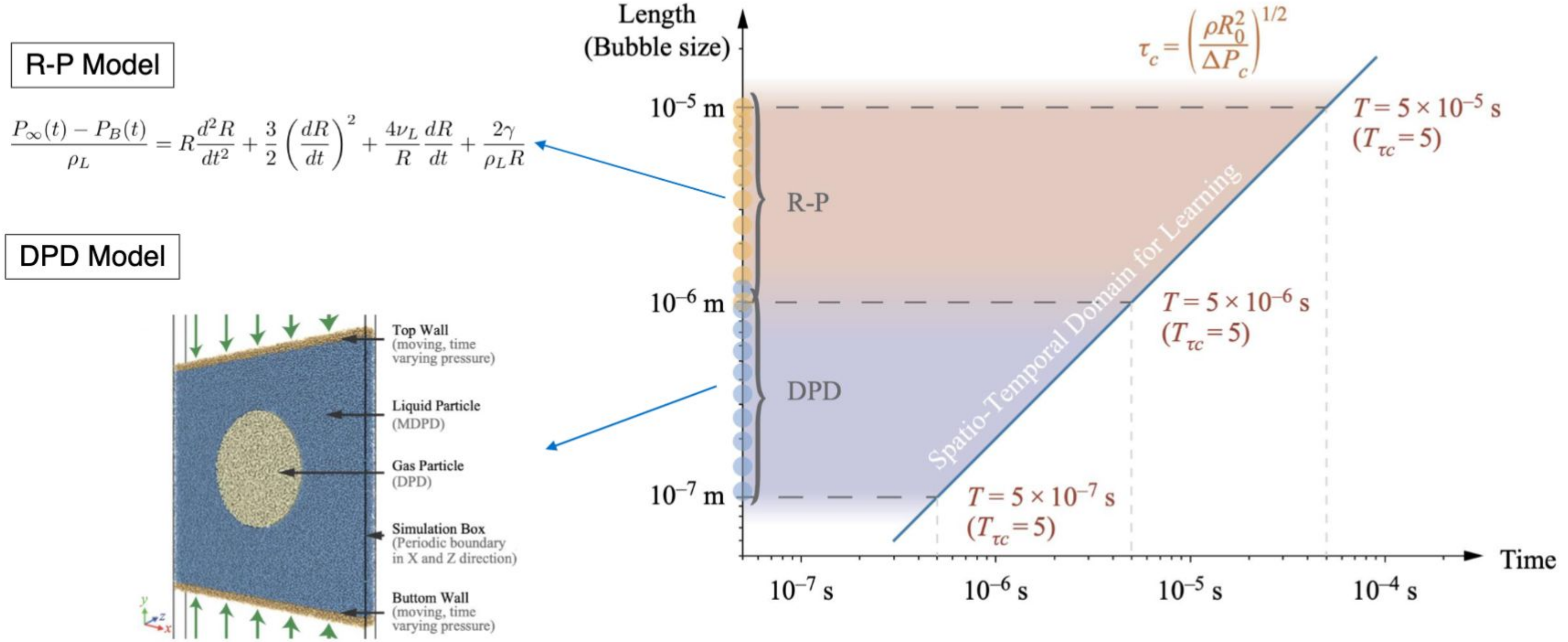
- Predicting drag/lift coefficient
- Cut the time-dependent signal into pieces, to generate a bunch of input-output pairs



Applications of Neural Operators

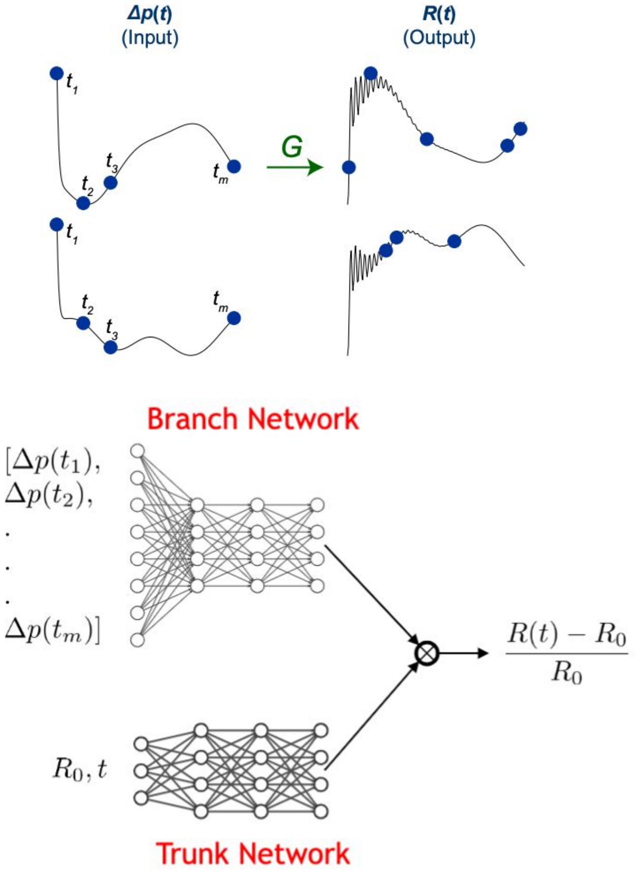
DeepONet for Bubble Dynamics

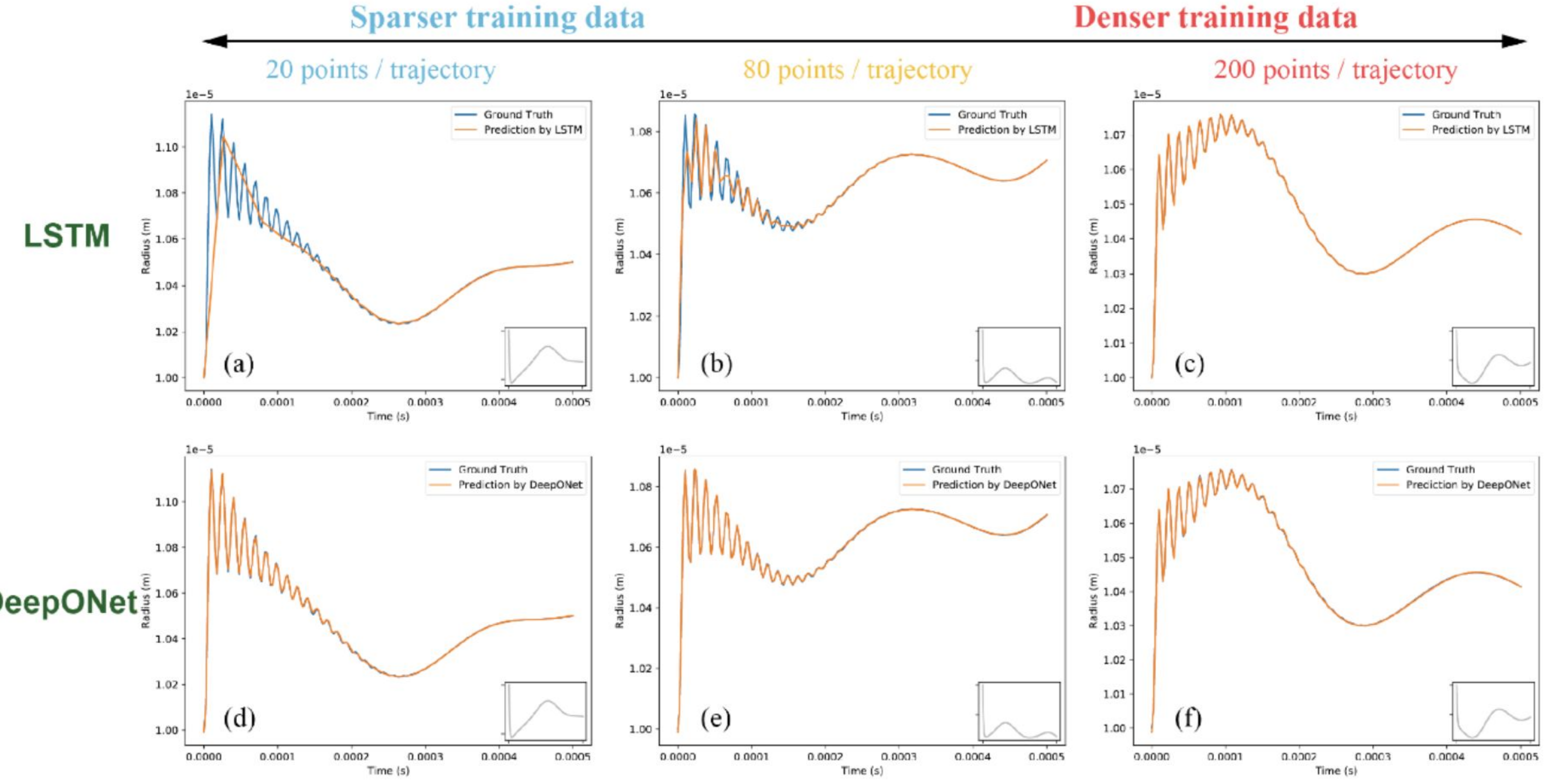
- Rayleigh-Plesset equation is an ordinary differential equation which governs the dynamics of a spherical bubble in an infinite body of imcomporessible fluid
- For nanobubbles, the thermal fluctuation cannot be ignored, and trainning data are generated by particle simulation



Applications of Neural Operators

DeepONet for Bubble Dynamics





References

- Cai S, Wang Z, Lu L, Zaki TA, Karniadakis GE. DeepM&Mnet: Inferring the electroconvection multiphysics fields based on operator approximation by neural networks. Journal of Computational Physics. 2021 Jul 1;436:110296.
- Goswami S, Yin M, Yu Y, Karniadakis GE. A physics-informed variational DeepONet for predicting crack path in quasi-brittle materials. Computer Methods in Applied Mechanics and Engineering. 2022 Mar 1;391:114587.
- Lanthaler S, Mishra S, Karniadakis GE. Error estimates for deeponets: A deep learning framework in infinite dimensions. arXiv preprint arXiv:2102.09618. 2021 Feb 18.
- Li Z, Kovachki N, Azizzadenesheli K, Liu B, Bhattacharya K, Stuart A, Anandkumar A. Fourier neural operator for parametric partial differential equations. arXiv preprint arXiv:2010.08895. 2020 Oct 18.
- Lu L, Jin P, Pang G, Zhang Z, Karniadakis GE. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. Nature Machine Intelligence. 2021 Mar;3(3):218-29.
- Lu L, Meng X, Cai S, Mao Z, Goswami S, Zhang Z, Karniadakis GE. A comprehensive and fair comparison of two neural operators (with practical extensions) based on FAIR data. arXiv preprint arXiv:2111.05512. 2021 Nov 10.
- Mao Z, Lu L, Marxen O, Zaki TA, Karniadakis GE. DeepM&Mnet for hypersonics: Predicting the coupled flow and finite-rate chemistry behind a normal shock using neural-network approximation of operators. Journal of Computational Physics. 2021 Dec 15;447:110698.

THANK YOU