Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang and George Em Karniadakis

Nature Machine Intelligence **volume 3**, pages218–229 (2021)

Physical system

- Newton's second law: $$F = ma = m\frac{d^2x}{dt^2}$$

- Elasticity: $$F = kx$$

↓

- Damping force: $$F = f\frac{dx}{dt}$$

- Burgers equation: $$\frac{\partial y}{\partial t} + y\frac{\partial y}{\partial x} = d\frac{\partial^2 y}{\partial x^2}$$

Operator: An operation on a function

- Differential operator:

$$D = \frac{d}{dx}$$

- gradient operator:

$$\nabla f(x,y) = \{\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\}$$

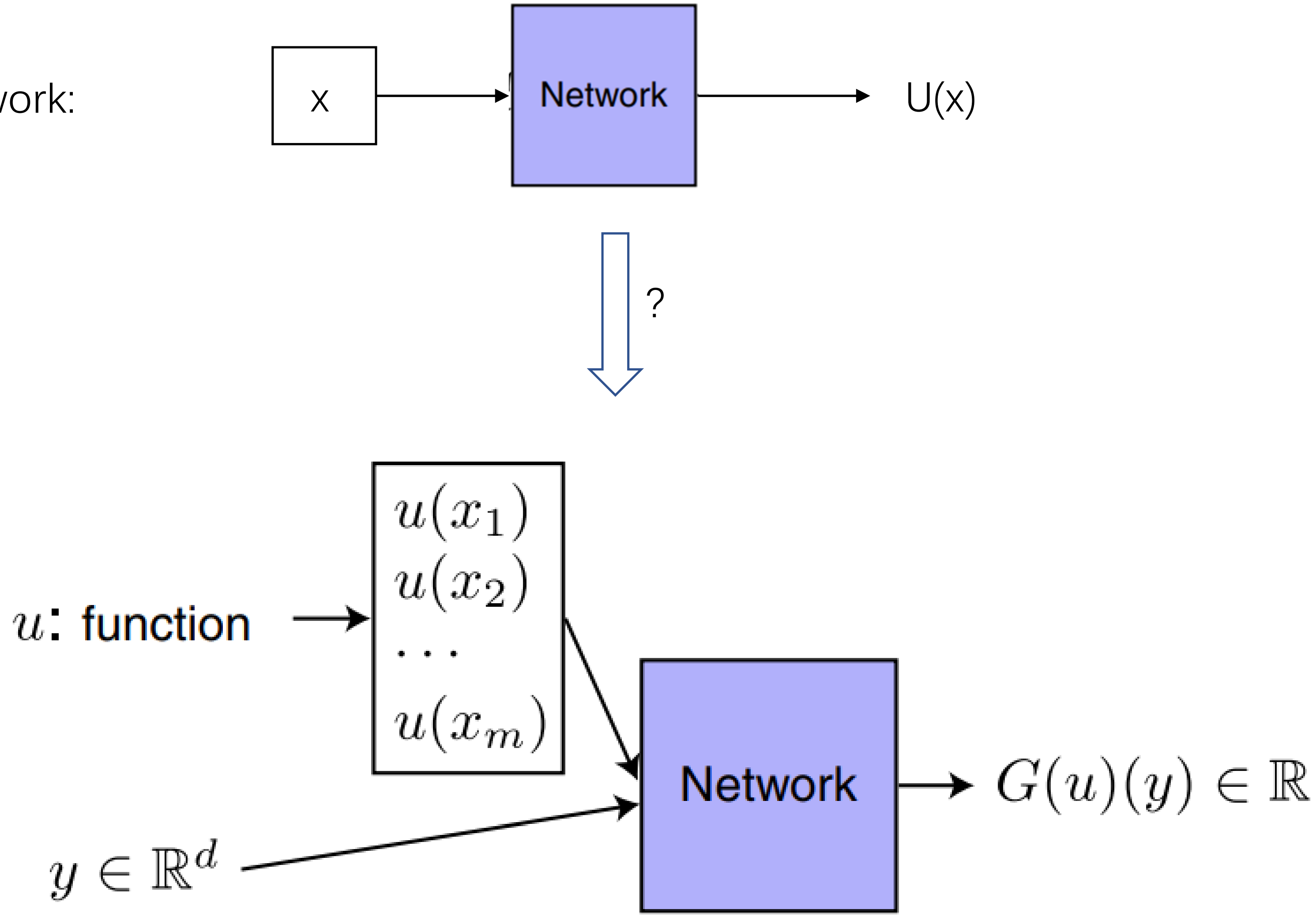
- Laplace operator:

$$\Delta = \sum_{i=1}^{n} \frac{\partial^2}{\partial x_i^2}$$

- CNN:

$$CNN(\mathbf{Image}_1) = \mathbf{Image}_2$$

Neural network:



- neural networks: universal approximators of continuous functions (a space of functions ⟶ real numbers) (widely known)
- a NN with a single hidden layer can accurately approximate any nonlinear continuous operator (a space of functions ⟶ another space of functions) (less known)

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, $\zeta_k \in \mathbb{R}$, $w_k \in \mathbb{R}^d$, $x_j \in K_1$, $i = 1, \ldots, n$, $k = 1, \ldots, p$ and $j = 1, \ldots, m$, such that*

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

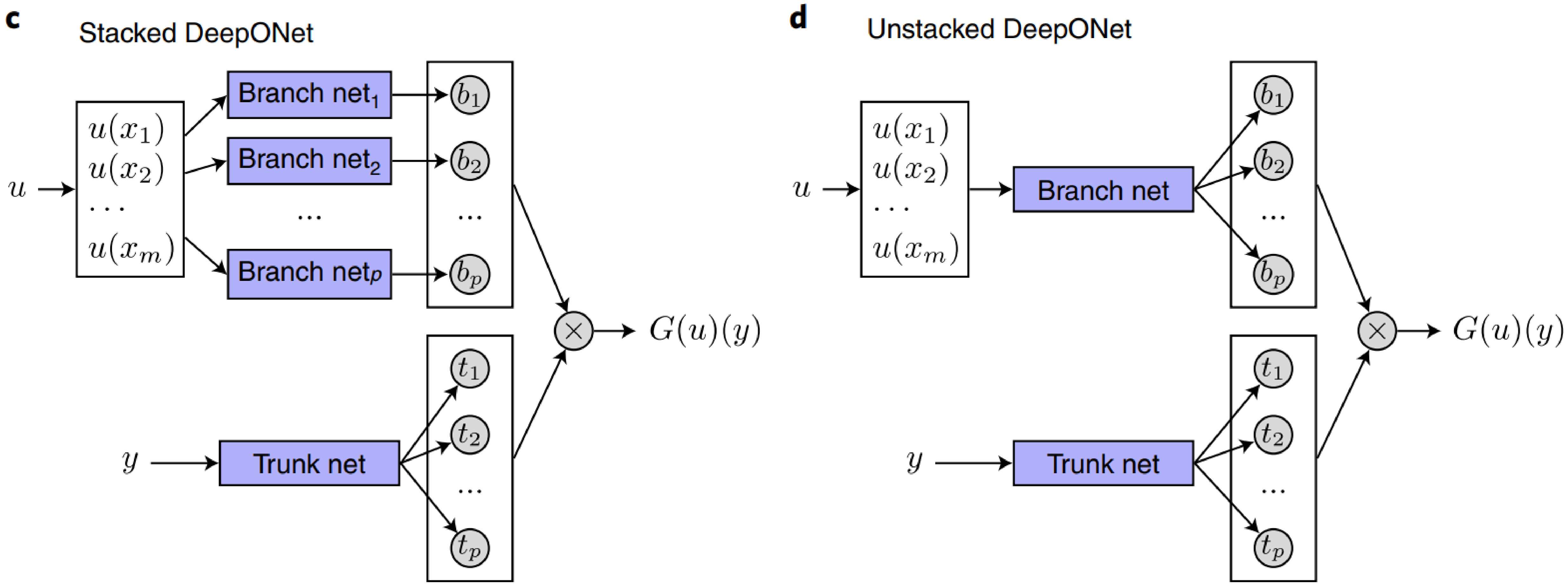
holds for all $u \in V$ and $y \in K_2$. Here, $C(K)$ is the Banach space of all continuous functions defined on K with norm $\|f\|_{C(K)} = \max_{x \in K} |f(x)|$.

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.

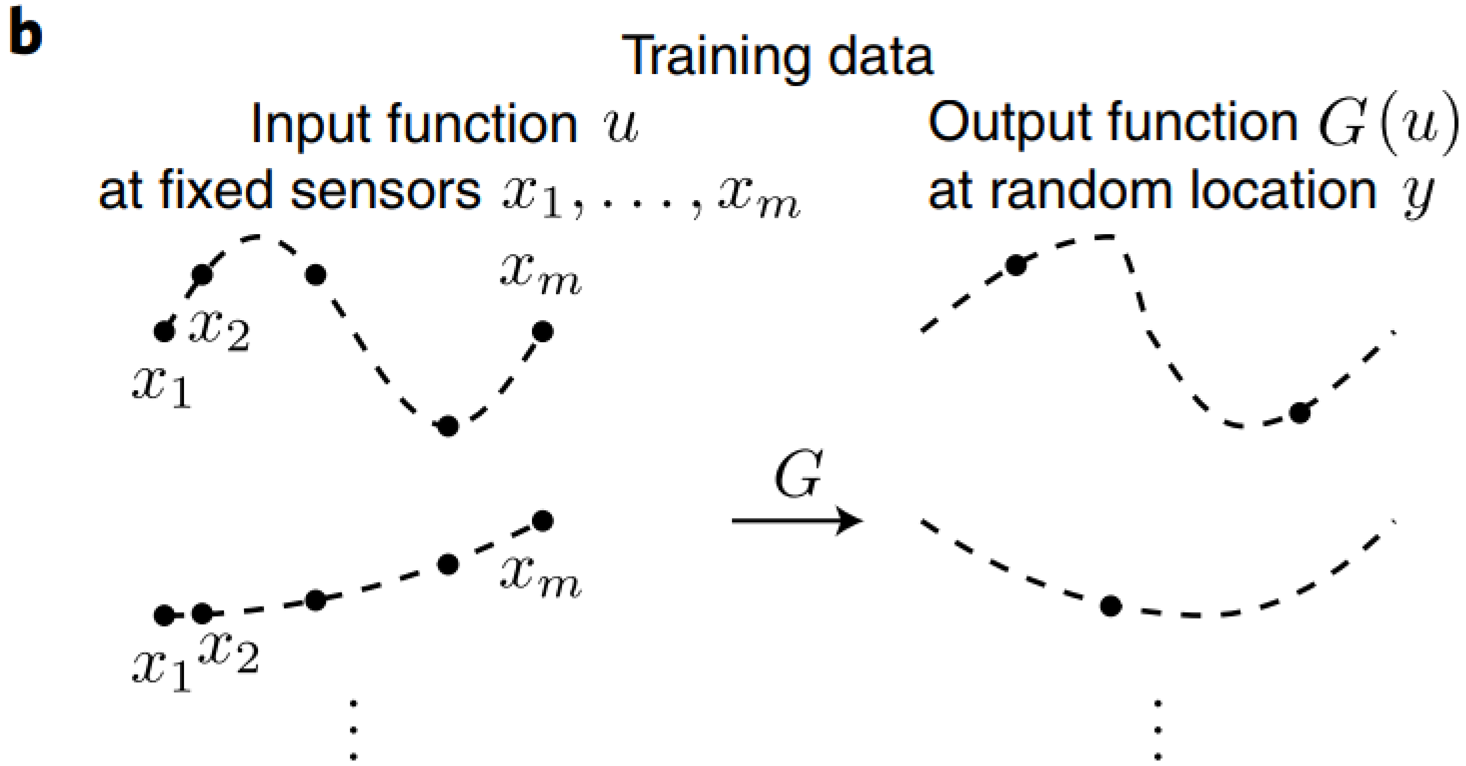
DeepONet: learn diverse continuous nonlinear operators



Data generation: 3 input function spaces

- Gaussian random fields
- spectral representations
- formulating the input functions as images

one data point: triplet (u, y, G(u)(y))

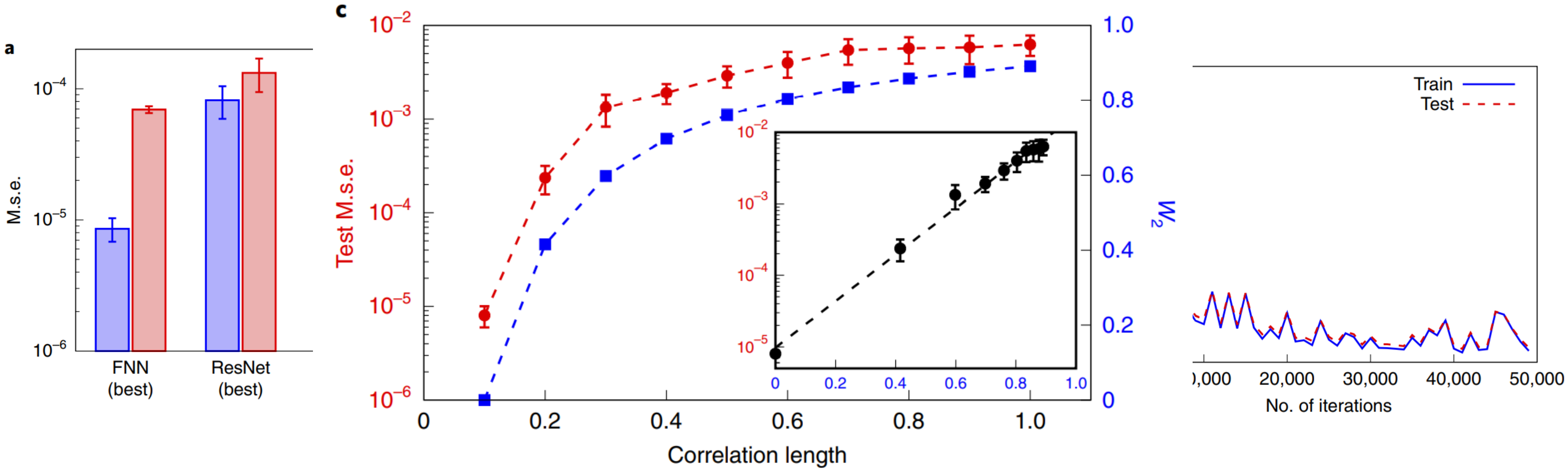


Integral operator: $$\frac{\mathrm{d}s(x)}{\mathrm{d}x} = g(s(x), u(x), x), \quad x \in (0, 1]$$

$$\Downarrow$$

$$\frac{\mathrm{d}s(x)}{\mathrm{d}x} = u(x) \text{ and } G: u(x) \mapsto s(x)$$
$$= s_0 + \int_0^x u(\tau)\mathrm{d}\tau, \ x \in [0, 1]$$

training dataset are sampled from the space of a GRF with the covariance kernel $k_l(x_1, x_2) = \exp(-\| x_1 - x_2\|^2/2l^2)$



fractional differential operators: **d** $G(u)(y,\alpha): u(x) \mapsto s(y,\alpha) = \frac{1}{\Gamma(1-\alpha)} \int_0^y (y-\tau)^{-\alpha} u'(\tau) \mathrm{d}\tau,$

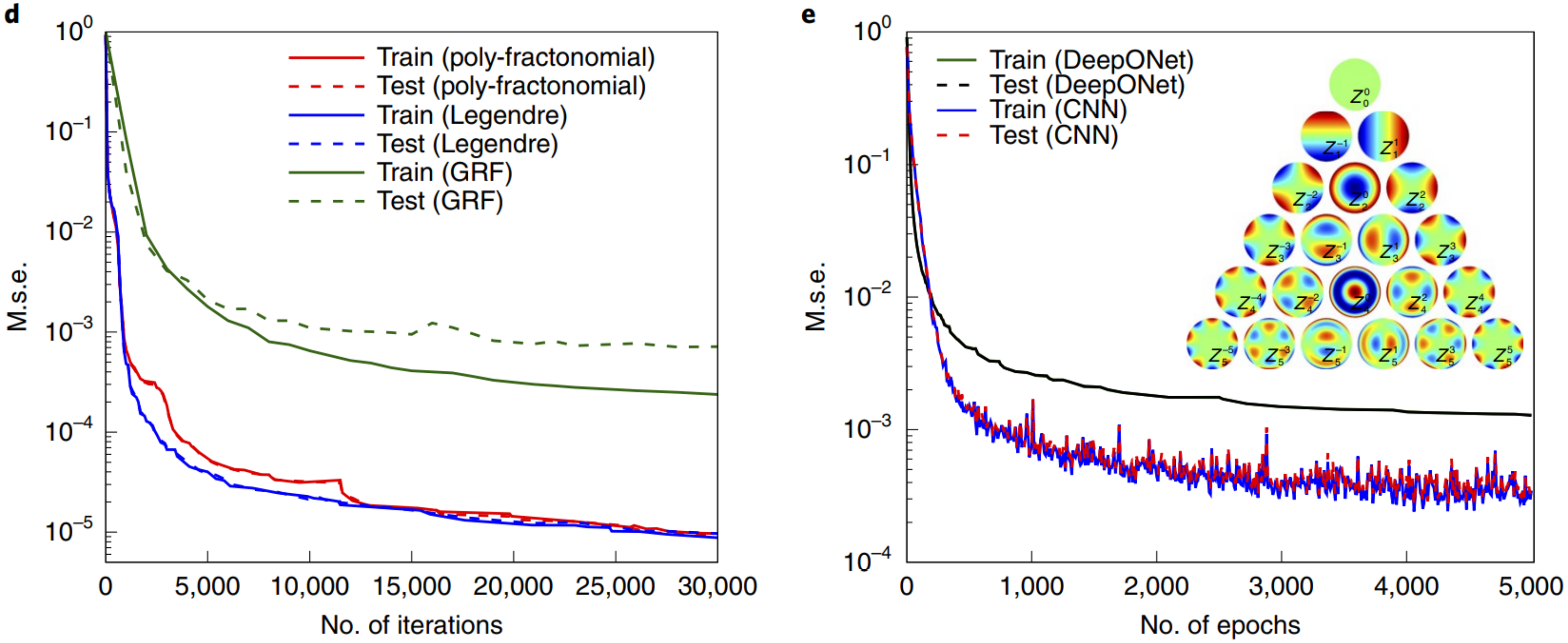
$y \in [0,1], \alpha \in (0,1)$

*1D Caputo fractional derivative

e $G(u)(y,\alpha): u(x) \mapsto s(y,\alpha)$

$= \frac{2^\alpha \Gamma(1+\frac{\alpha}{2})}{\pi |\Gamma(-\frac{\alpha}{2})|} \times \text{p.v.} \int_{\mathbb{R}^2} \frac{u(y)-u(\tau)}{\|y-\tau\|_2^{2+\alpha}} \mathrm{d}\tau, \quad \alpha \in (0,2)$

*2D Riesz fractional Laplacian

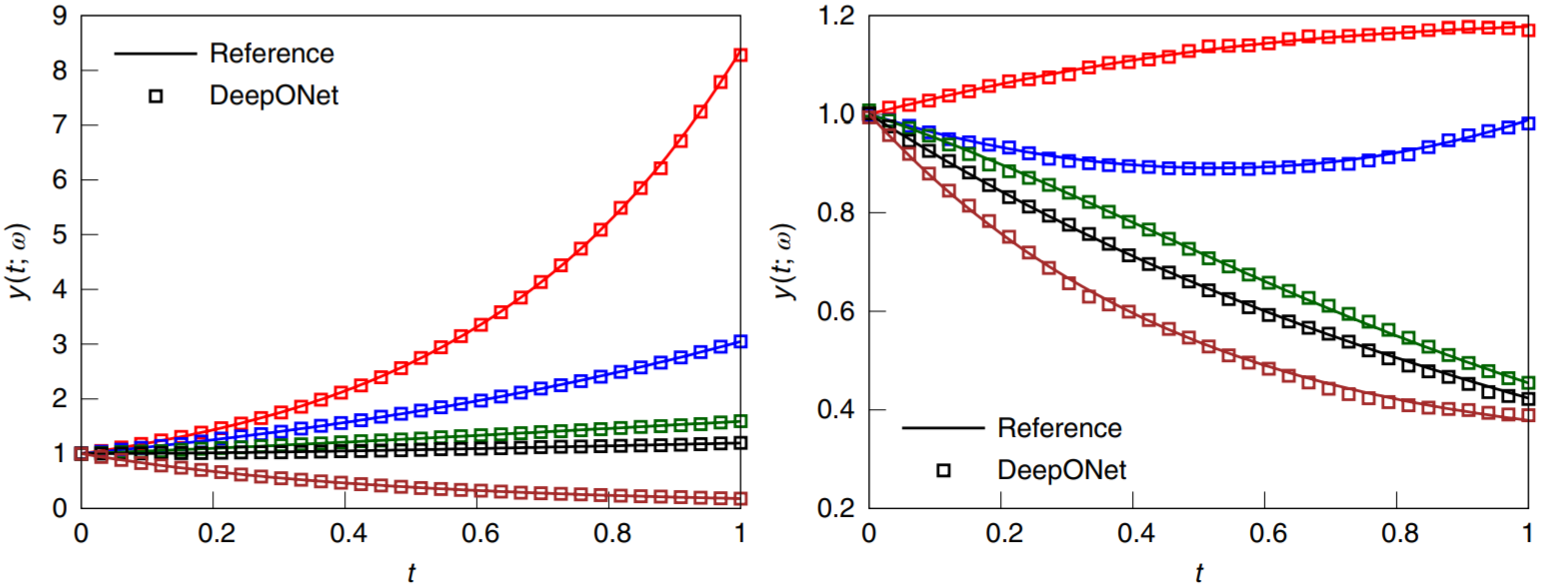


Stochastic operators: $dy(t;\omega) = k(t;\omega)y(t;\omega)dt, \quad t \in (0, 1] \text{ and } \omega \in \Omega$ *population growth model

$k(t;\omega) \longrightarrow y(t;\omega).$

$k(t;\omega) \sim \mathcal{GP}(k_0(t), \text{Cov}(t_1, t_2))$

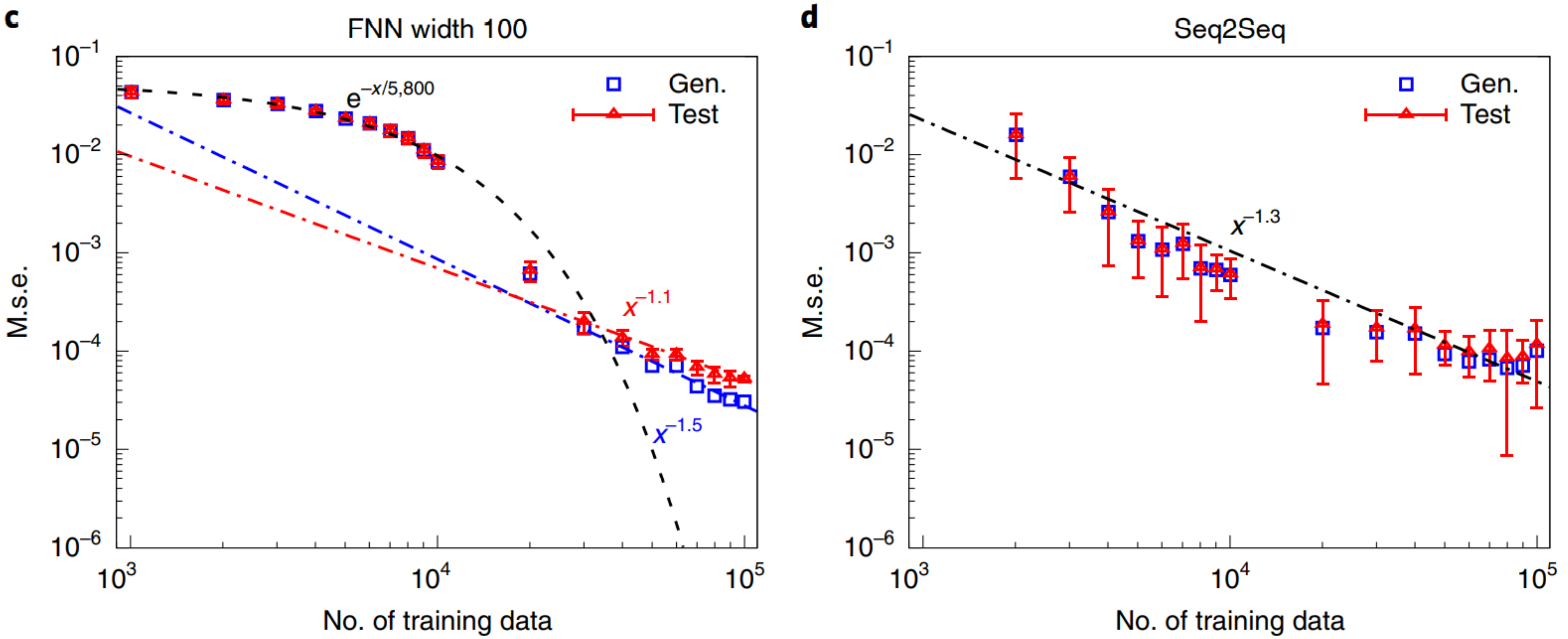
$\text{Cov}(t_1, t_2) = \sigma^2 \exp(-\| t_1 - t_2\|^2/2l^2).$



convergence rates

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

*the motion of a gravity pendulum with an external force



- have exponential convergence for small training datasets and then converge with polynomial rates
- The transition point depends on the width, and a bigger network has a later transition point

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

*nonlinear diffusion-reaction PDE