Pacific Northwest NATIONAL LABORATORY

Multifidelity, domain decomposition, and stacking for improving training for physics-informed networks

November 13, 2024

Amanda Howard

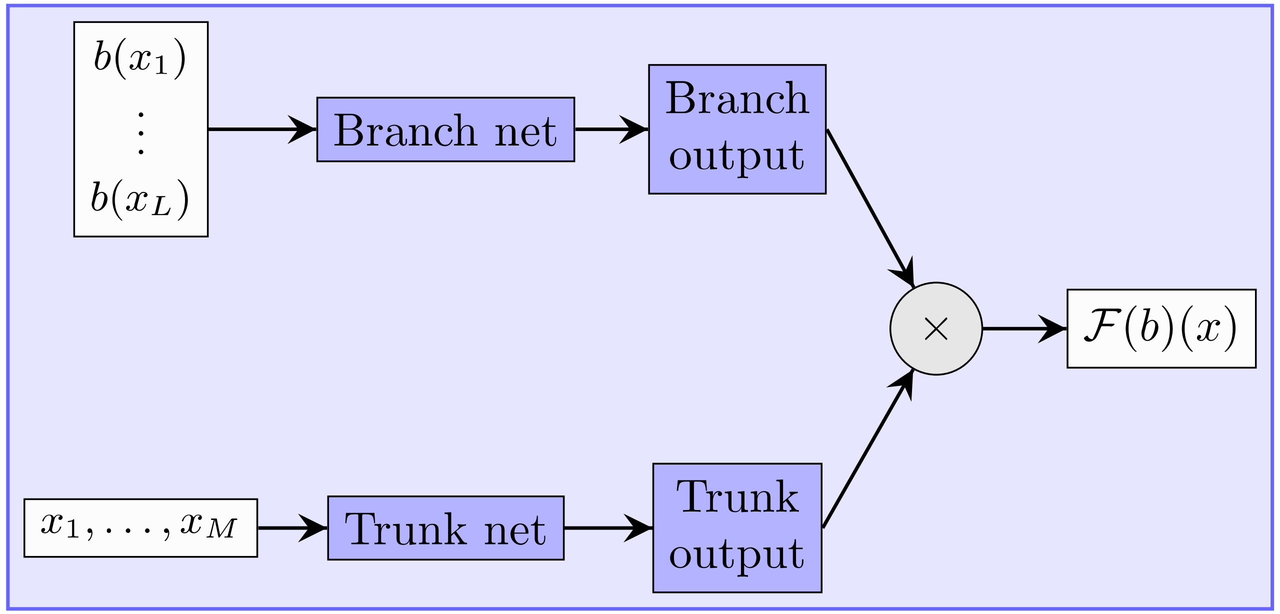
Mauro Perego, George Karniadakis, Panos Stinis, Alexander Heinlein, Sarah Murphy, Damien Beecroft, Saad Qadeer, Bruno Jacob

U.S. DEPARTMENT OF ENERGY BATTELLE

PNNL is operated by Battelle for the U.S. Department of Energy

Deep Operator Networks (DeepONets)

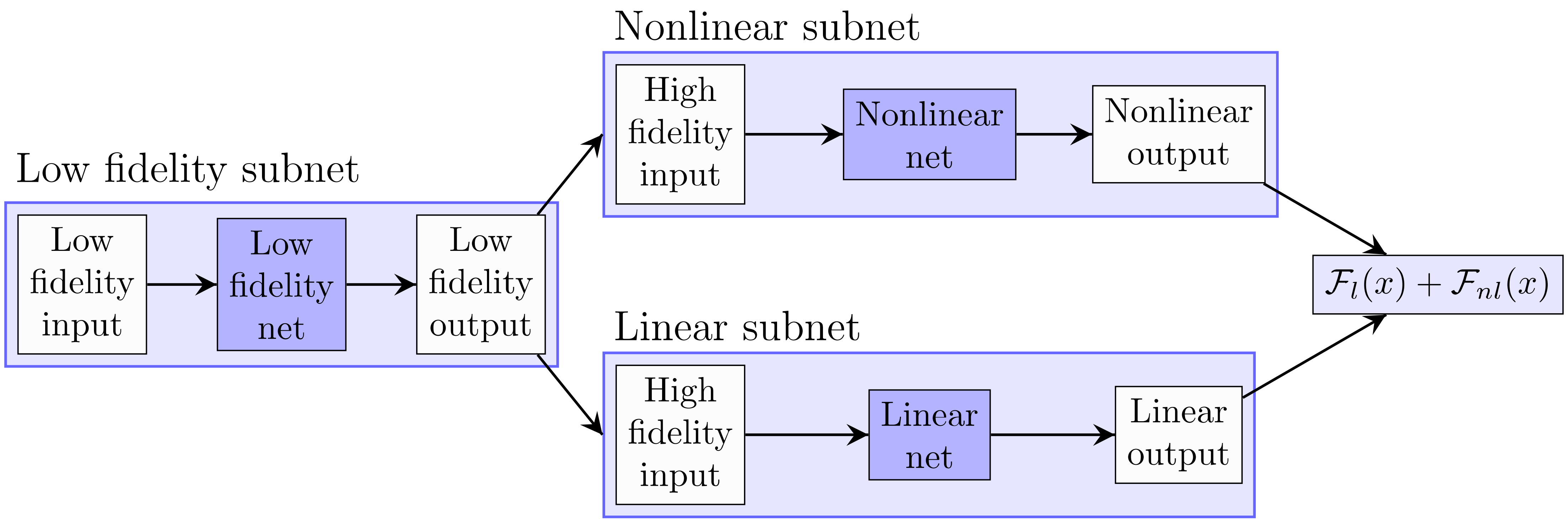
- PINNs map one input to one output
- PI-DeepONets map a *space* of inputs to a *space* of outputs
- PI-DeepONets are much more powerful, but therefore much more difficult to train.



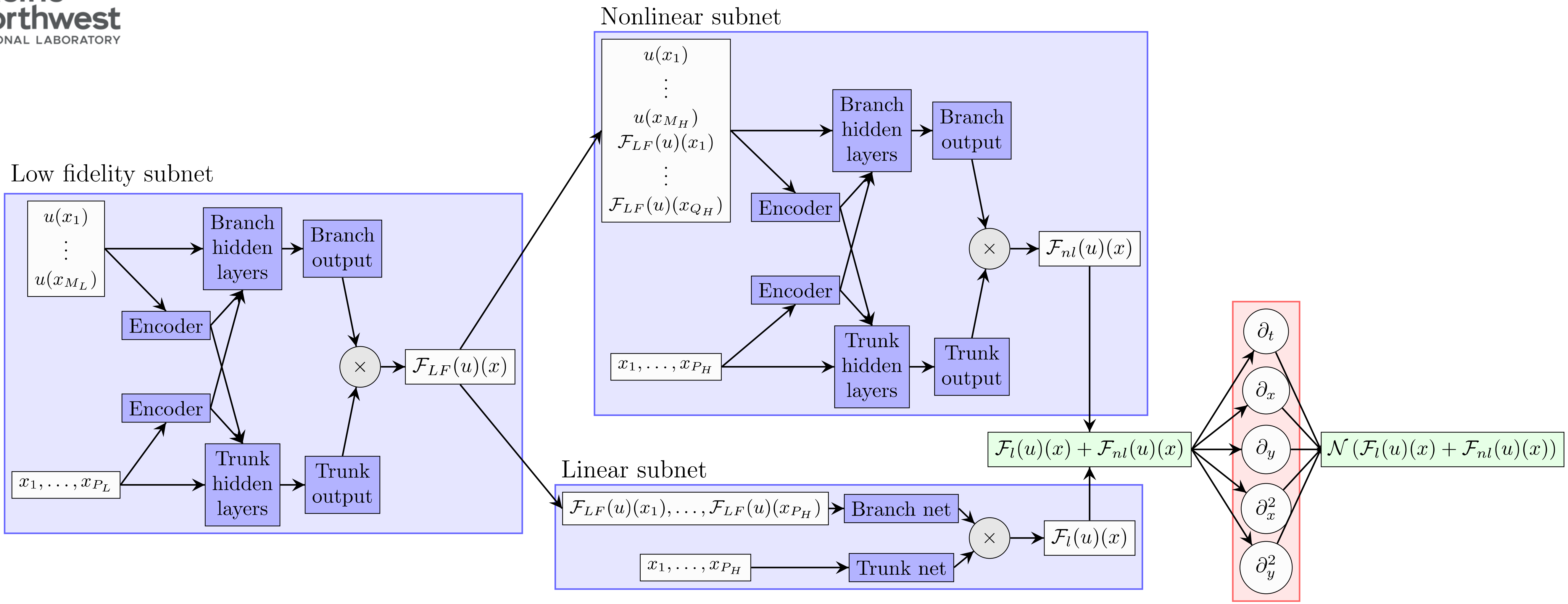
Lu, Lu, et al. "Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators." *Nature machine intelligence* 3.3 (2021): 218-229.
Wang, Sifan, Hanwen Wang, and Paris Perdikaris. "Learning the solution operator of parametric partial differential equations with physics-informed DeepONets." *Science advances* 7.40 (2021): eabi8605.

Multifidelity

- Allows the use of one or more dataset and physics to enhance training
- Specific structure allows for exploiting correlations between the data



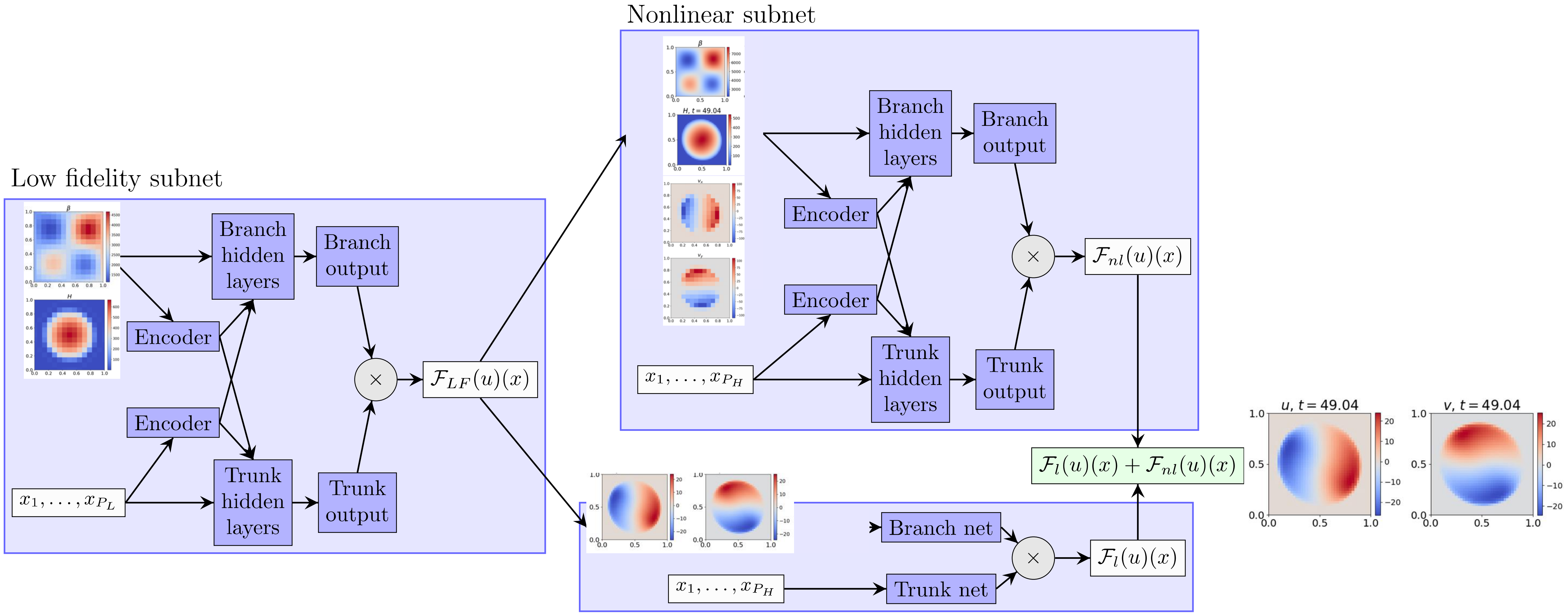
General framework



$$\mathcal{L}(\theta) = \lambda_1 \mathcal{L}_{HF}(\theta_{nl}, \theta_l) + \lambda_2 \mathcal{L}_{LF}(\theta_{LF}) + \lambda_3 \left(\sum w_{nl}^2 + \sum b_{nl}^2 \right) + \lambda_4 \left(\sum w_{LF}^2 + \sum b_{LF}^2 \right)$$
$$+ \lambda_5 \mathcal{L}_{IC}(\theta_{nl}, \theta_l) + \lambda_6\, \mathcal{L}_{BC}(\theta_{nl}, \theta_l) + \lambda_7 \mathcal{L}_{physics}(\theta_{nl}, \theta_l)$$

AAH, et al. "Multifidelity deep operator networks." *Journal of Computational Physics* (2023).

Pacific Northwest
NATIONAL LABORATORY



Pacific Northwest
NATIONAL LABORATORY

One-dimensional jump function

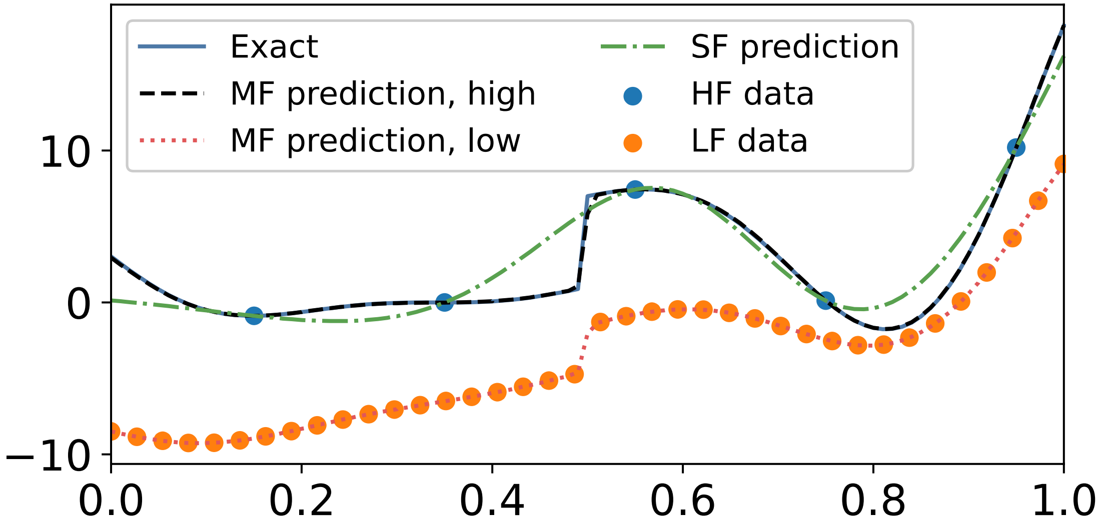
$$y_L(u)(x) = \begin{cases} 0.5(6x-2)^2\sin(u) + 10(x-0.5) - 5 & x \le 0.5 \\ 0.5(6x-2)^2\sin(u) + 10(x-0.5) - 2 & x > 0.5 \end{cases}$$

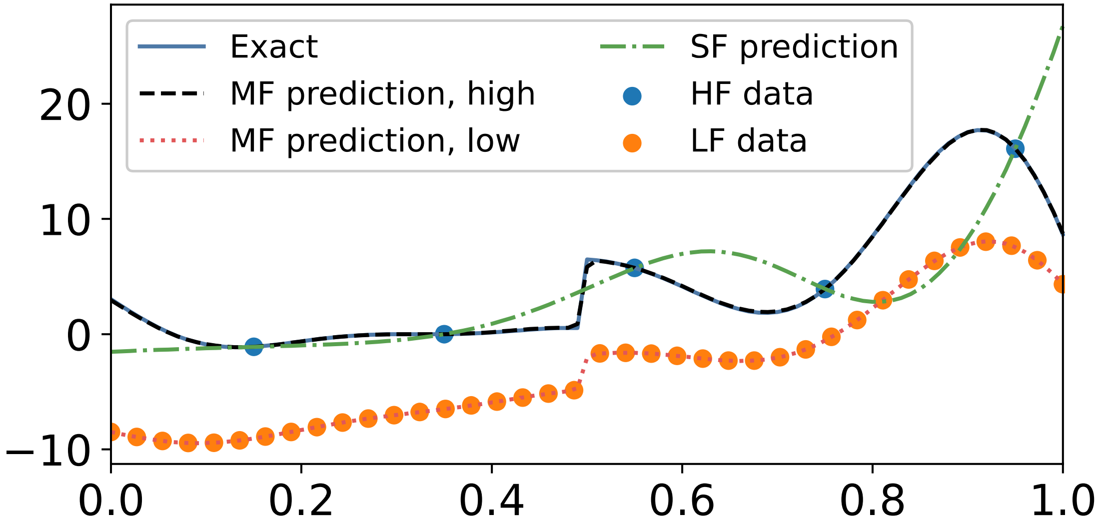
$$y_H(u)(x) = 2y_L(u)(x) - 20x + 20$$

$$u = ax - 4$$

$N_L = 20$
$M_L = \mathrm{P_L} = 38$
$N_H = 10$
$M_H = P_H = 5$

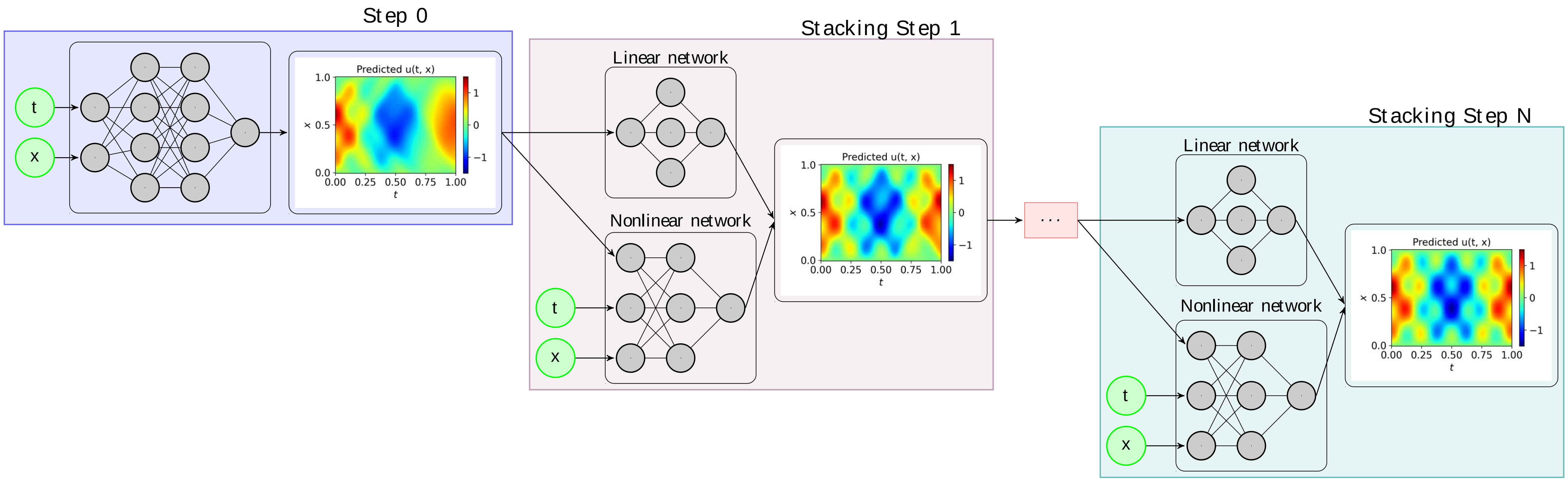
$$\mathcal{F}_l(u)(x) = 1.9479\mathcal{F}_{LF}(u)(x) - 19.1719x + 19.3459 - 0.04870x\mathcal{F}_{LF}(u)(x)$$





Stacking multifidelity networks improves predictions

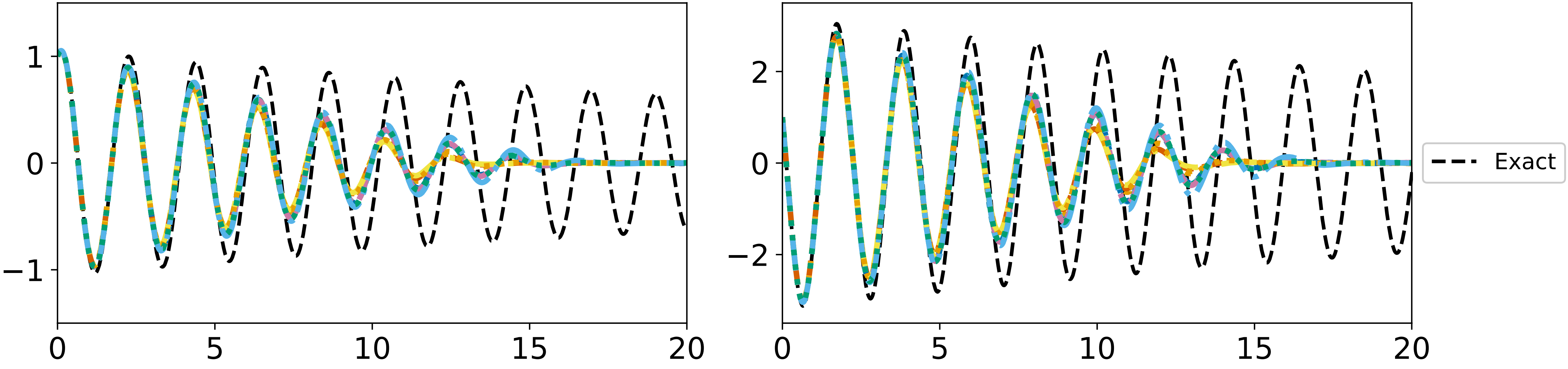
- Train a vanilla PINN/DeepONet
- Continually train MF networks until the desired accuracy is reached



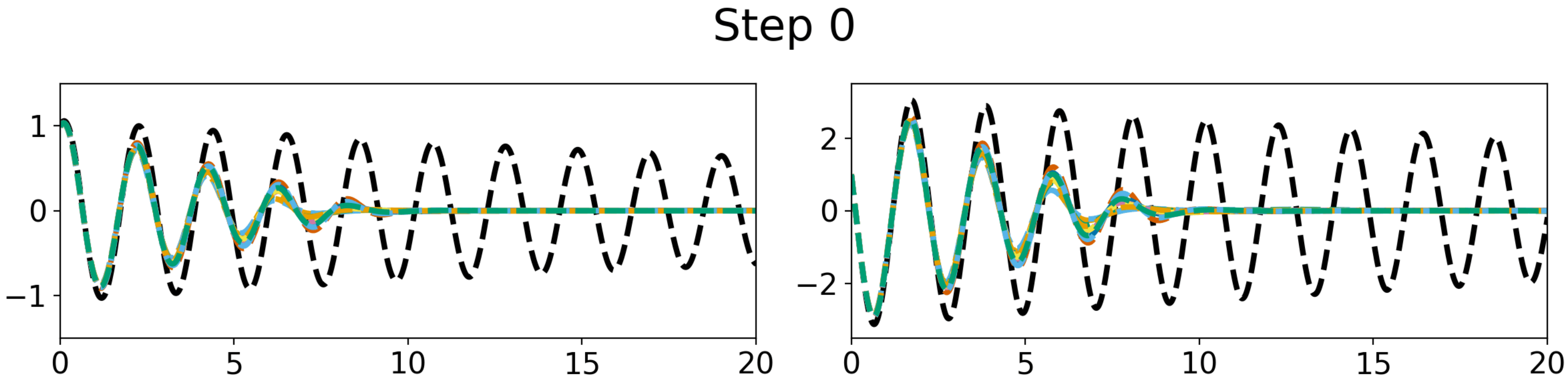
Pendulum

$$\frac{ds_1}{dt} = s_2$$
$$\frac{ds_s}{dt} = -\frac{b}{m}s_2 - \frac{g}{L}\sin(s_1)$$

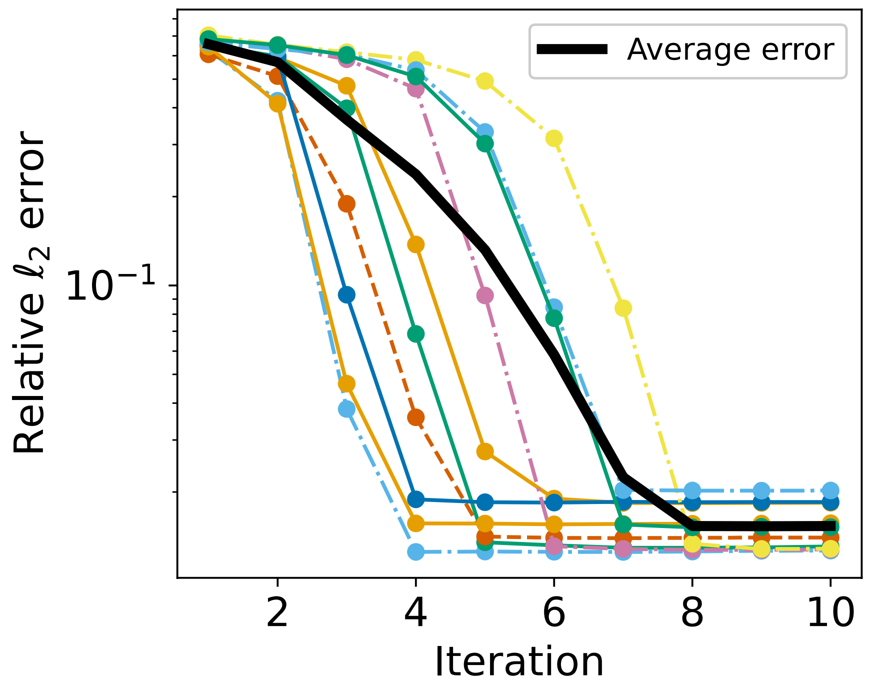
$$s_1(0) = 1$$
$$s_2(0) = 1$$



Pendulum



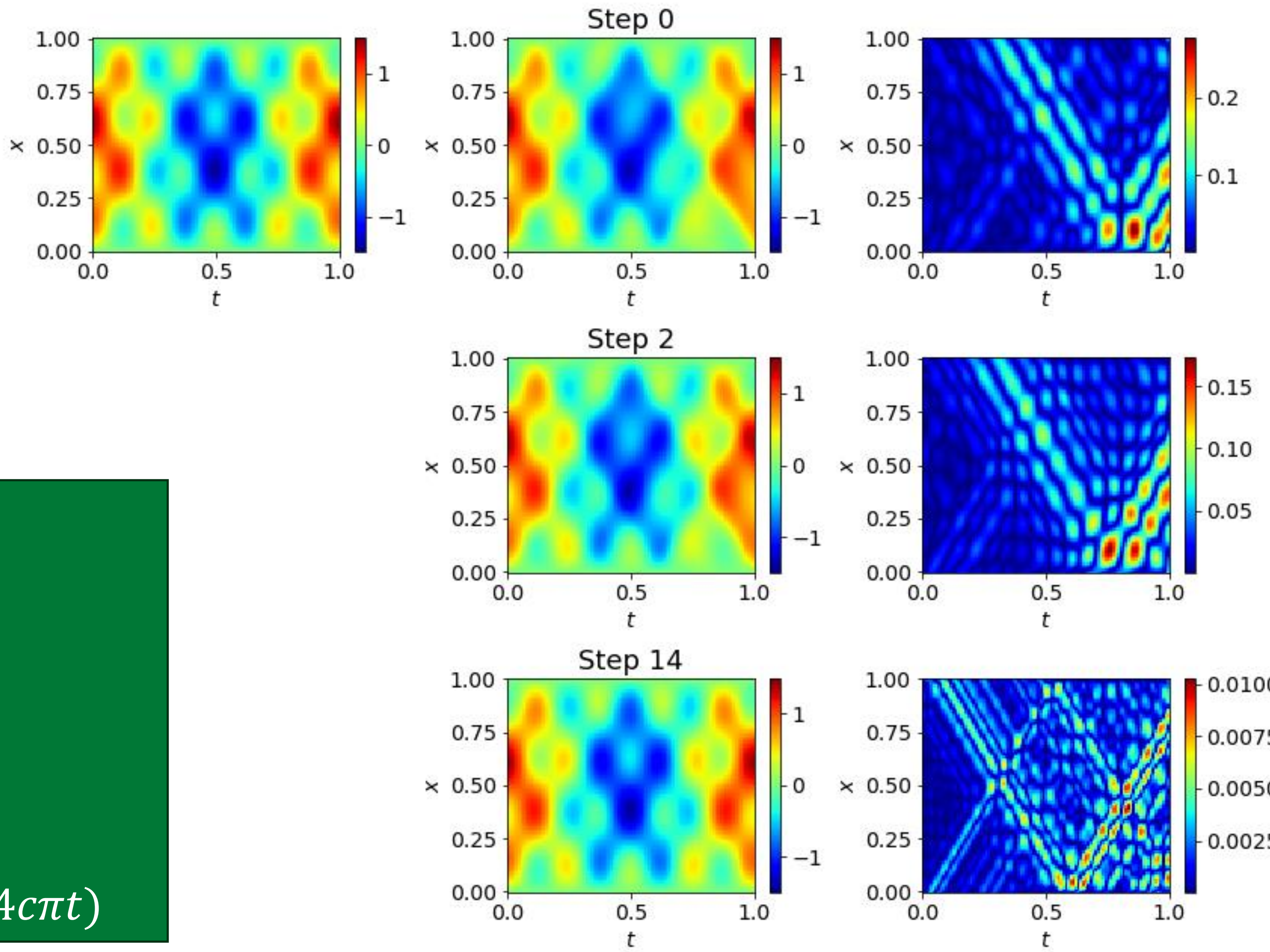
Step 0
1
0
−1
0
5
10
15
20
2
0
−2
0
5
10
15
20



Average error
Relative ℓ_2 error
10^{-1}
2
4
6
8
10
Iteration

Wave equation

$c = 2$



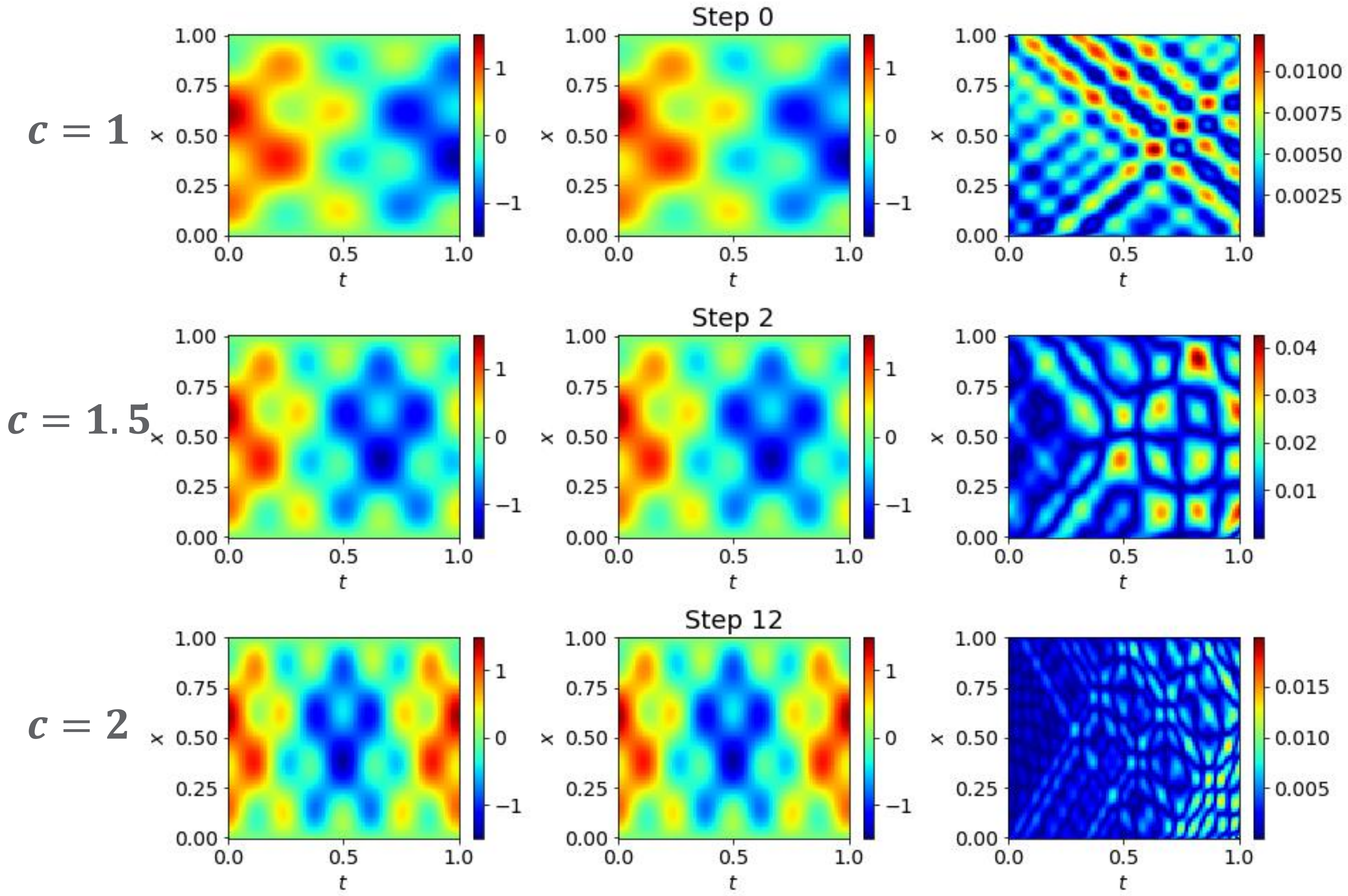
$$s_{tt}(x,t) - c^2 s_{xx}(x,t) = 0$$

$$s(x,0) = \sin(\pi x) + 0.5\sin(4\pi x)$$
$$s_t(x,0) = 0$$
$$s(0,t) = s(1,t) = 0$$

$$s(x,t) = \sin(\pi x)\cos(c\pi t) + 0.5\sin(4\pi x)\cos(4c\pi t)$$

Pacific Northwest
NATIONAL LABORATORY

Wave equation

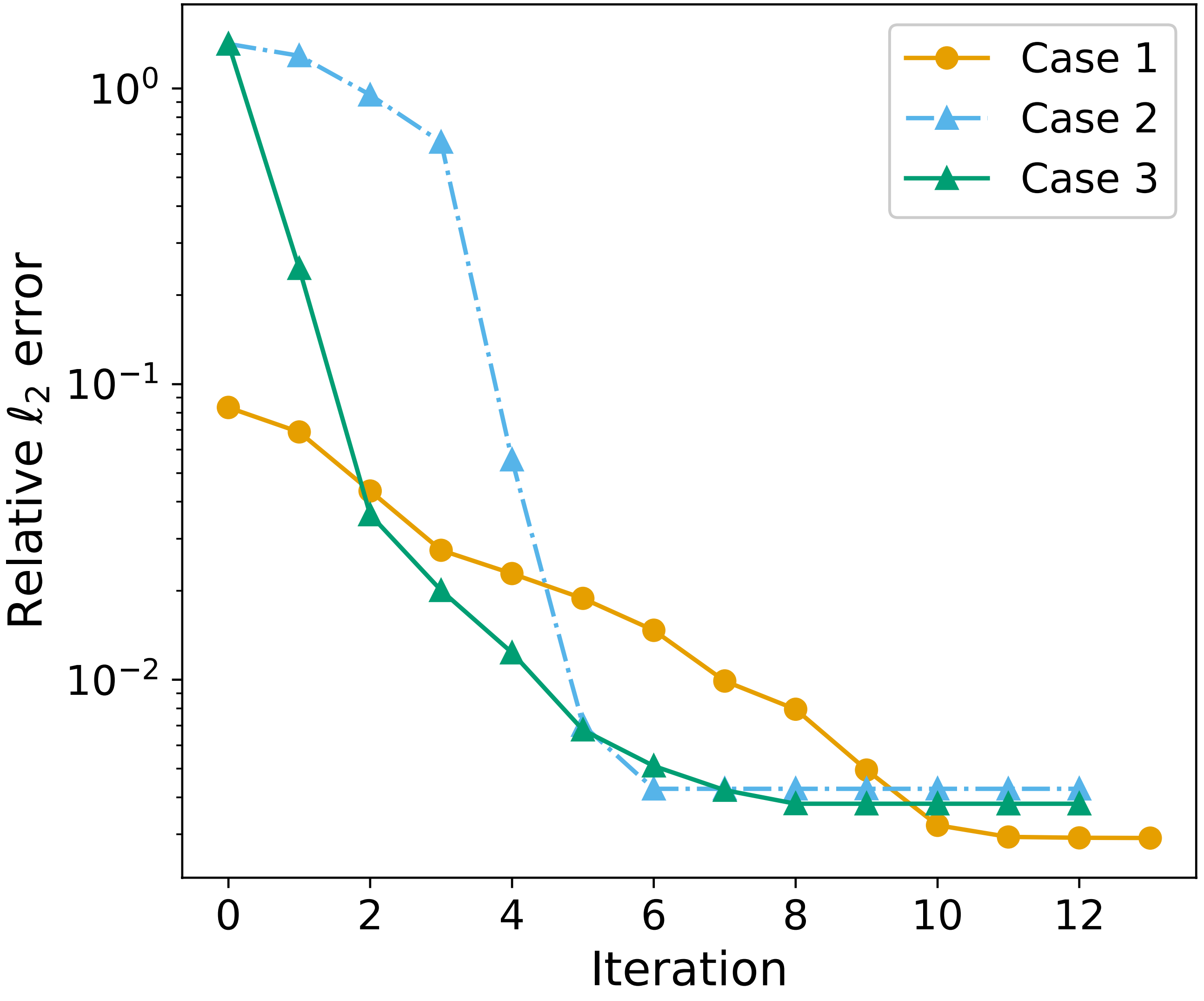


$$s_{tt}(x,t) - c^2 s_{xx}(x,t) = 0$$

$$s(x,0) = \sin(\pi x) + 0.5\sin(4\pi x)$$
$$s_t(x,0) = 0$$
$$s(0,t) = s(1,t) = 0$$

Wave equation

Case 1: $c = 2$
Case 2: $c = [1, 1.25, 1.5, 1.75, 2, 2 \ldots]$
Case 3: $c = [1, 2, 2 \ldots]$



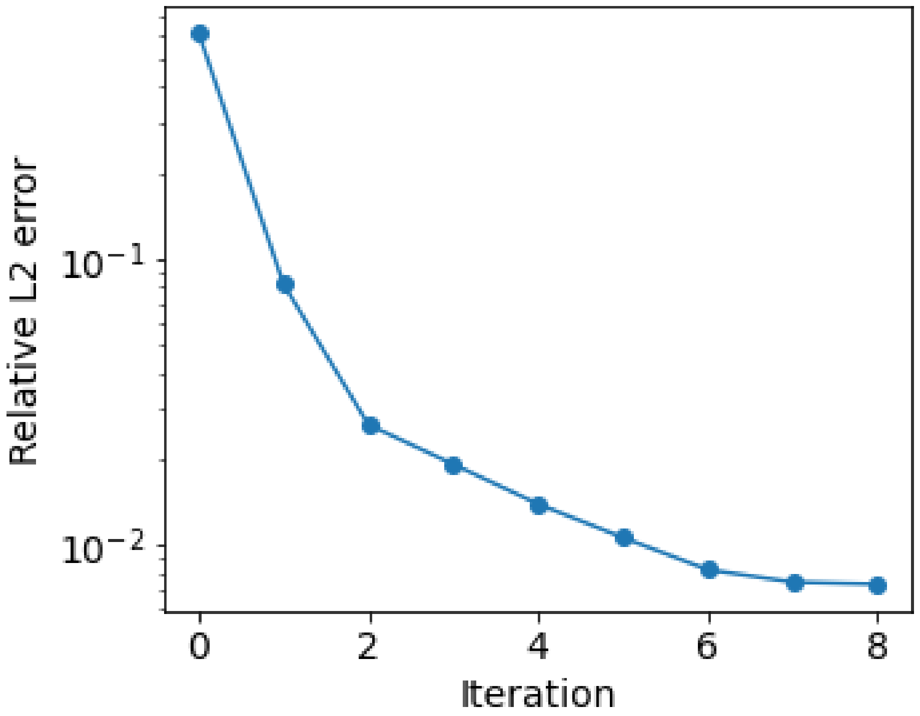
$$s_{tt}(x,t) - c^2 s_{xx}(x,t) = 0$$

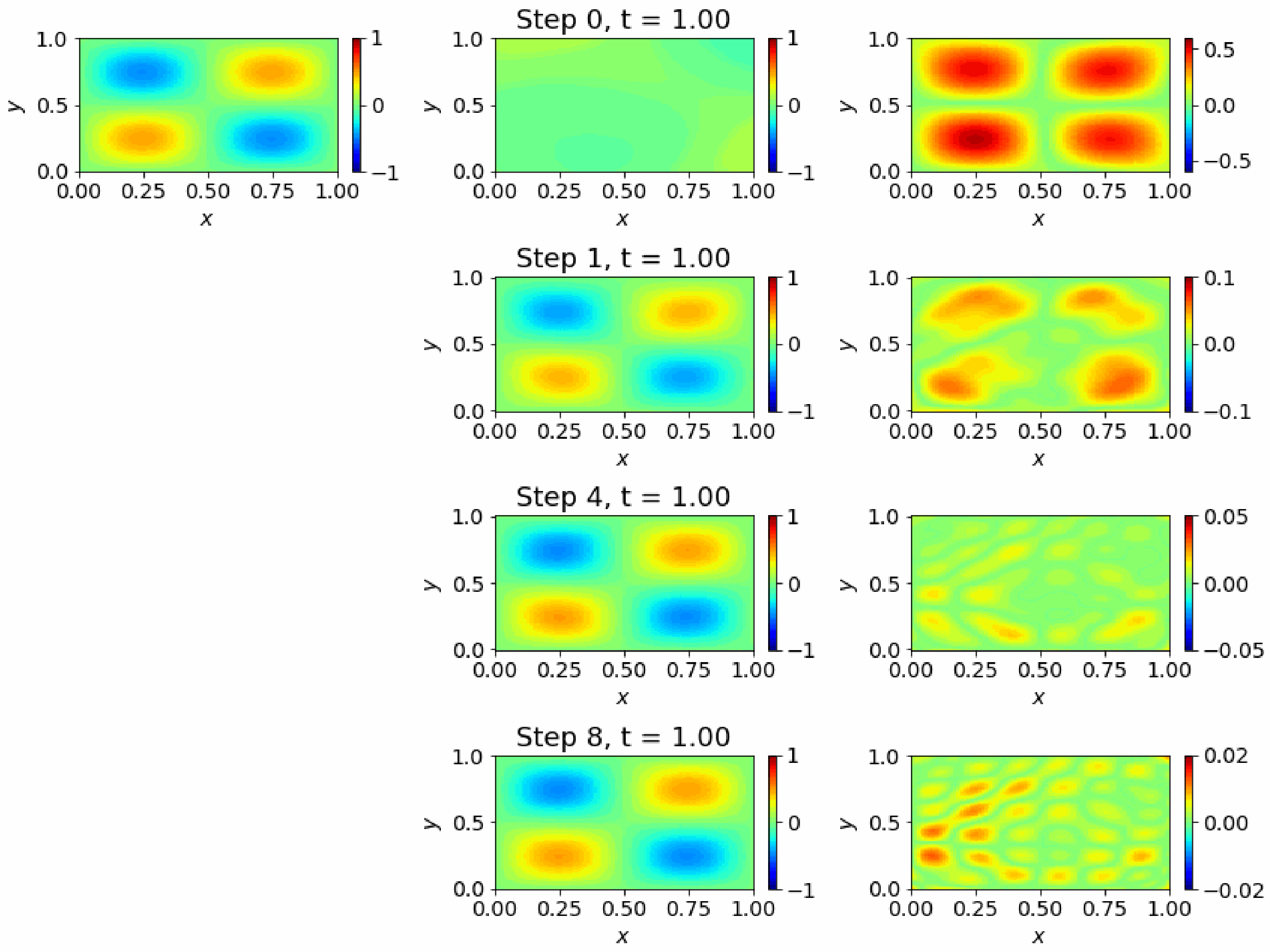
$$s(x,0) = \sin(\pi x) + 0.5\sin(4\pi x)$$
$$s_t(x,0) = 0$$
$$s(0,t) = s(1,t) = 0$$

Wave equation, 2d

$$s_{tt}(x,y,t) - c^2 s_{xx}(x,t) - c^2 s_{yy}(x,y,t) = 0$$

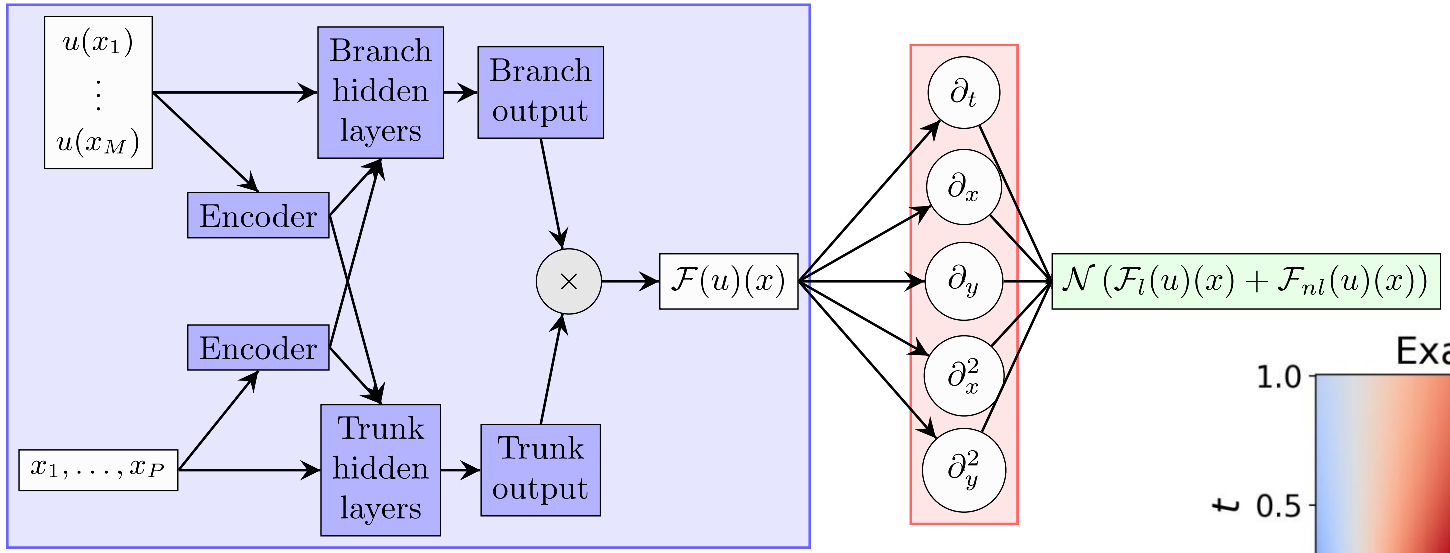
$$s(x,y,0) = \sin(\pi x)\sin(\pi y)$$
$$s_t(x,y,0) = 0$$
$$s(0,y,t) = s(1,y,t) = 0$$
$$s(x,0,t) = s(x,1,t) = 0$$

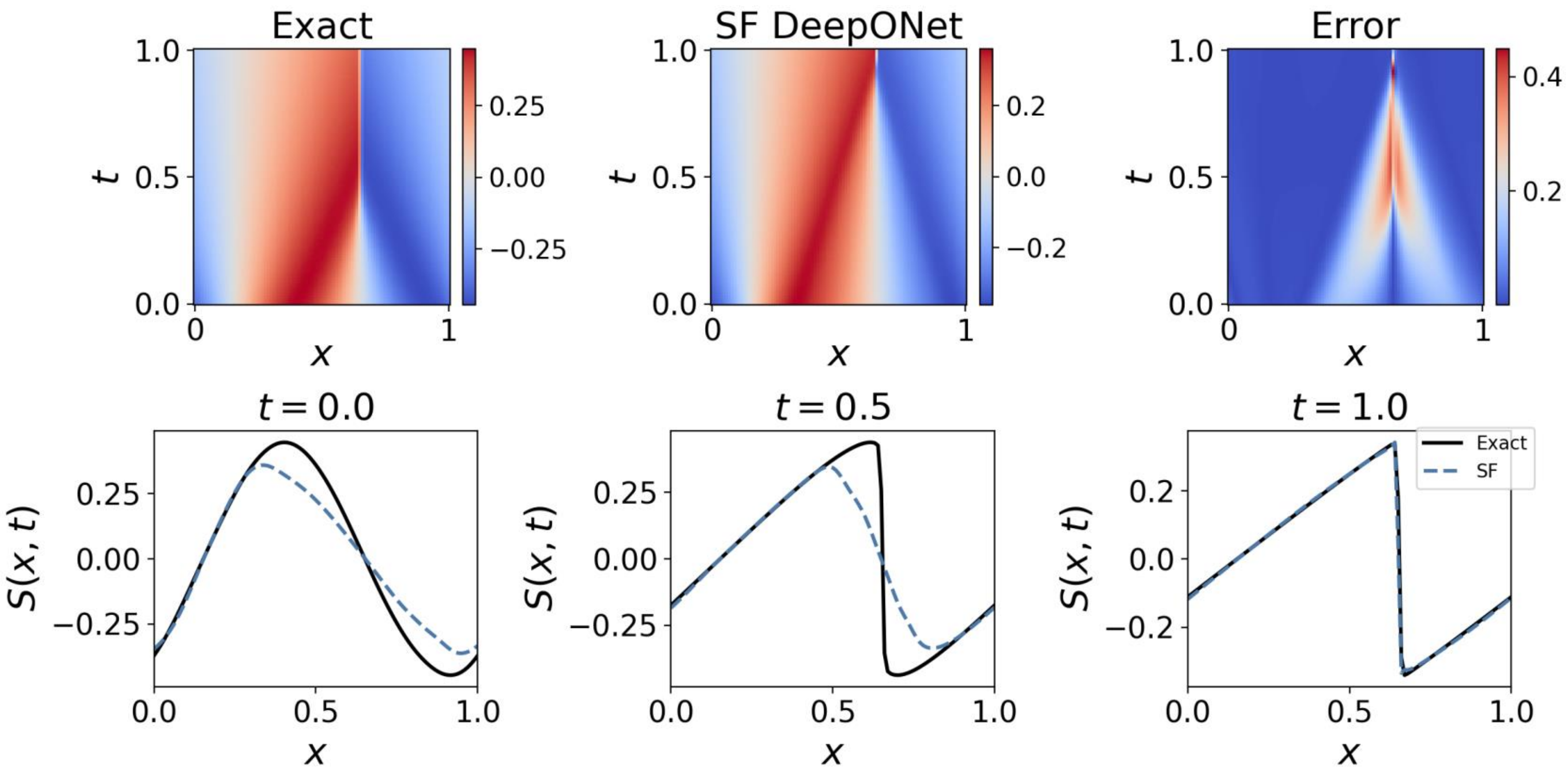




Stacking DeepONets

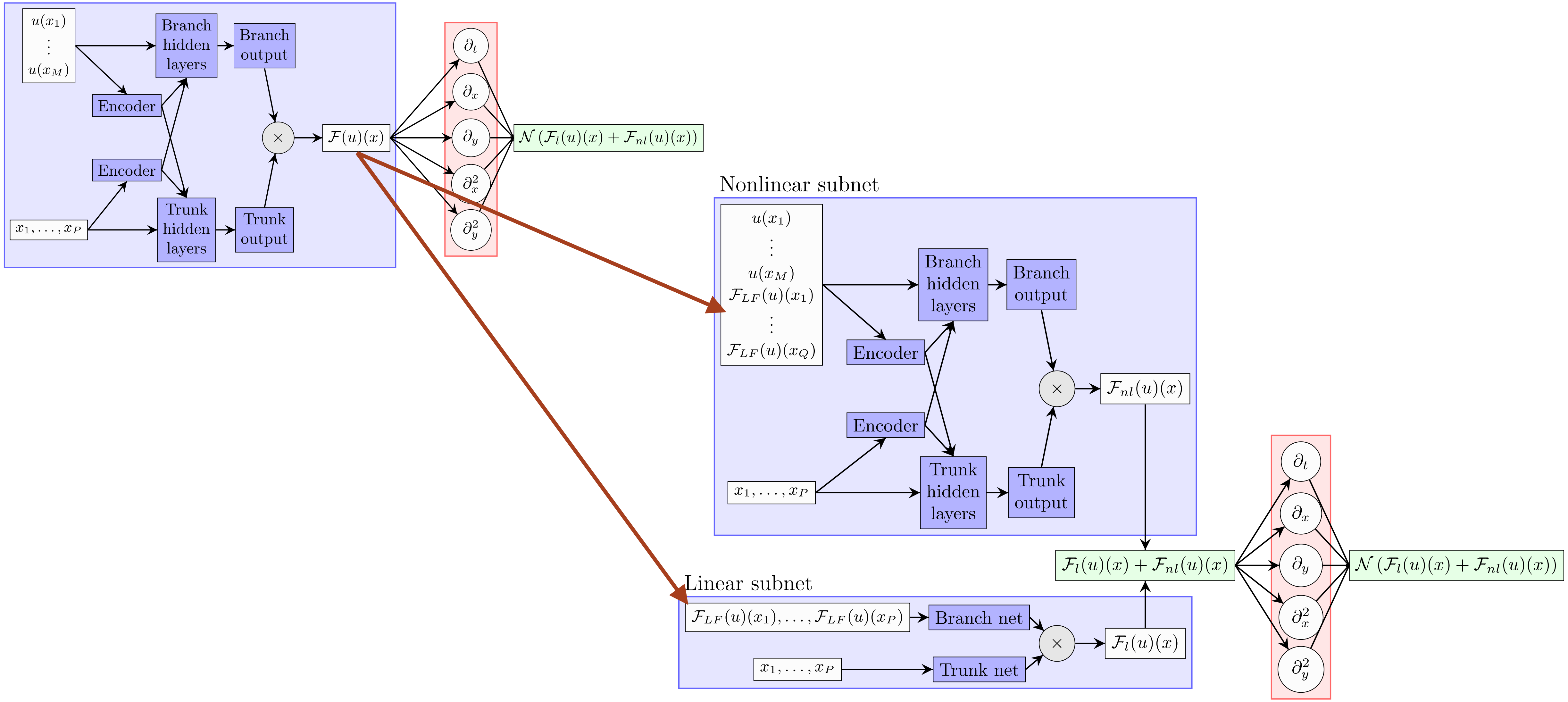
1) Train a single fidelity physics-informed DeepONet





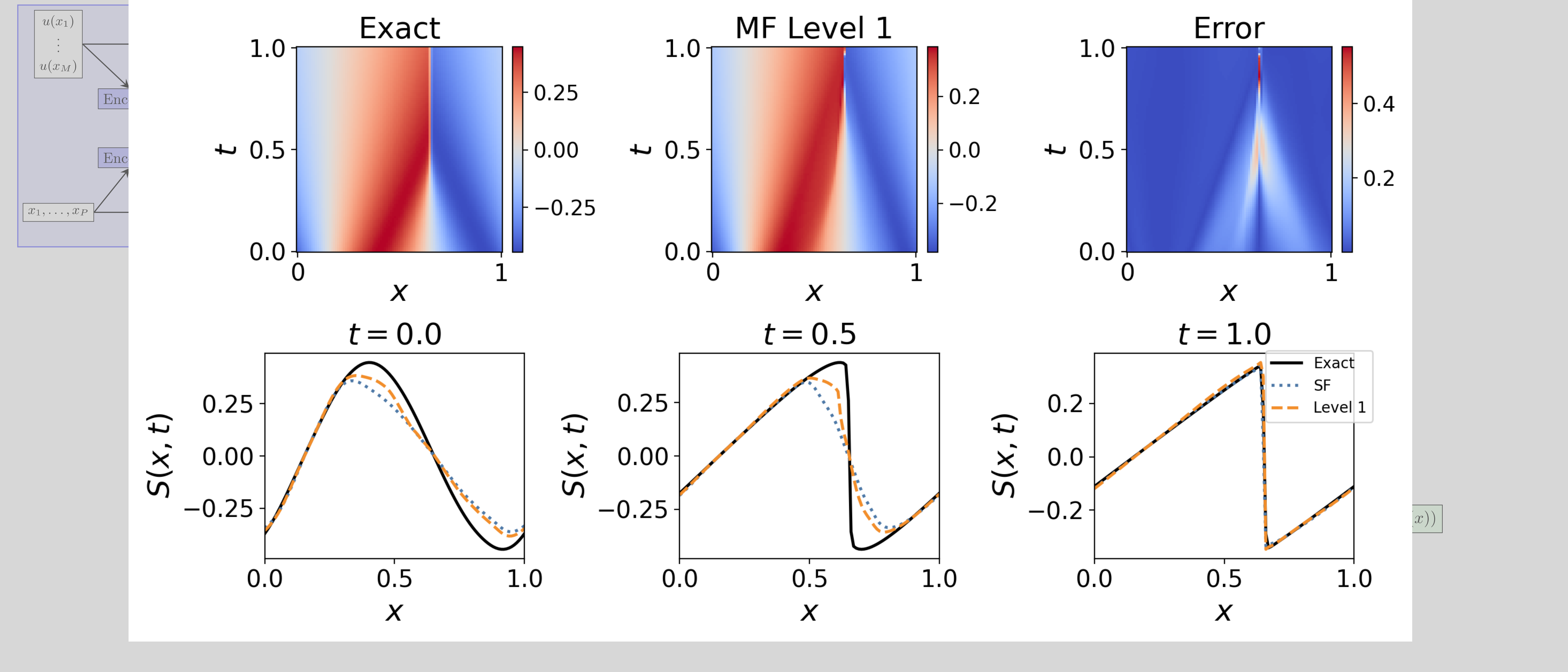
Stacking DeepONets

2) Train a non-composite multifidelity physics-informed DeepONet



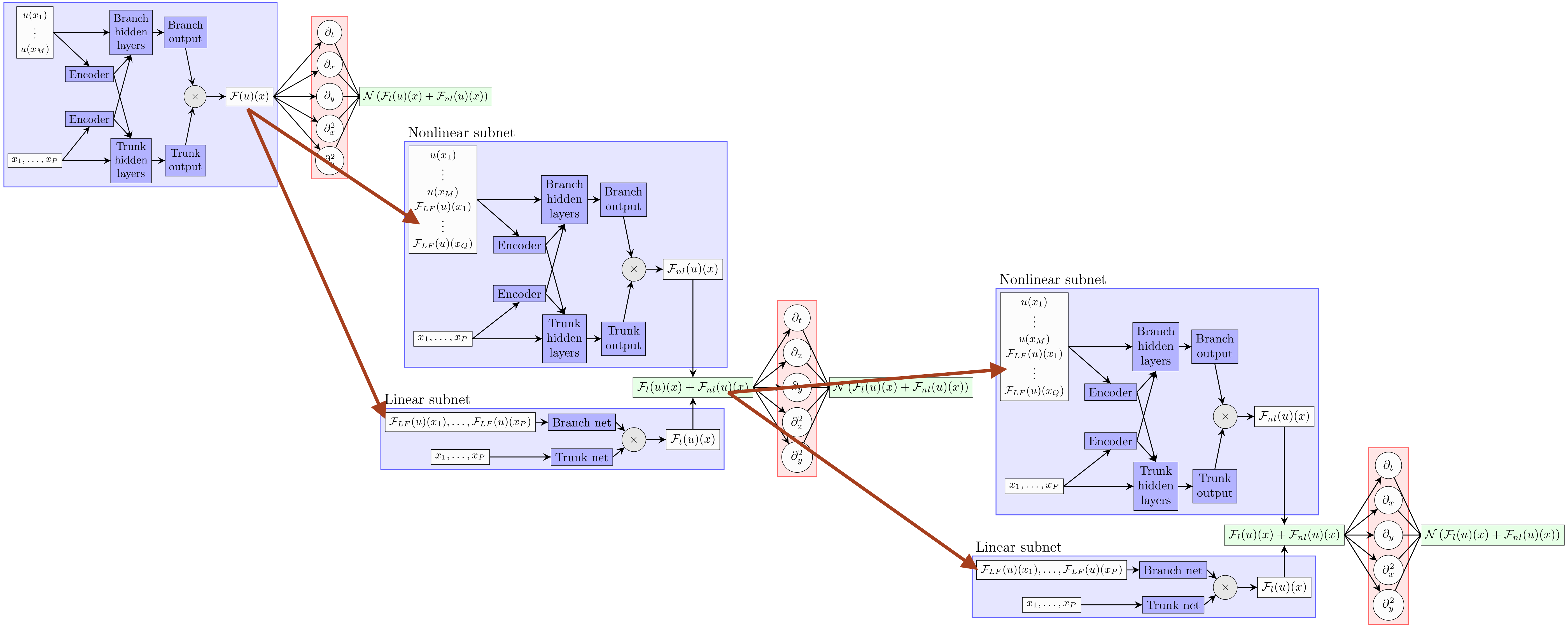
Stacking DeepONets

2) Train a non-composite multifidelity physics-informed DeepONet



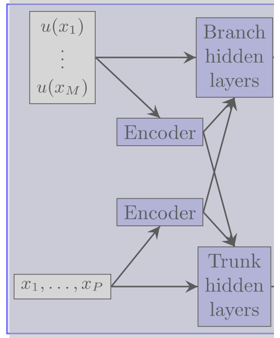
Stacking DeepONets

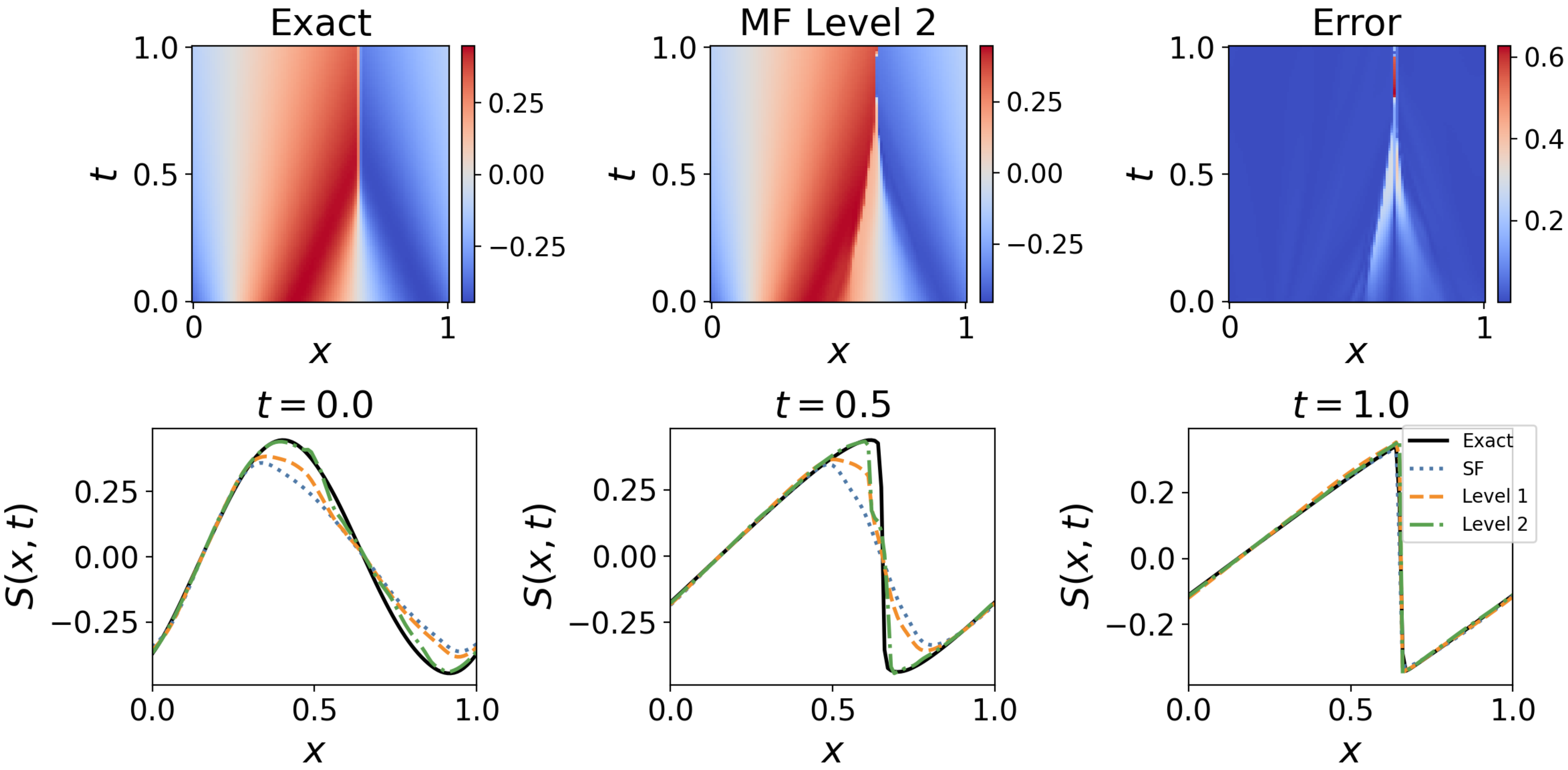
3) Train another non-composite multifidelity physics-informed DeepONet



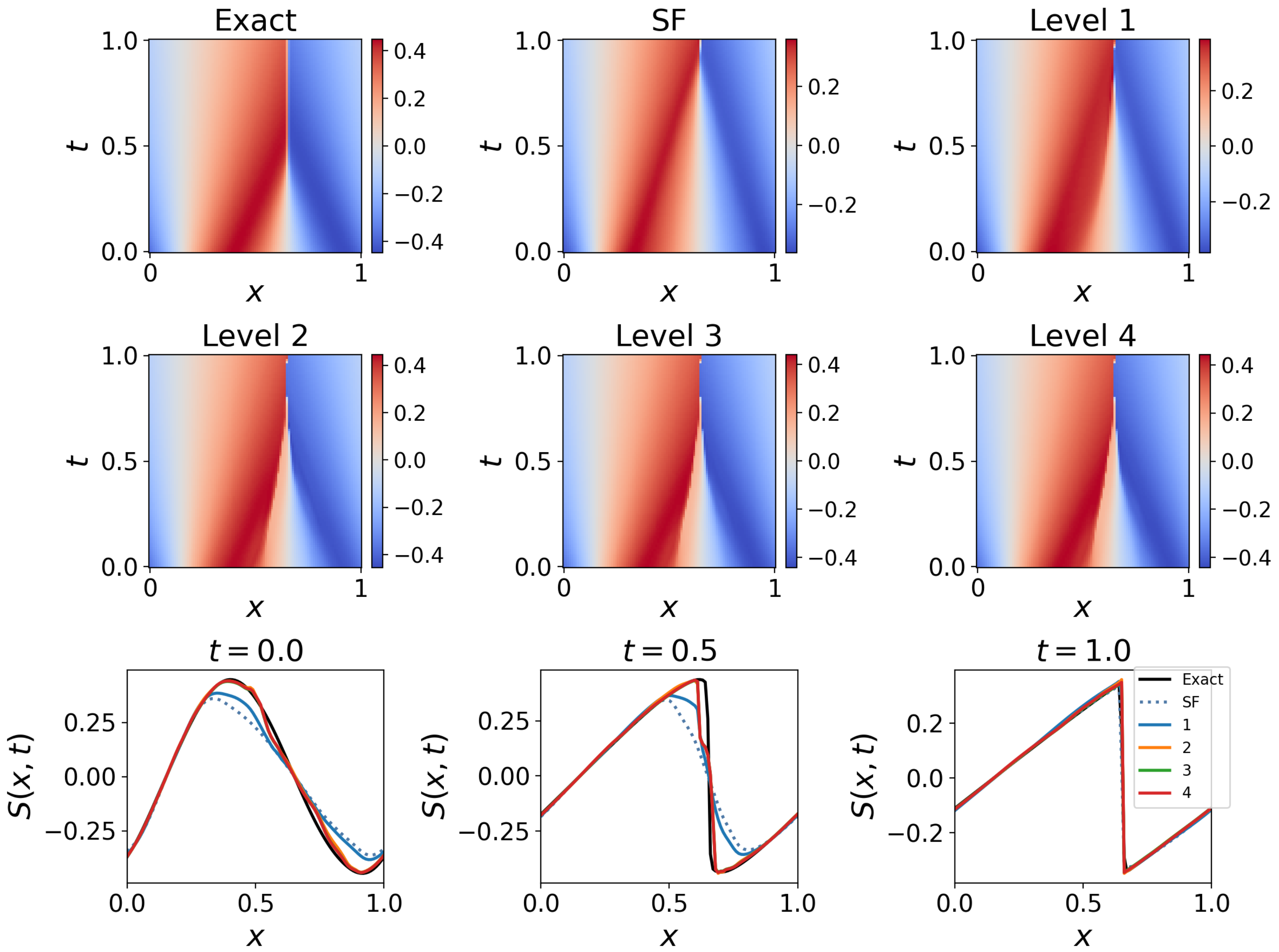
Stacking DeepONets

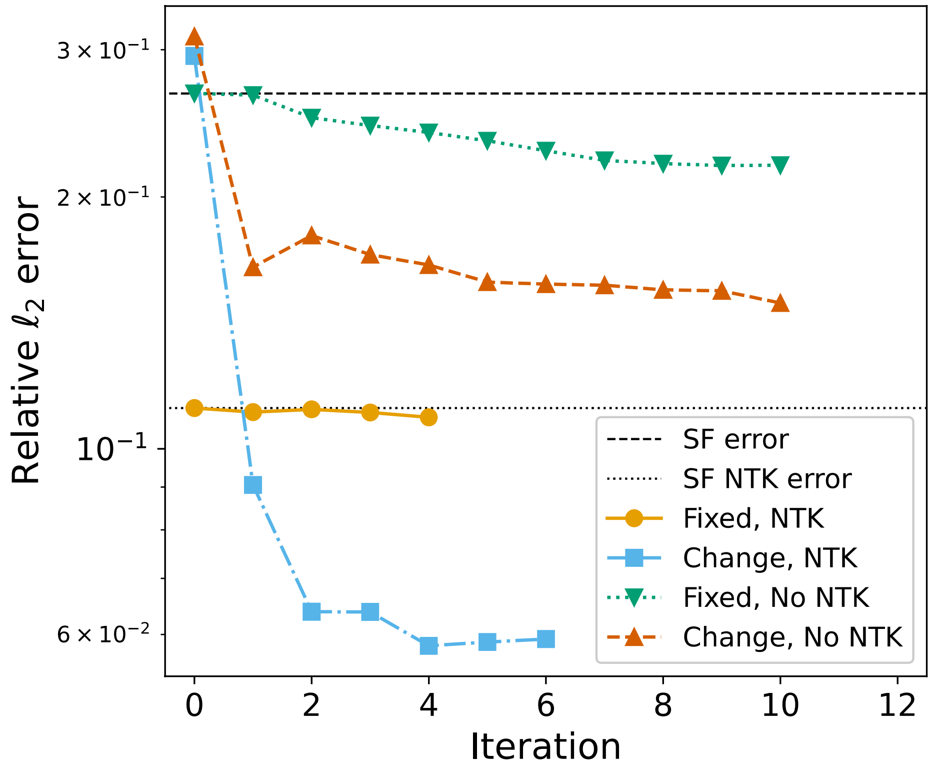
3) Train another non-composite multifidelity physics-informed DeepONet



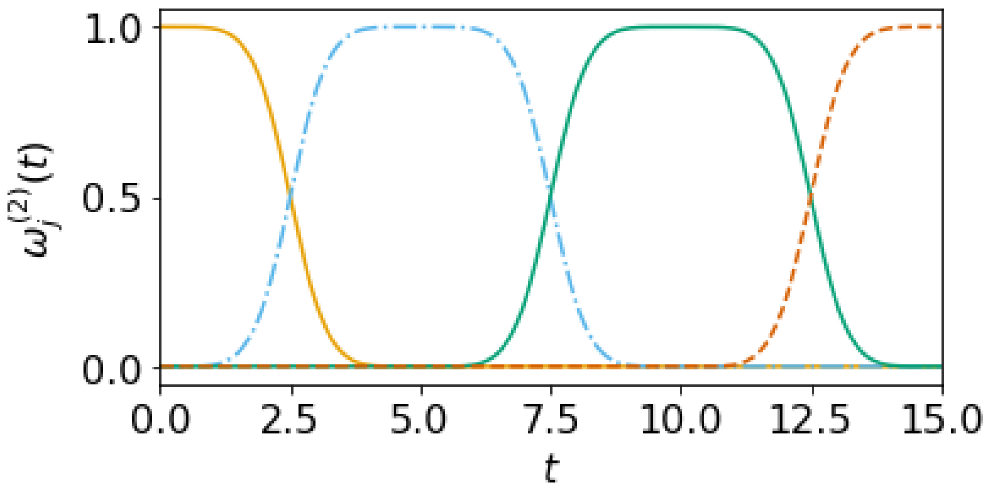


Stacking DeepONet





Finite basis PINNs



- Finite basis PINNs elegantly allow for domain decomposition through a weighted sum using partition of unity functions

$$u^{(l)}(x,\theta^{(l)}) = \sum_{j=1}^{J^{(l)}} \omega_j^{(l)} u_{j,MF}^{(l)}(x,\theta_j^{(l)},u^{(l-1)})$$

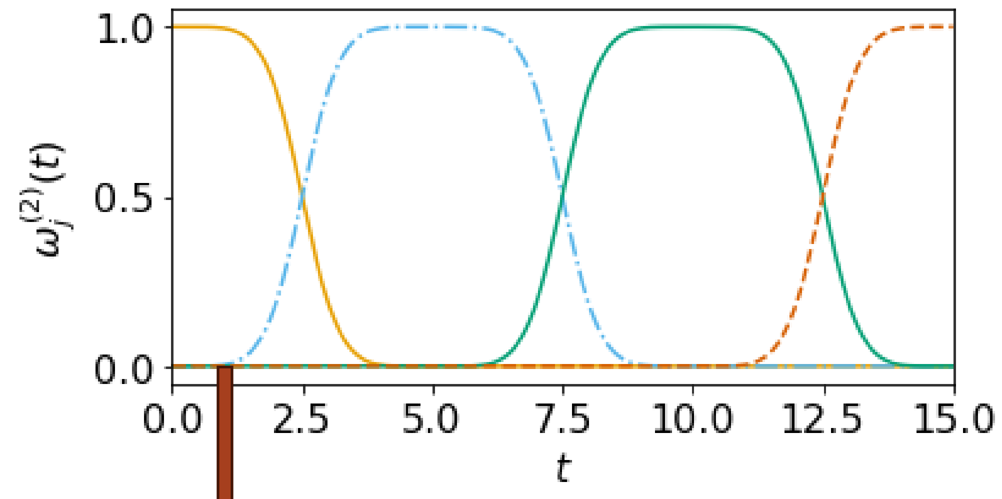
$$u_{j,MF}^{(l)}\left(x,\theta_j^{(l)}\right) = (1-|\alpha|)\,u_{j,linear}^{(l)}\left(x,\theta_j^{(l)}\right) + |\alpha|\,u_{j,nonlinear}^{(l)}\left(x,\theta_j^{(l)}\right)$$

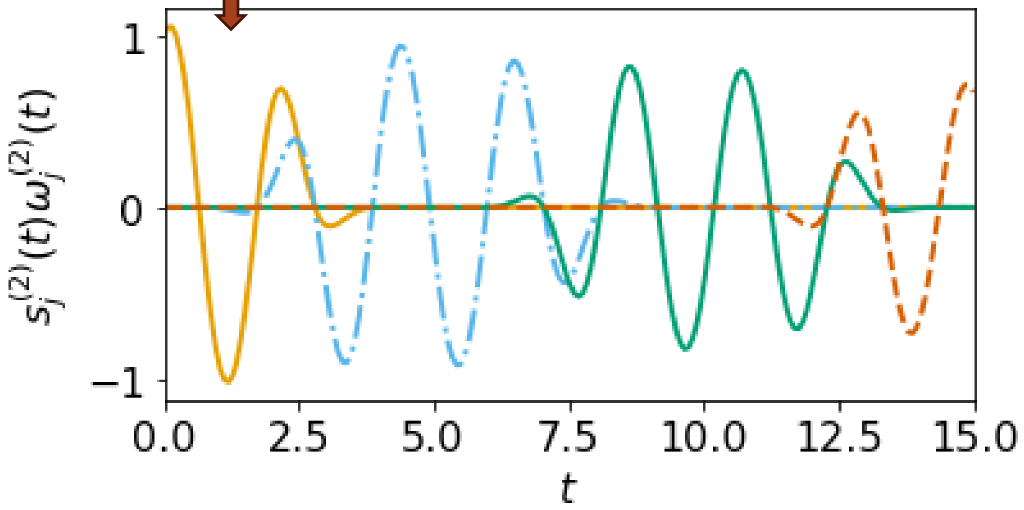
Finite basis PINNs

- Finite basis PINNs elegantly allow for domain decomposition through a weighted sum using partition of unity functions

$$u^{(l)}(x,\theta^{(l)}) = \sum_{j=1}^{J^{(l)}} \omega_j^{(l)} u_{j,MF}^{(l)}\left(x,\theta_j^{(l)},u^{(l-1)}\right)$$

$$u_{j,MF}^{(l)}\left(x,\theta_j^{(l)}\right) = (1-|\alpha|)\, u_{j,linear}^{(l)}\left(x,\theta_j^{(l)}\right) + |\alpha|\, u_{j,nonlinear}^{(l)}\left(x,\theta_j^{(l)}\right)$$



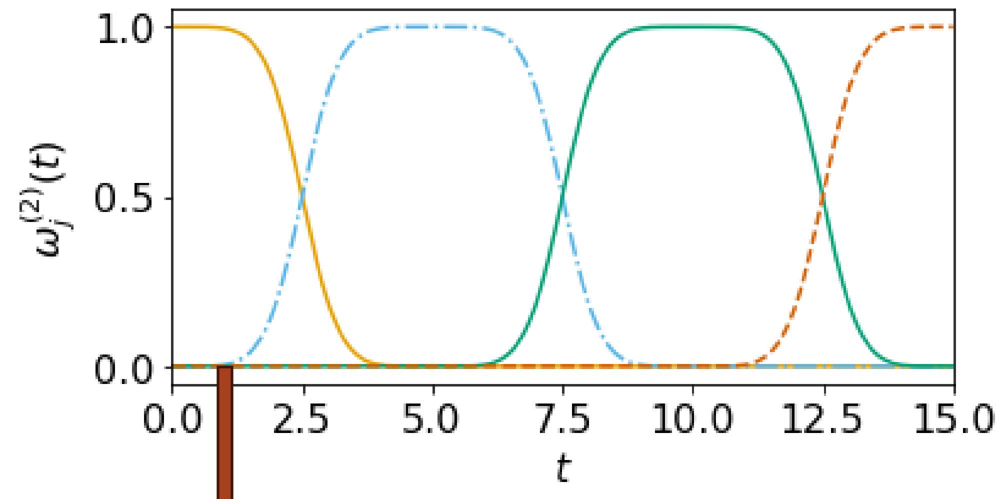


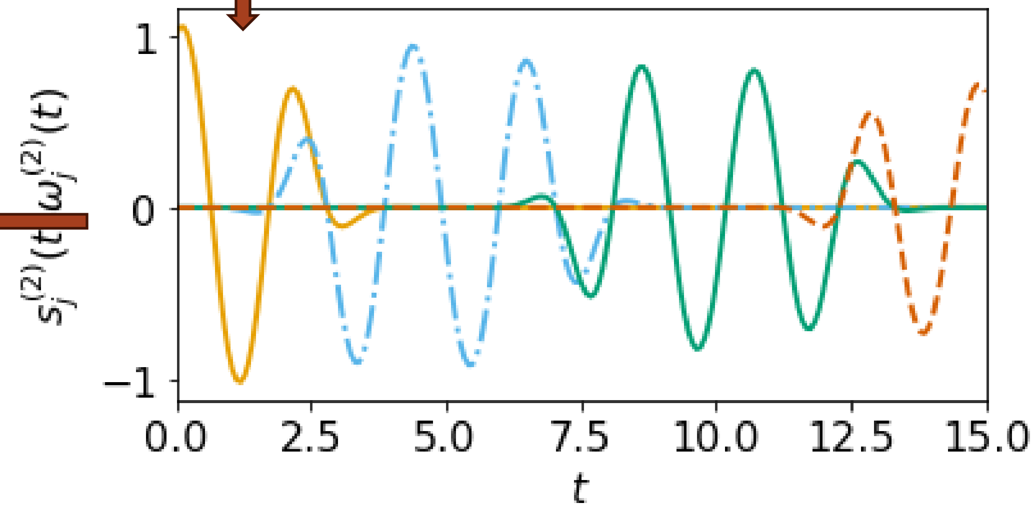
Finite basis PINNs

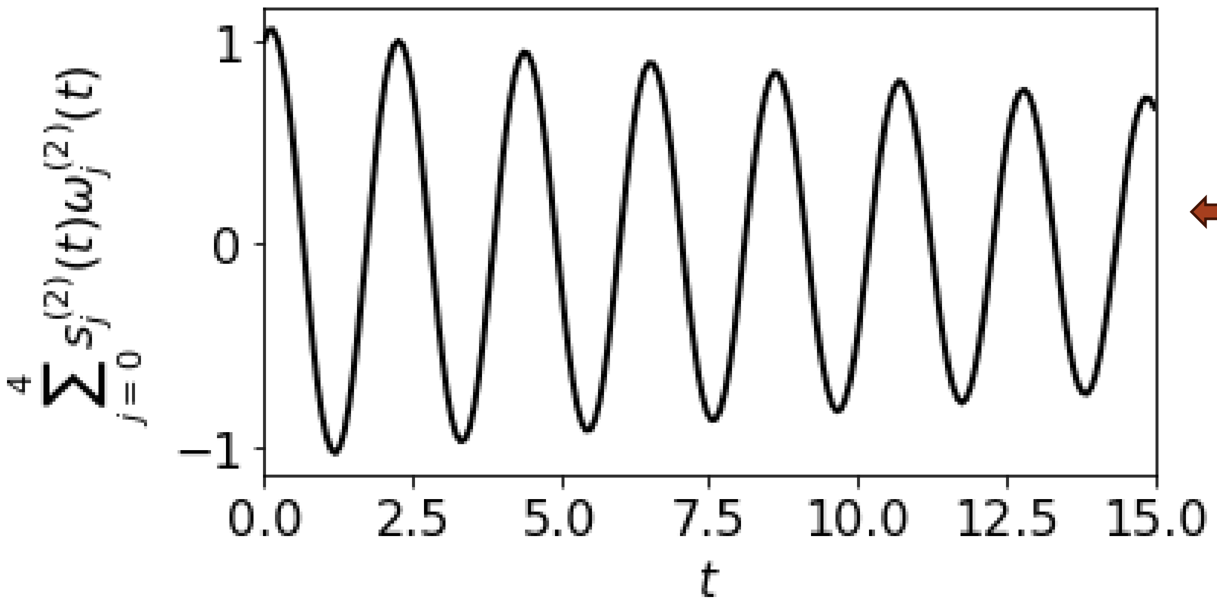
- Finite basis PINNs elegantly allow for domain decomposition through a weighted sum using partition of unity functions

$$u^{(l)}\left(x,\theta^{(l)}\right)=\sum_{j=1}^{J^{(l)}}\omega_j^{(l)}u_{j,MF}^{(l)}\left(x,\theta_j^{(l)},u^{(l-1)}\right)$$

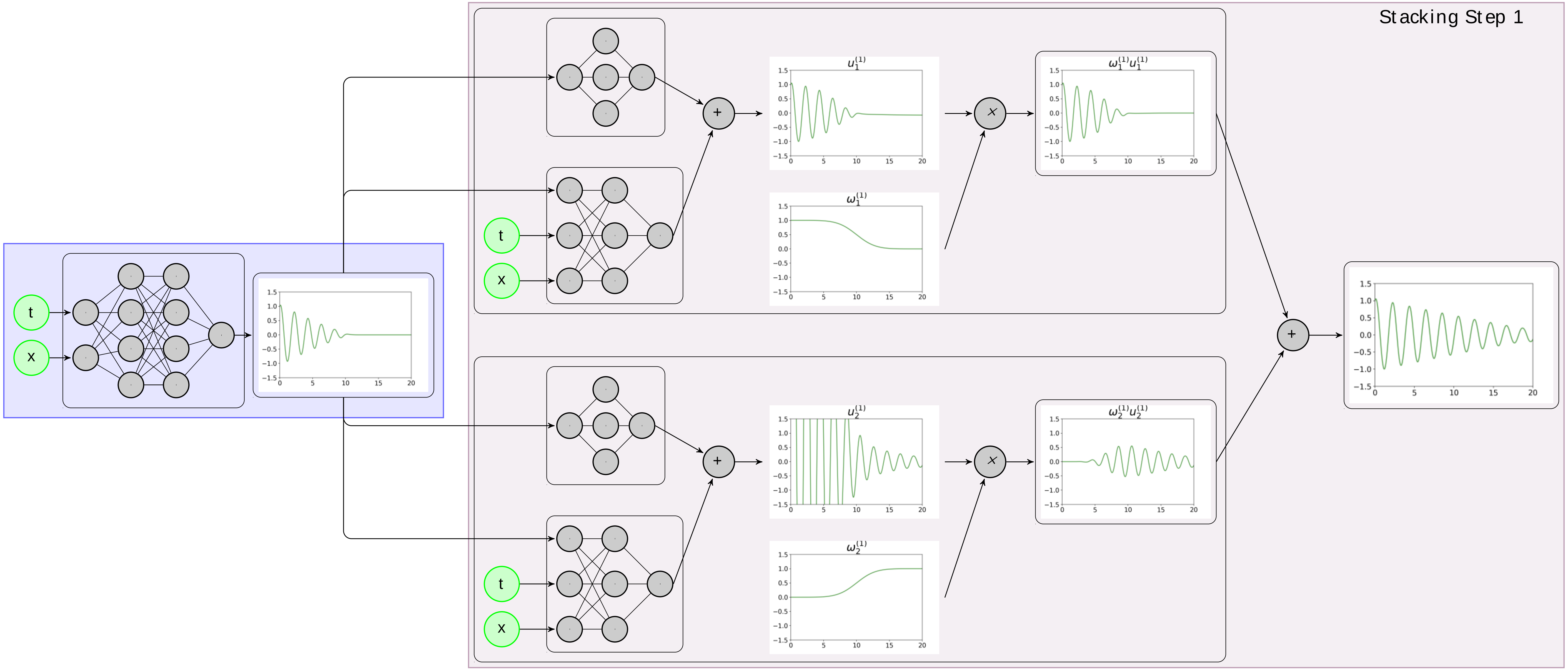
$$u_{j,MF}^{(l)}\left(x,\theta_j^{(l)}\right)=(1-|\alpha|)\,u_{j,linear}^{(l)}\left(x,\theta_j^{(l)}\right)+|\alpha|\,u_{j,nonlinear}^{(l)}\left(x,\theta_j^{(l)}\right)$$



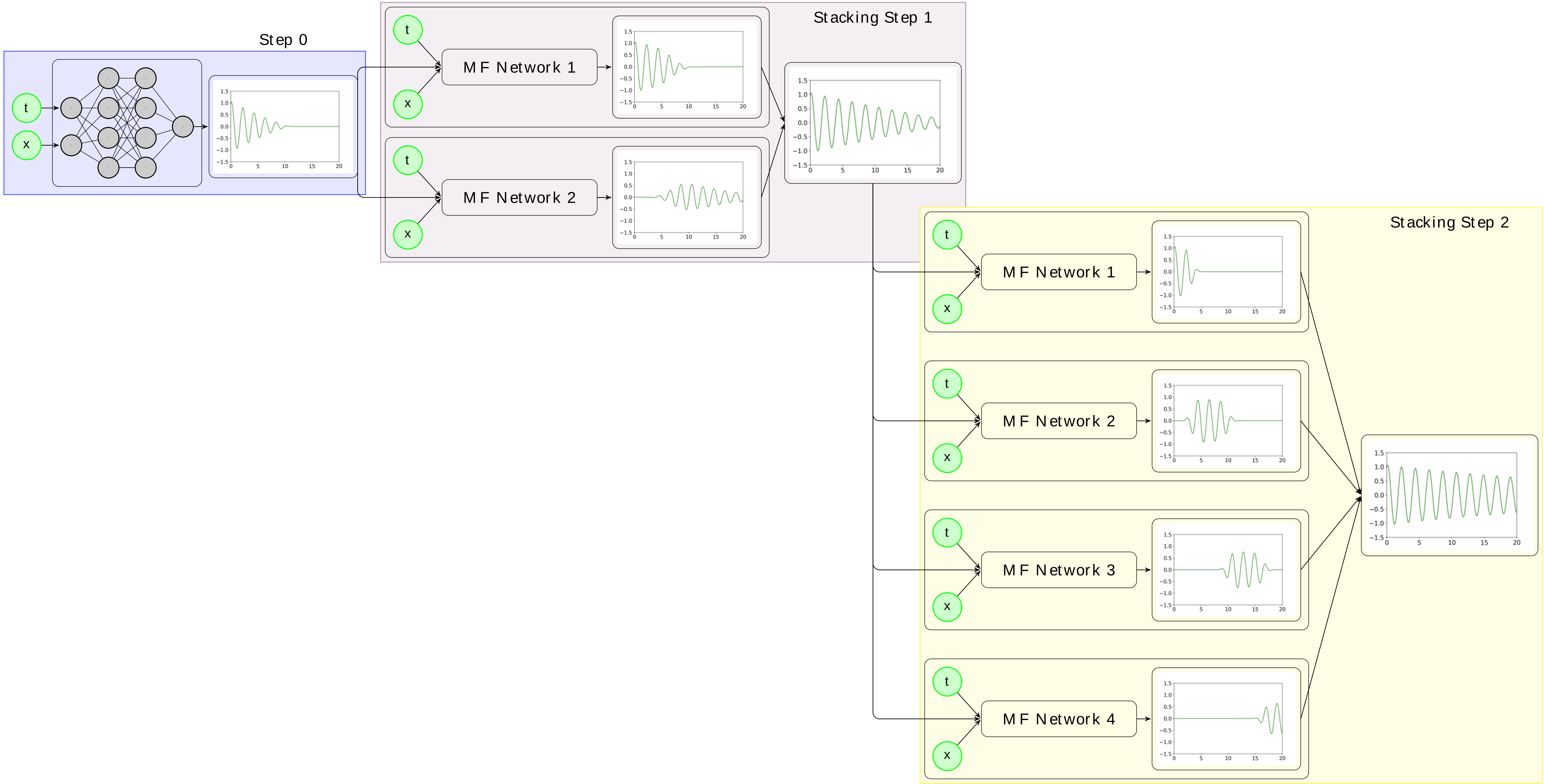




Stacking FBPINNs

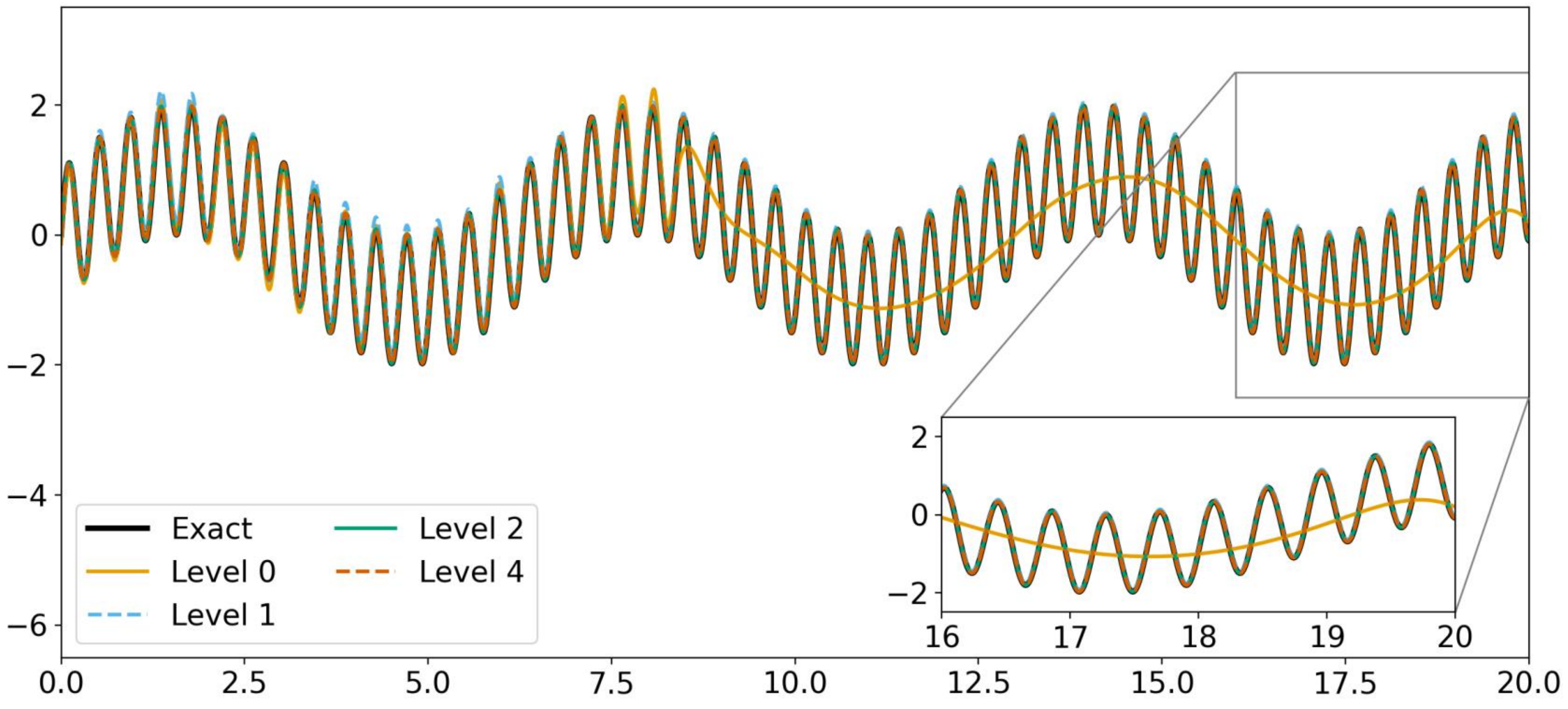


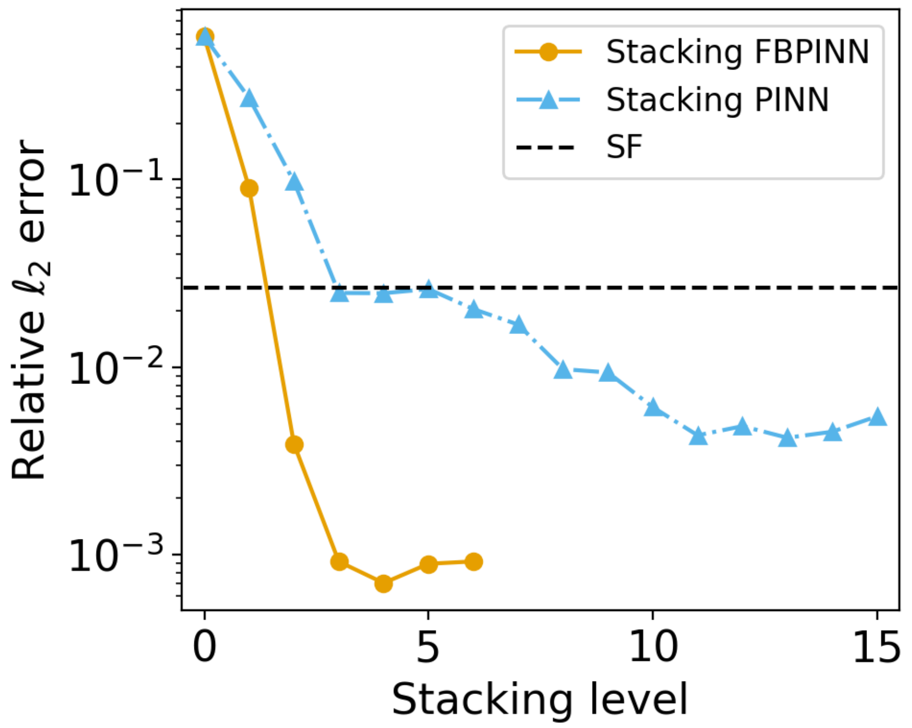
Stacking FBPINNs



Pacific Northwest NATIONAL LABORATORY

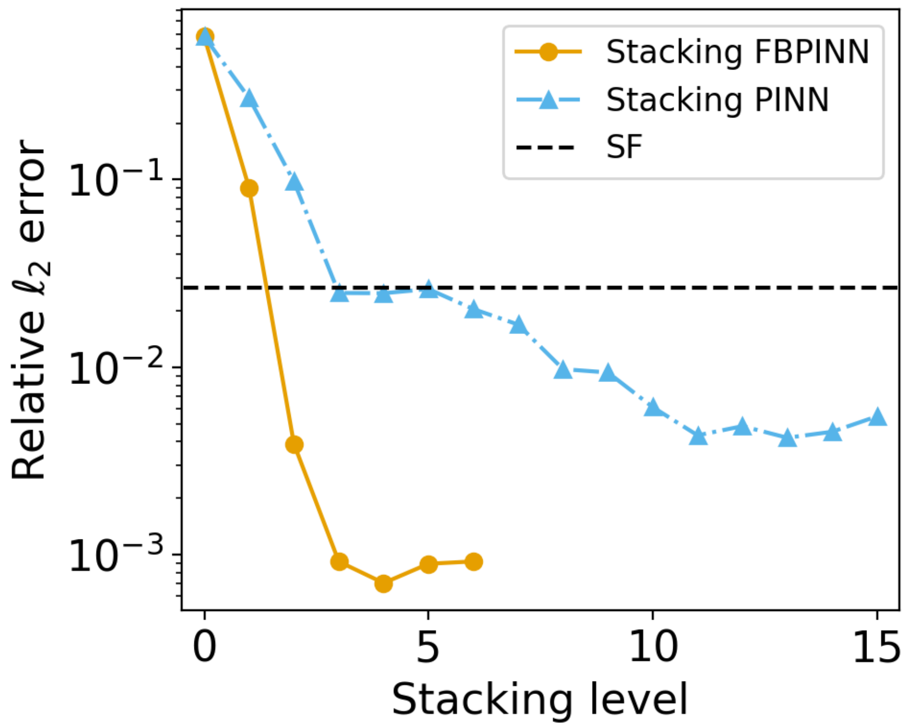
Multiscale



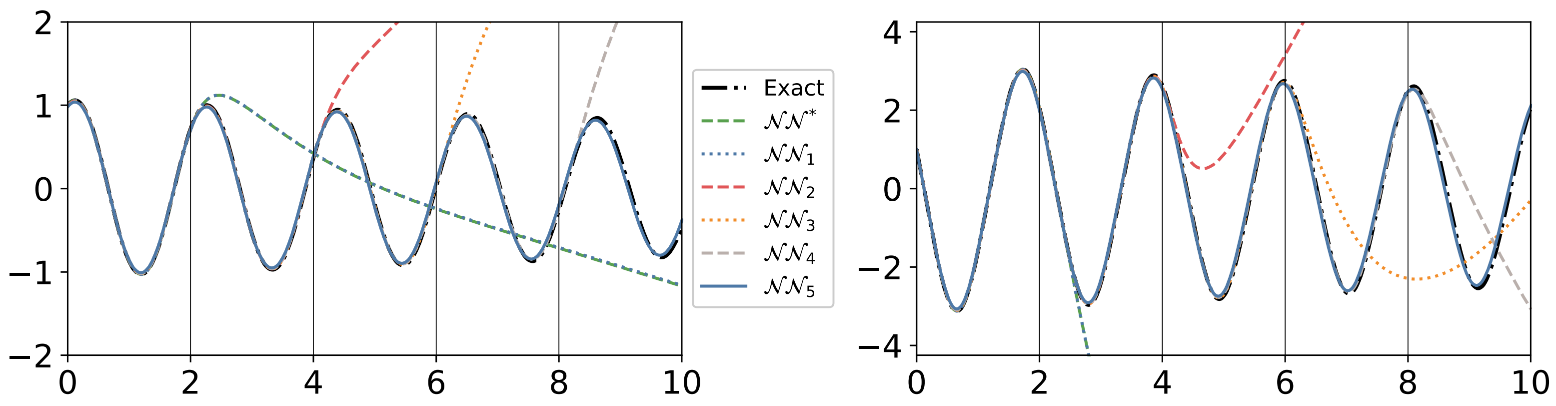


Multiscale

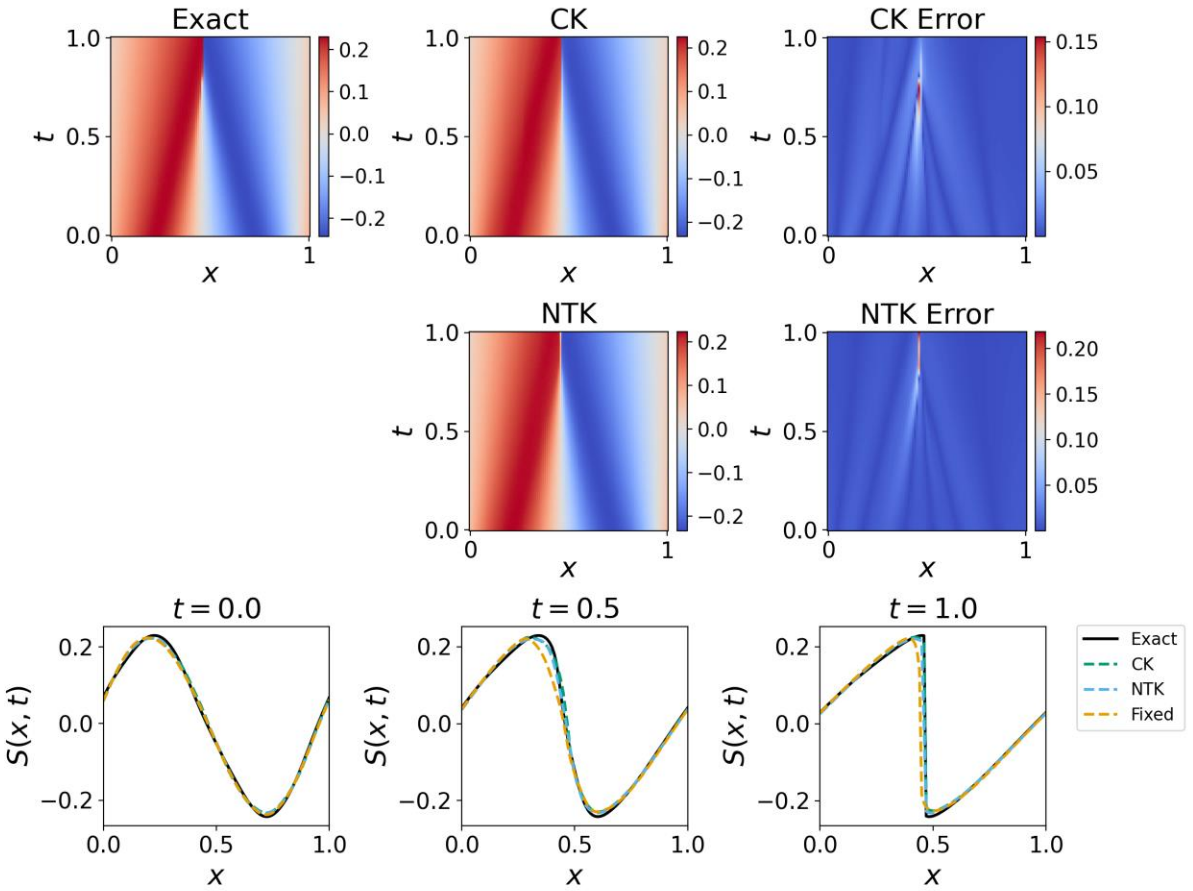
Method	Network size	Parameters	Error
PINN	4x64	12673	0.6543
PINN	5x64	16833	0.0265
Stacking PINN	4x16, 1x5, 3 levels	4900	0.0249
Stacking PINN	4x16, 1x5, 10 levels	11179	0.0061
Stacking FBPINN	**4x16, 1x5, 2 levels**	**7822**	**0.00415**
Stacking FBPINN	4x16, 1x5, 5 levels	59902	0.00083



Multifidelity methods can improve training of physics- and data-informed networks

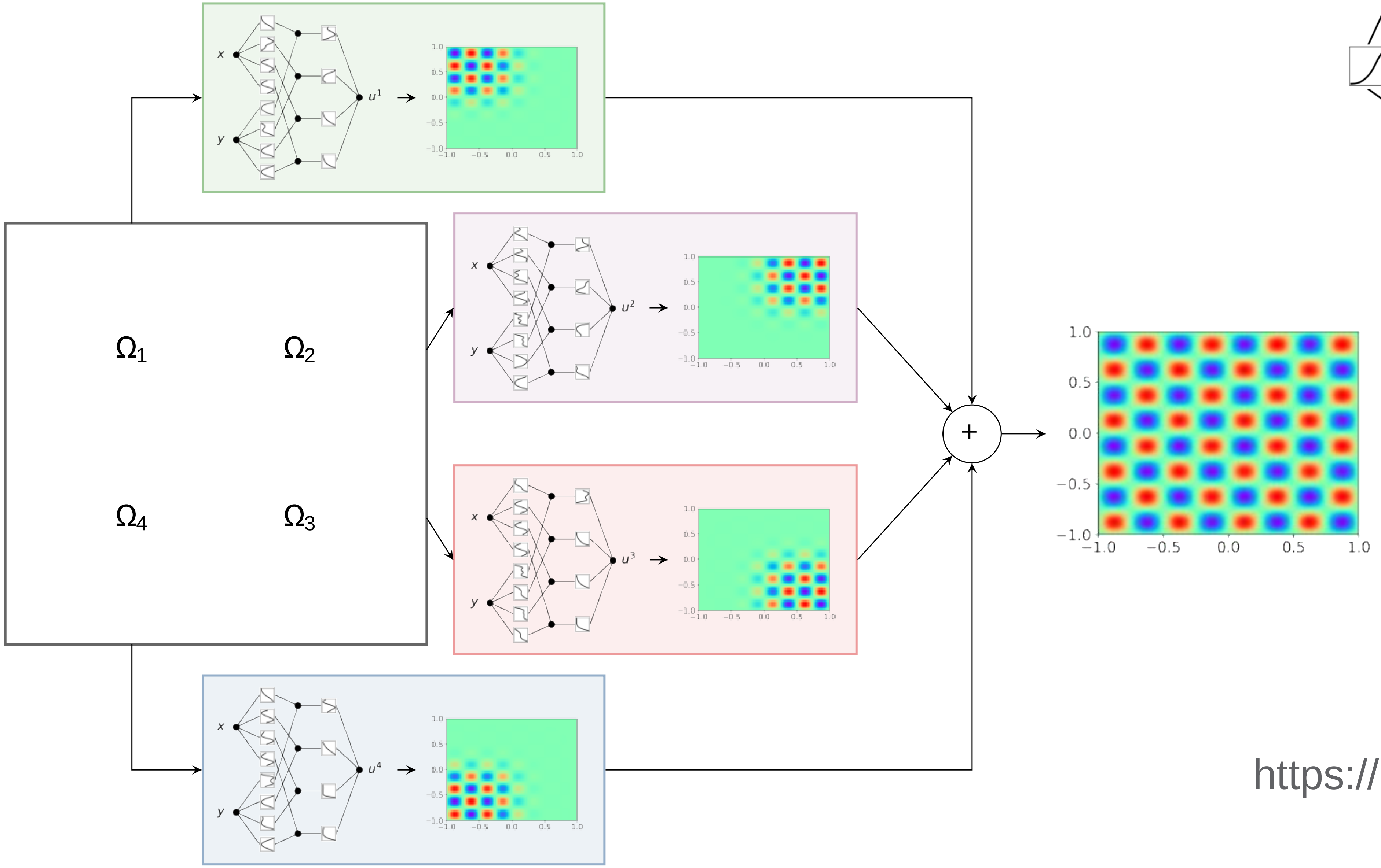


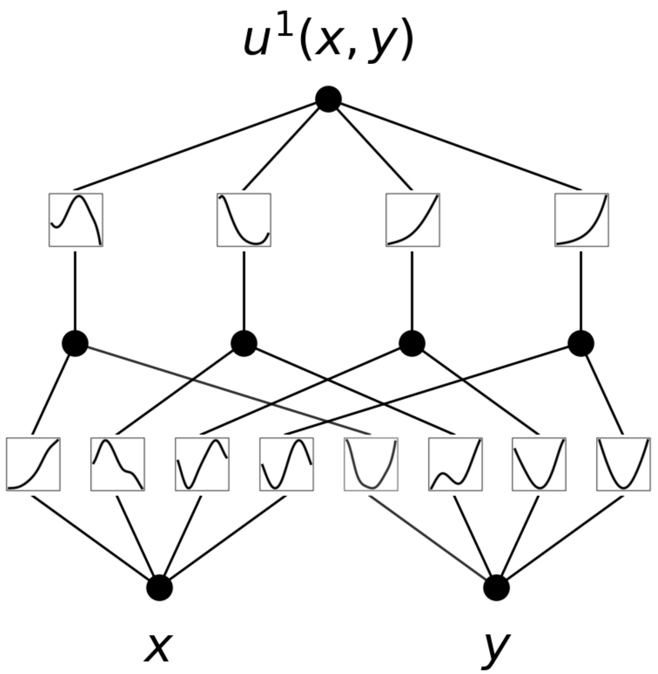
Continual learning with multifidelity arXiv:2304.03894



Conjugate kernel for accurate and fast training arXiv:2310.18612

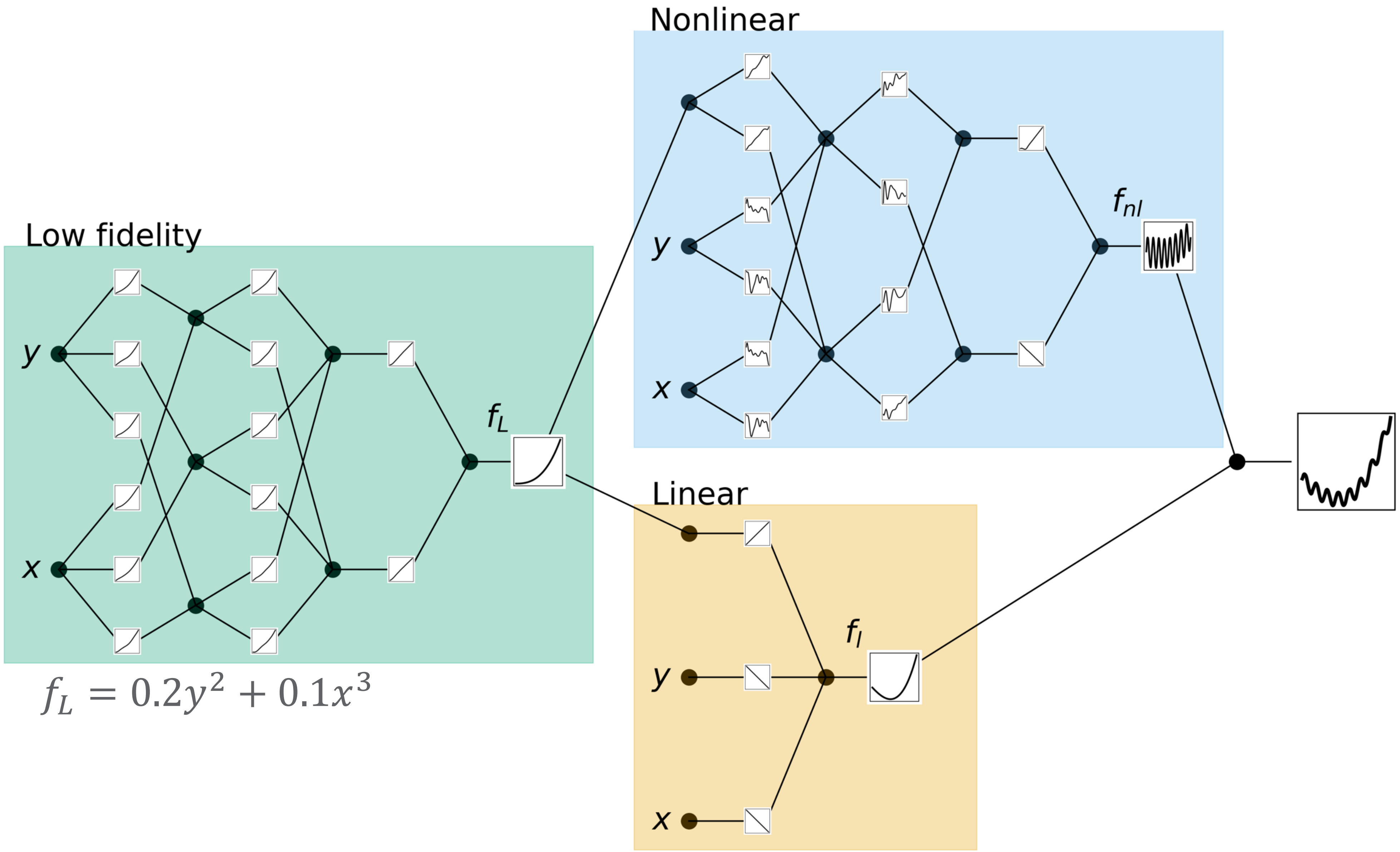
FBKANs





https://arxiv.org/abs/2406.19662

Multifidelity KANs

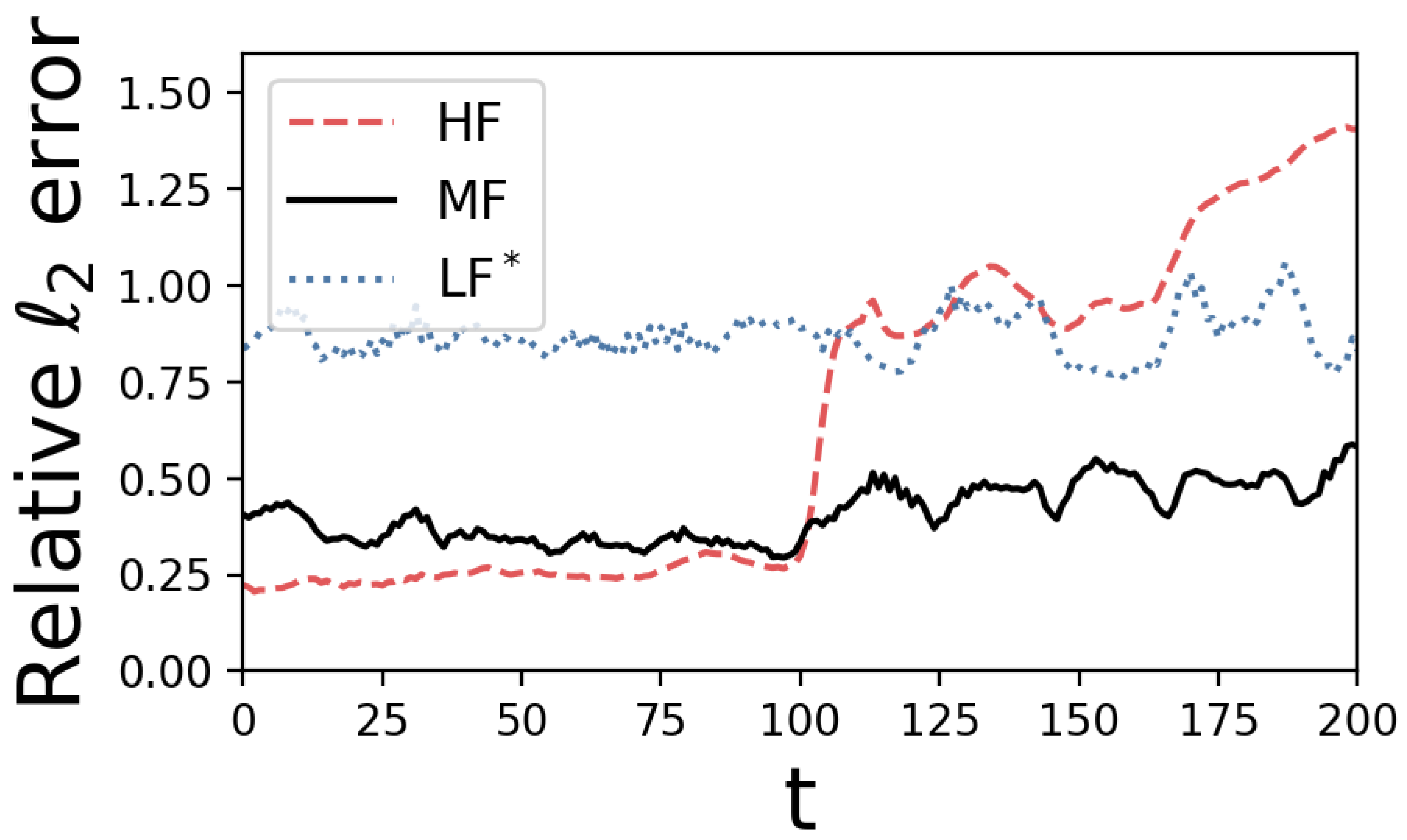


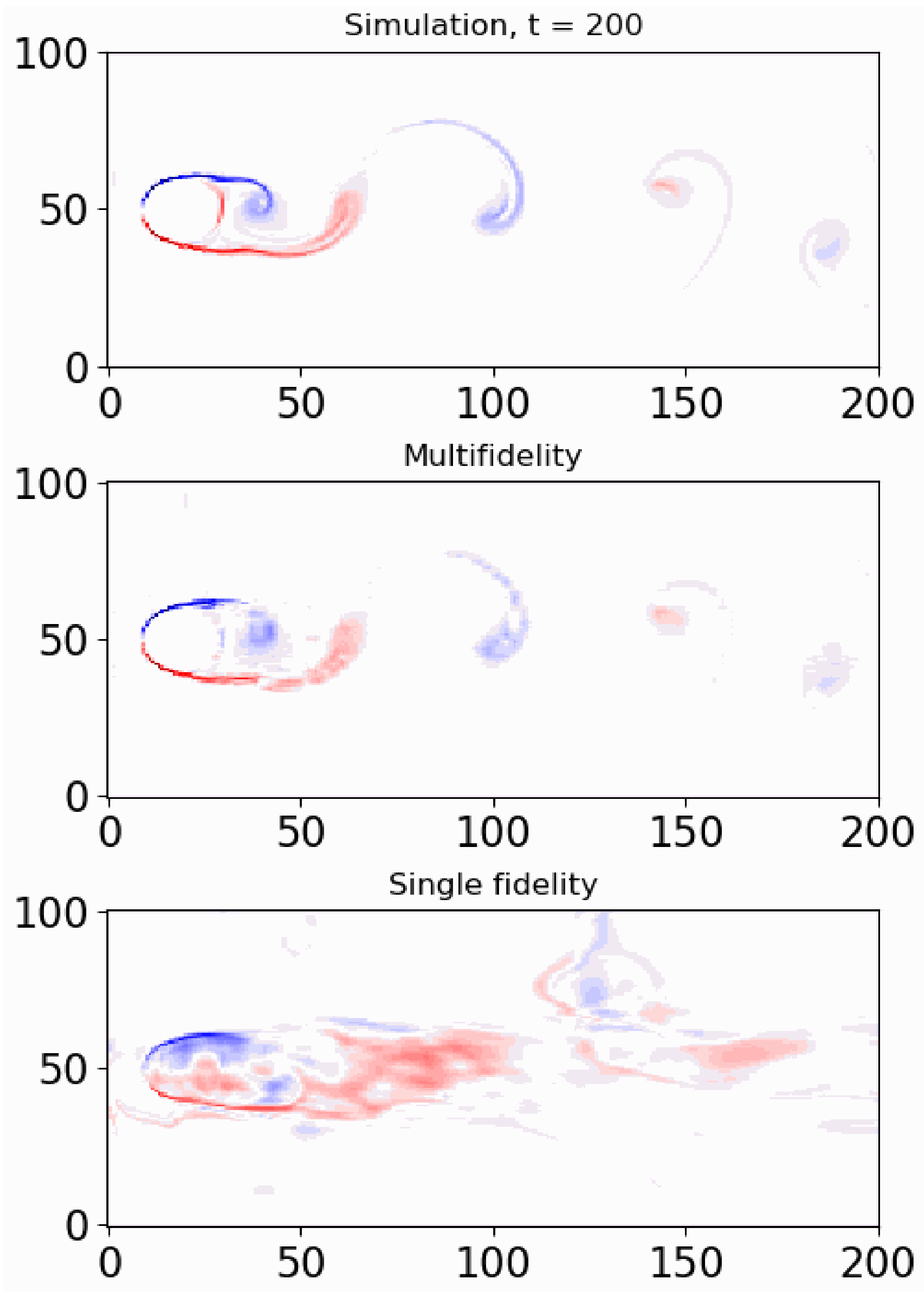
$f_L = 0.2y^2 + 0.1x^3$

$f_H = f_L - x - y + \sin(10x)$

Fluid flow

- Low fidelity data on t=[0, 200]
- High fidelity data only on t=[0, 100]
- Use low fidelity solver directly as a surrogate





Pacific Northwest
NATIONAL LABORATORY

Thank you

SEA-CROGS

Papers & code

https://multy.me/p1KTst

This project was completed with support from the U.S. Department of Energy, Advanced Scientific Computing Research program, under the Scalable, Efficient and Accelerated Causal Reasoning Operators, Graphs and Spikes for Earth and Embedded Systems (SEA-CROGS) project (Project No. 80278). Pacific Northwest National Laboratory (PNNL) is a multi-program national laboratory operated for the U.S. Department of Energy (DOE) by Battelle Memorial Institute under Contract No. DE-AC05-76RL01830. The computational work was performed using PNNL Institutional Computing.