Physics-Informed Operator Learning on Latent Spaces

Somdatta Goswami
Civil and Systems Engineering, Johns Hopkins University

January 10, 2025

Physics-based Models

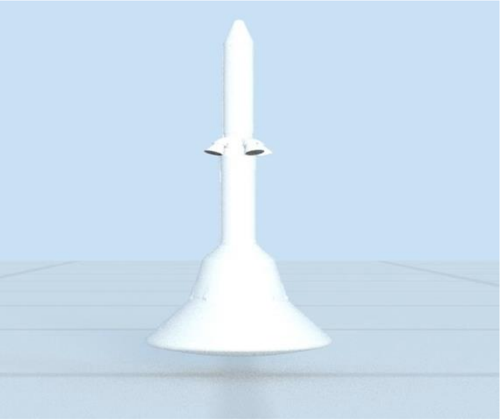
Can represent the Processes of Nature

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

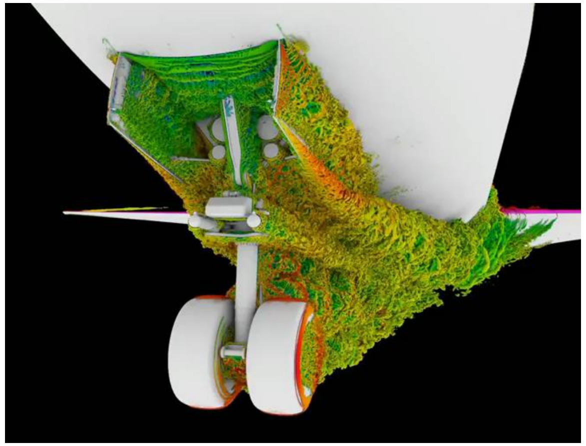
  To model earthquake: $m\frac{d^2u}{dt^2} + \mathrm{k}\frac{du}{dt} + F_0 = 0$

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

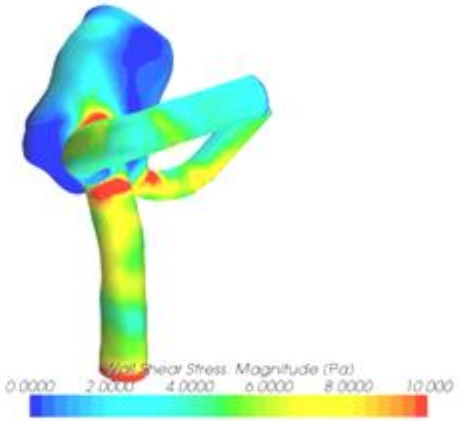
- Computational Mechanics helps us simulate these equations.



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

- Functional Data
- Discretization Invariant
- Continuous quantities
- Learning <u>operators</u> between function spaces

Data-driven

PCA

Auto-encoders

K-PCA

Diffusion maps

DeepONet

LNO

FNO

WNO

Finite Dimensional

Infinite Dimensional

Physics-Informed

PINNs

PI-DeepONet

PINO

Operator Learning Framework

Input-output map $\Phi\colon \mathcal{U} \rightarrow \mathcal{S}$

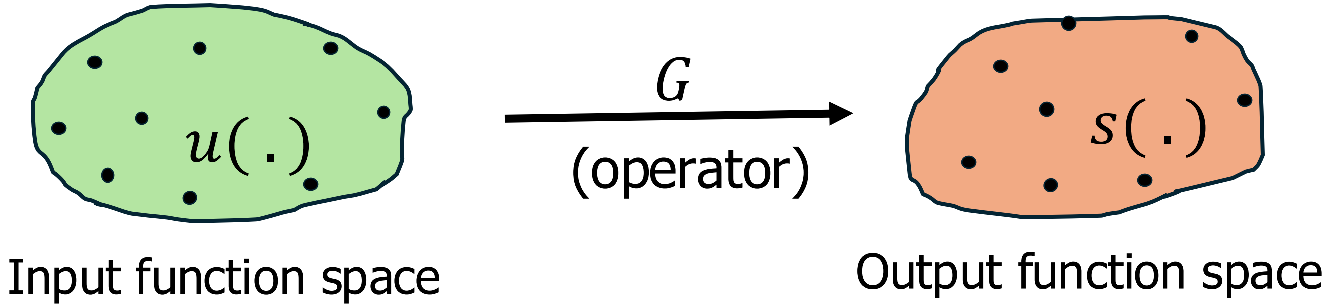
Data $\{\mathcal{U}_n, \mathcal{S}_n\}_{n=1}^{N}$ and/or Physics

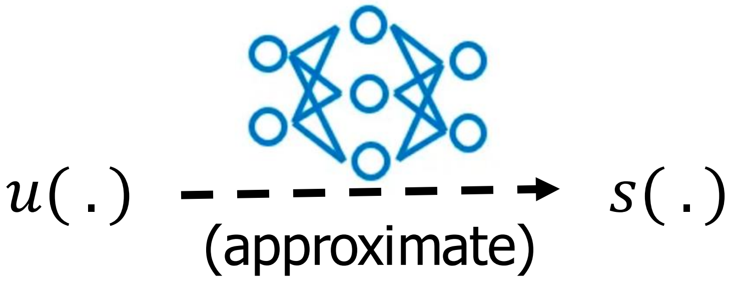
$$\mathcal{S}_n = \Phi(\mathcal{F}_n)\ ,\ \mathcal{F}_n \sim \mu\ i.i.d$$

Operator learning

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

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





NOs promise low generalization errors when trained with sufficiently rich dataset employing overparametrized networks for sufficiently large number of epochs.

Operator Learning Framework

NOs promise low generalization errors when trained with sufficiently rich dataset employing overparametrized networks for sufficiently long time.

overparametrized networks	→	Latent Operator Learning: Extract important features and learn the important features
sufficiently rich labeled dataset	→	Physics-Informed Operator Learning

So, to take advantage of both the solutions, we propose
Physics-Informed Operator Learning on Latent Spaces

Outline

Physics-Informed Operator Learning on Latent Spaces

Part – I: Data-driven operator learning on reduced spaces

Part – II: Integrating physics and data to learn operator on reduced spaces

Outline

Physics-Informed Operator Learning on Latent Spaces

Part – I: Data-driven operator learning on reduced spaces

Part – II: Integrating physics and data to learn operator on reduced spaces

nature communications

Article https://doi.org/10.1038/s41467-024-49411-w

Learning nonlinear operators in latent spaces for real-time predictions of complex dynamics in physical systems




Viscous Shallow water equation

- Model the dynamics of large-scale atmospheric flows
- Perturbation is used to induce the development of barotropic instability

$$\frac{\mathrm{d}\boldsymbol{V}}{\mathrm{d}t} = -f\boldsymbol{k} \times \boldsymbol{V} - g\nabla h + \nu\nabla^2\boldsymbol{V}$$

$$\frac{\mathrm{d}h}{\mathrm{d}t} = -h\nabla \cdot \boldsymbol{V} + \nu\nabla^2 h$$

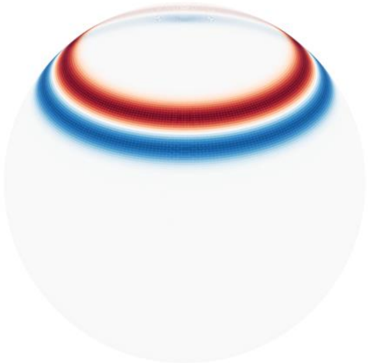
$$h'(\lambda, \phi) = \hat{h}\cos(\phi)\mathrm{e}^{-(\lambda/\alpha)^2}\mathrm{e}^{-[(\phi_2-\phi)/\beta]^2}$$

rvs: $\alpha \sim U[0.\bar{1}, 0.5]$ $\beta \sim U[0.0\bar{3}, 0.2]$

Operator: $\mathcal{G}: h'(\lambda, \varphi, t = 0) \mapsto u(\varphi, \lambda, t)$

Input Dimension: 65,536

Gaussian Random Perturbation



Output Dimension: 4,718,592

Atmospheric Flow

DeepONet for Viscous Shallow water equation

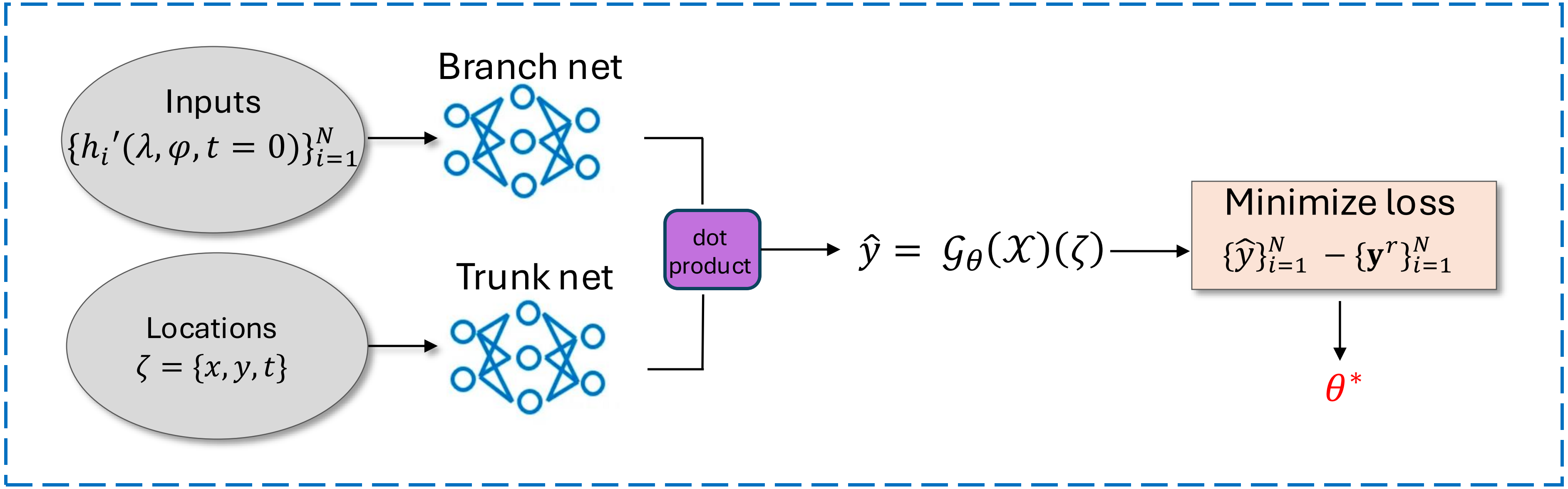
Operator $\mathcal{G} : \mathcal{X} \rightarrow \mathcal{Y}$

$\mathcal{G}_\theta : \mathcal{X} \rightarrow \mathcal{Y}, \quad \theta \in \Theta$

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

$\mathcal{X}$: the perturbed height field, $h_i'(\lambda, \varphi, t = 0)$

$\mathcal{Y}$: the velocity field, $u(\varphi, \lambda, t)$

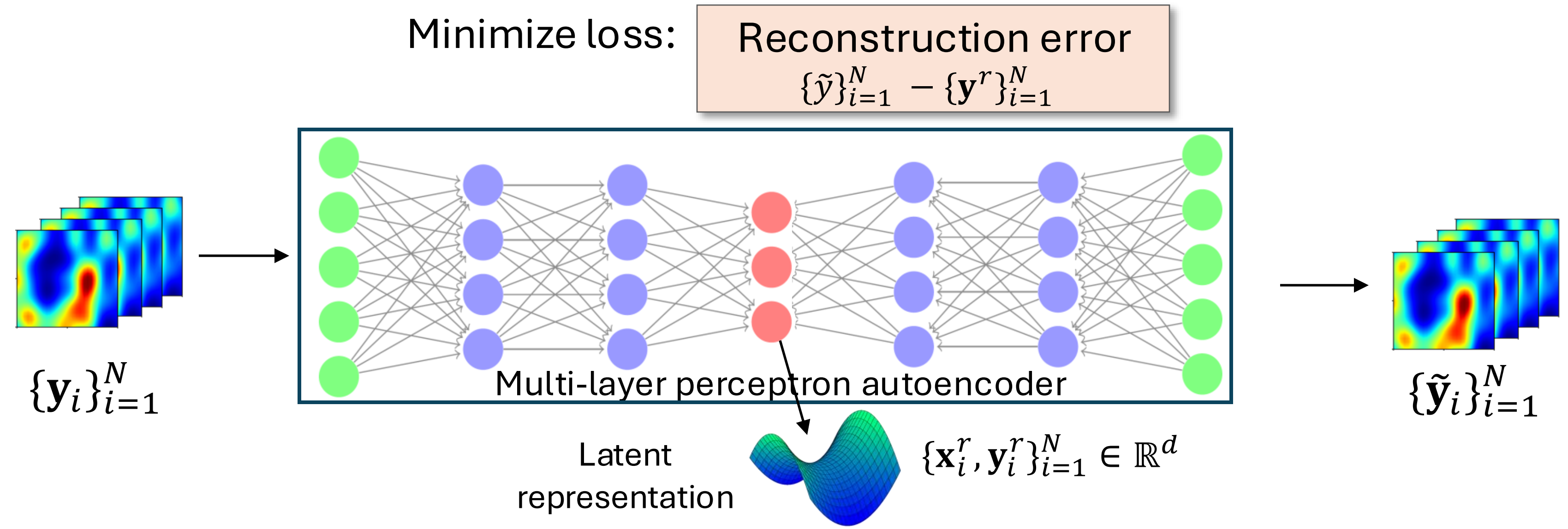


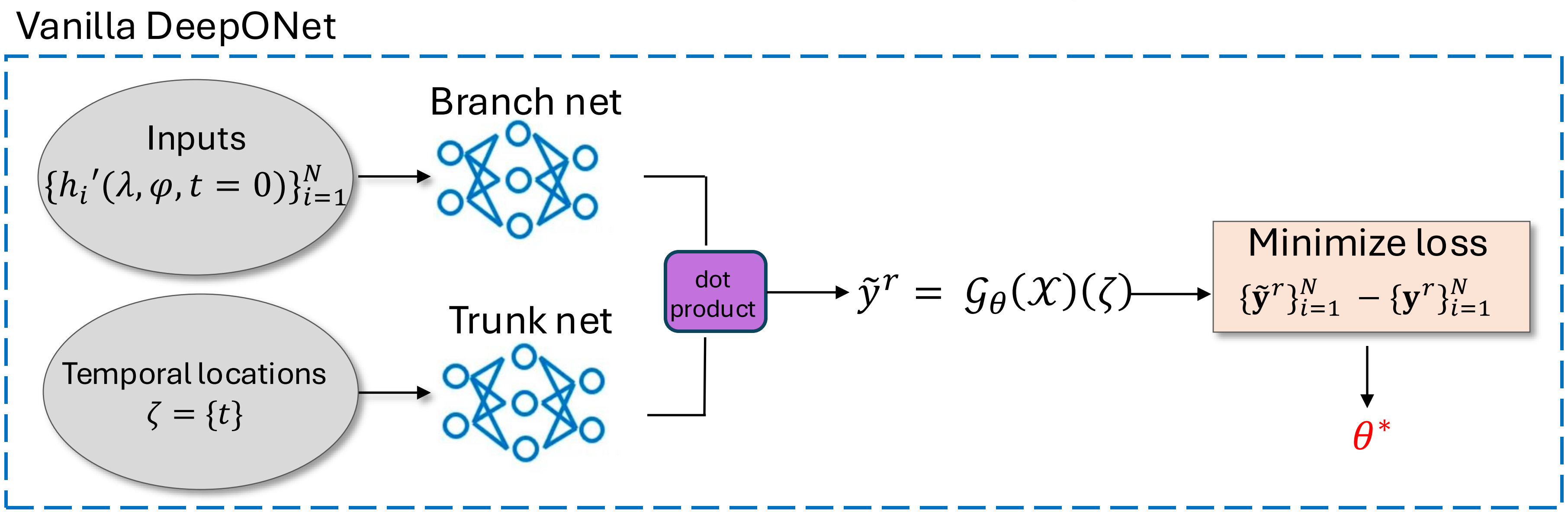
Latent DeepONet for Viscous Shallow Water

Operator $\mathcal{G}: \mathcal{X} \rightarrow \mathcal{Y}$

$\mathcal{G}_\theta: \mathcal{X} \rightarrow \mathcal{Y}, \quad \theta \in \Theta$

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





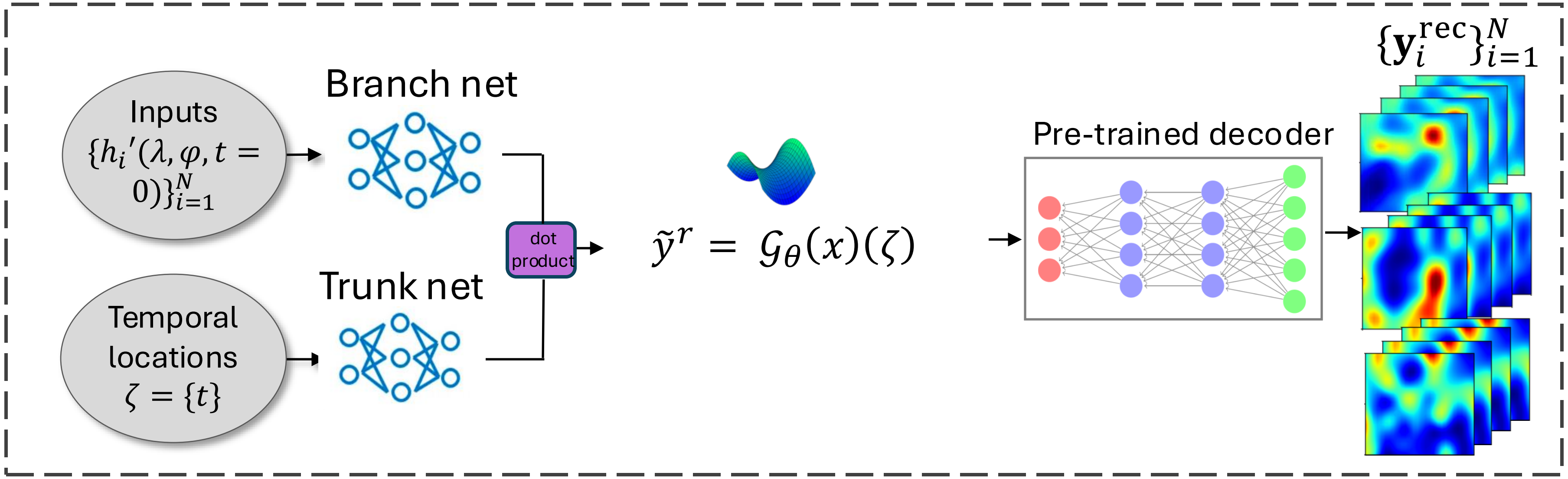
Latent DeepONet for Viscous Shallow Water

Operator $\mathcal{G} : \mathcal{X} \rightarrow \mathcal{Y}$

$\mathcal{G}_\theta : \mathcal{X} \rightarrow \mathcal{Y}, \quad \theta \in \Theta$

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

During Testing:

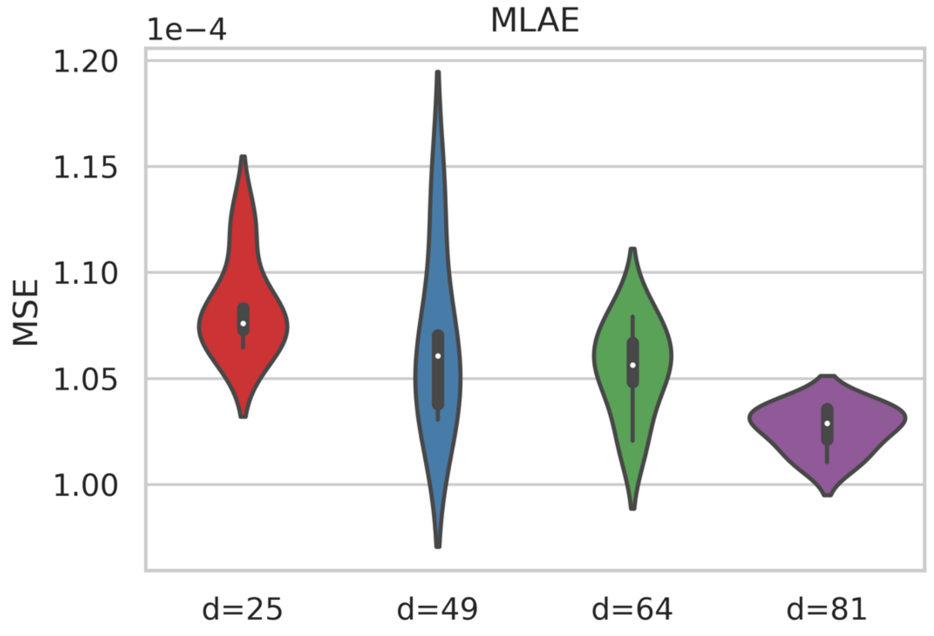


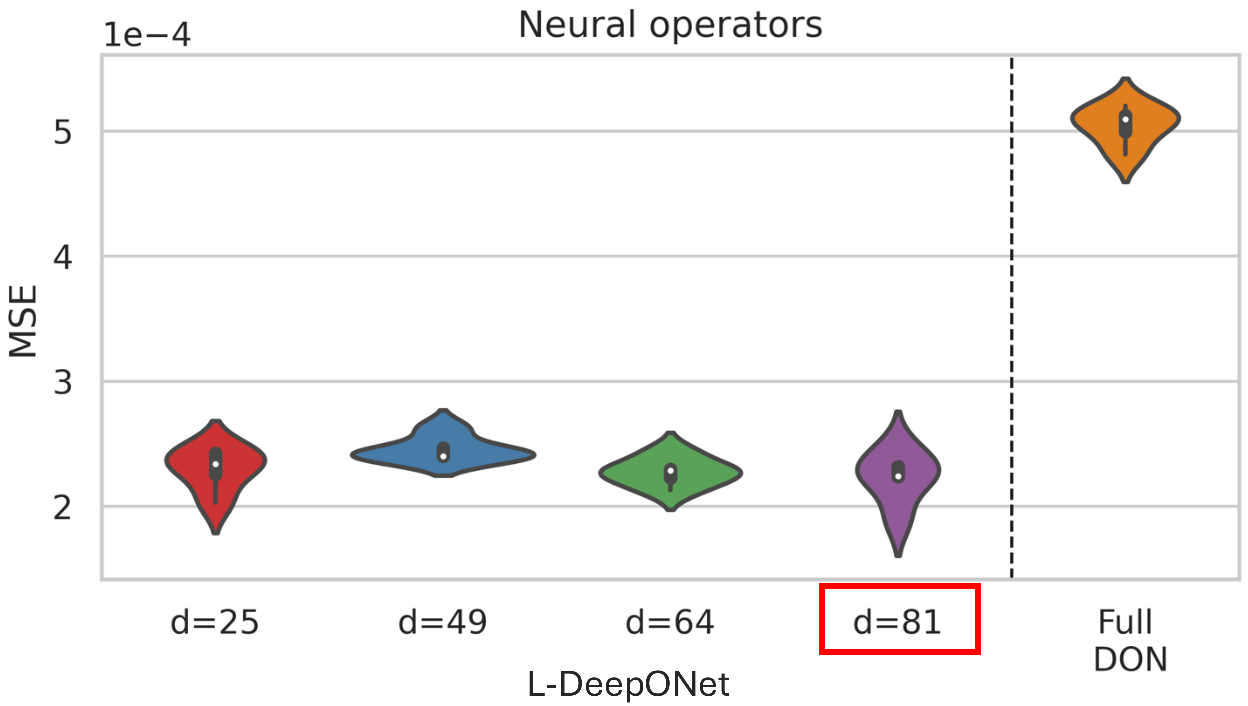
Results

- $\Omega = [0,2\pi]x[0,2\pi]$, $(n_x \; x \; n_y) = (256x256)$ mesh points
- Output dimensionality: $72x256x256 = 4{,}718{,}592$
- Simulation: $t = [0{,}360h], \delta t = 0.1\bar{6}h$, Time steps: $n_t = 72$

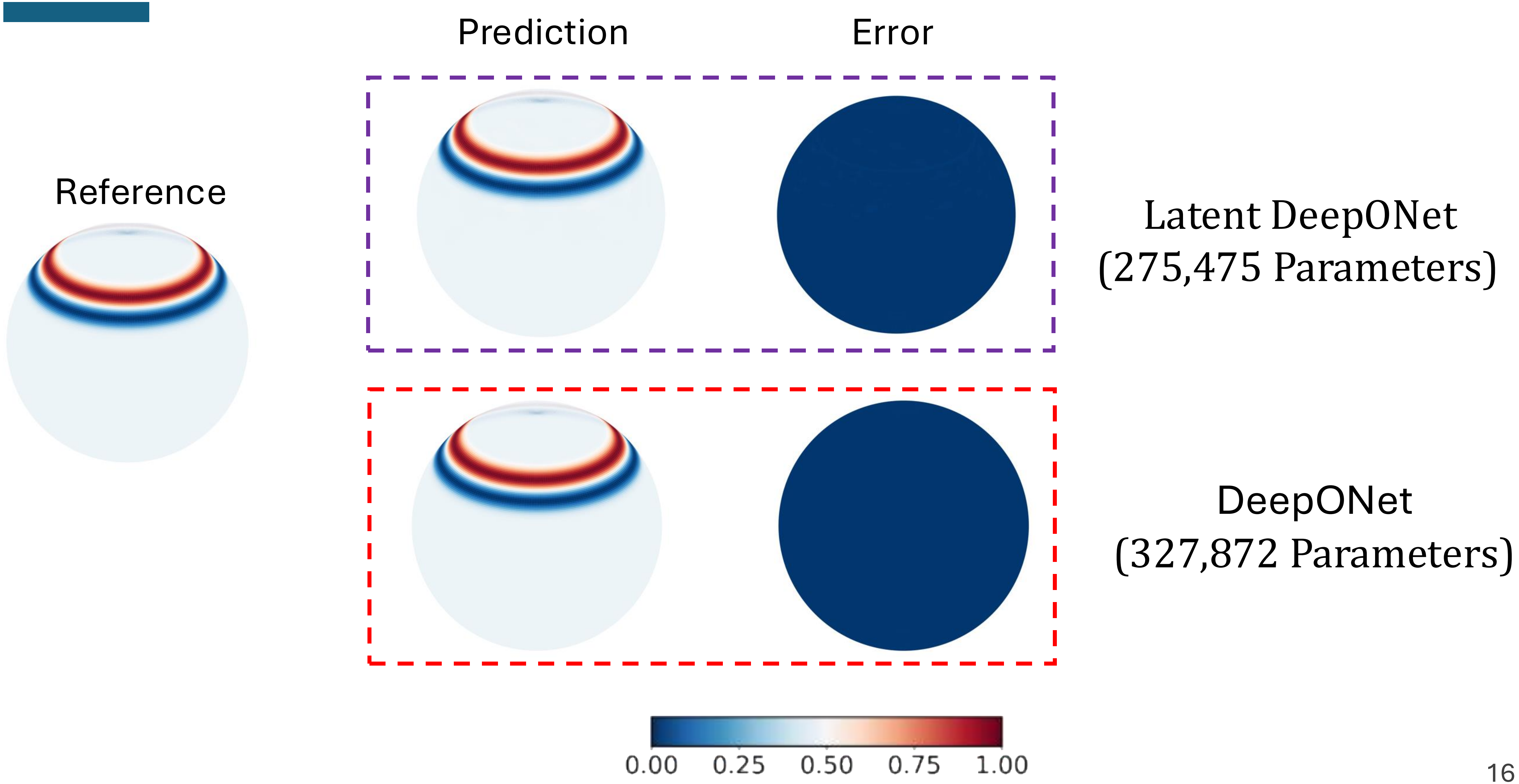
Training Time (seconds)

MLAE + Latent DON: 15, 218
Full DeepONet: 379,022

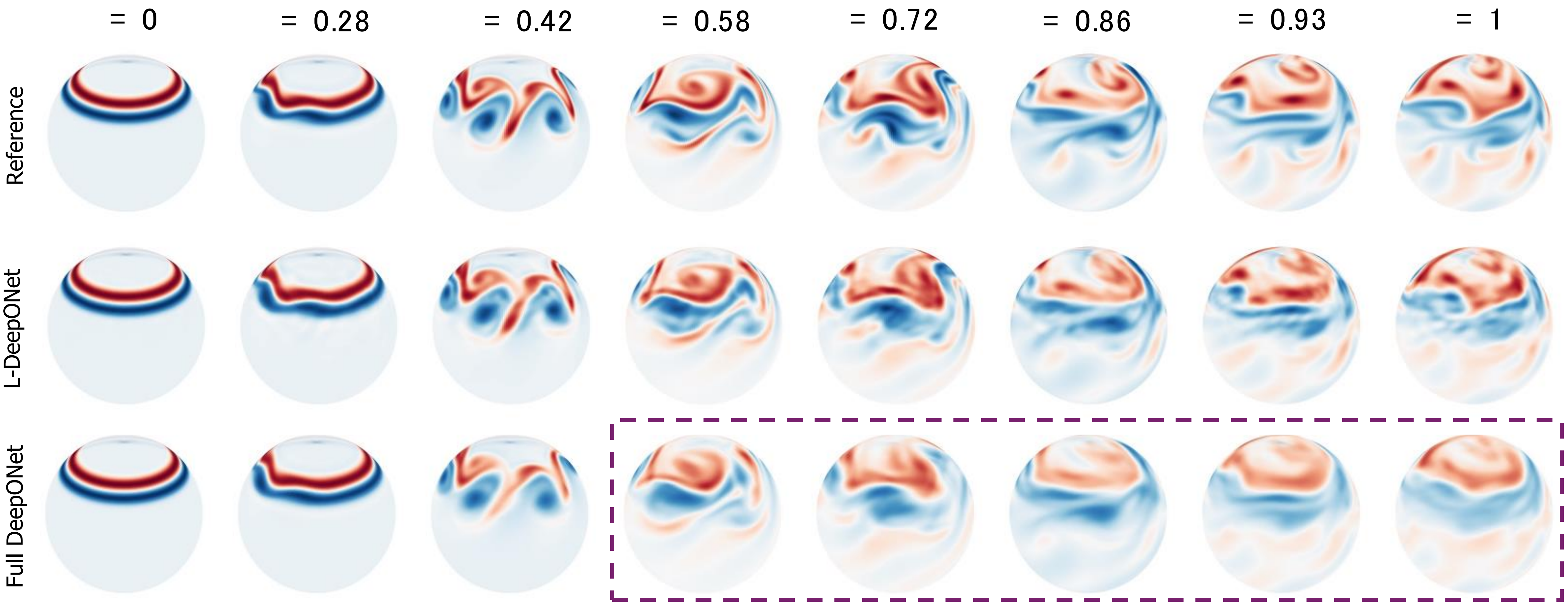




Results

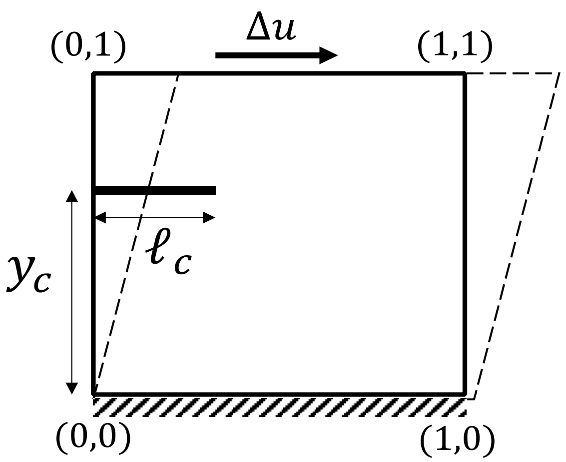

Prediction
Error
Reference
Latent DeepONet
(275,475 Parameters)
DeepONet
(327,872 Parameters)
0.00
0.25
0.50
0.75
1.00

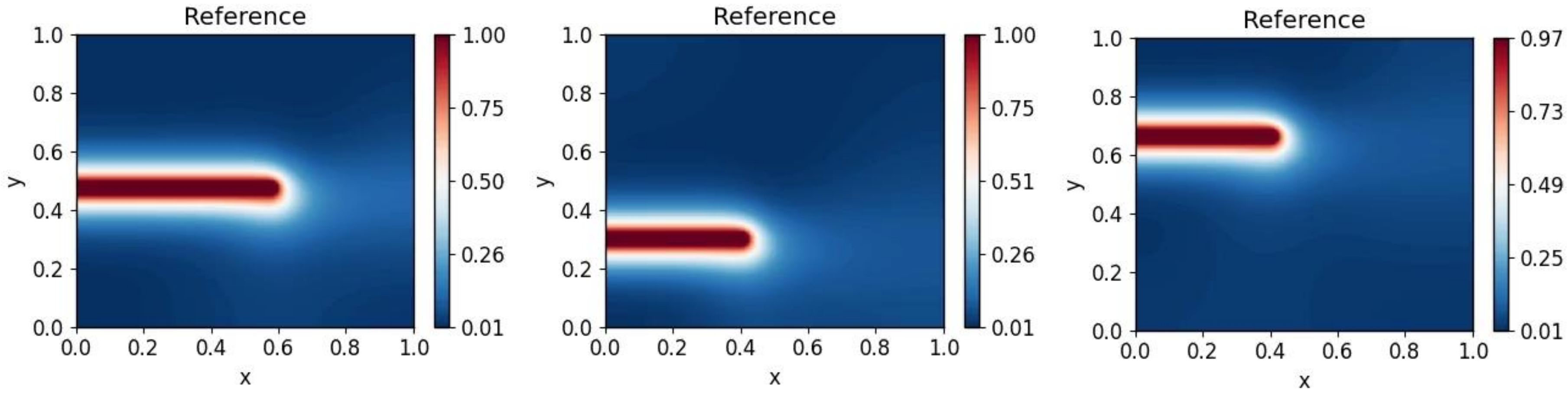
Latent DeepONet and Full DeepONet



Fracture: Shear failure of plate with notch

- Unit square plate with horizontal crack
- Both location y_c and length ℓ_c of the crack are considered random
- Boundary conditions: $u(x,0) = v(x,0) = 0, u(x,1) = \Delta u$
- Data: $N = 261$, $y_c \in [0.2, 0.675]$, $\ell_c \in [0.3, 0.65]$
- Input dimension: 162x162 Output dimension: 8x162x162

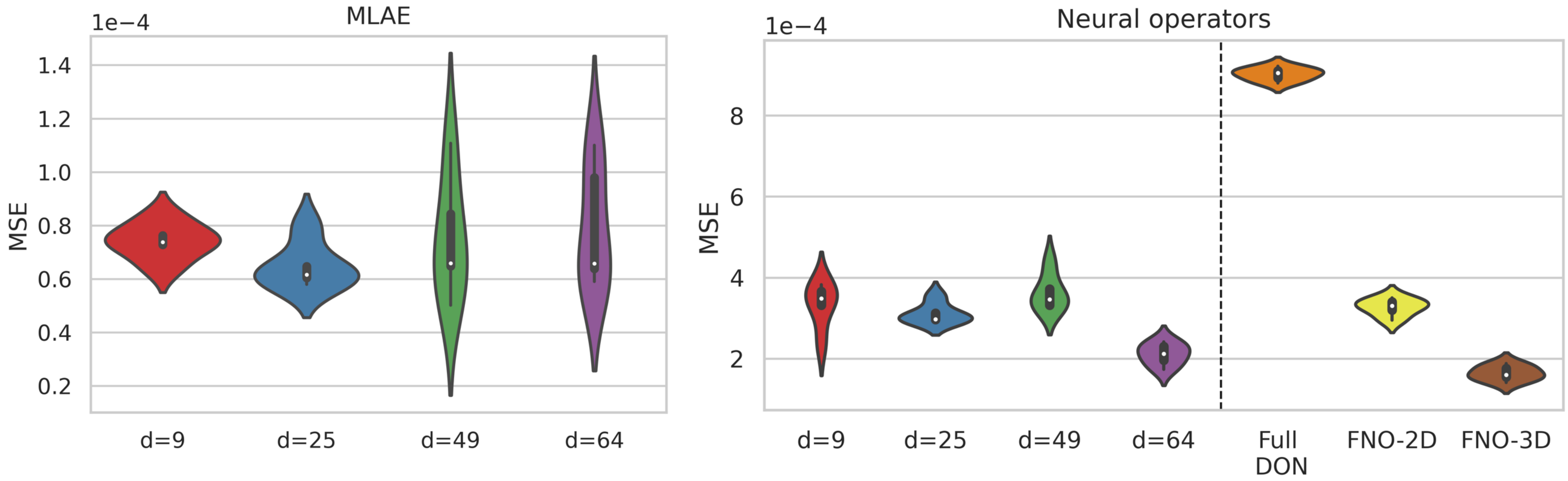




Goswami et al. (2021) A physics-informed variational DeepONet for predicting the crack path in brittle materials

Fracture: Shear failure of plate with notch

Error metric: $\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$

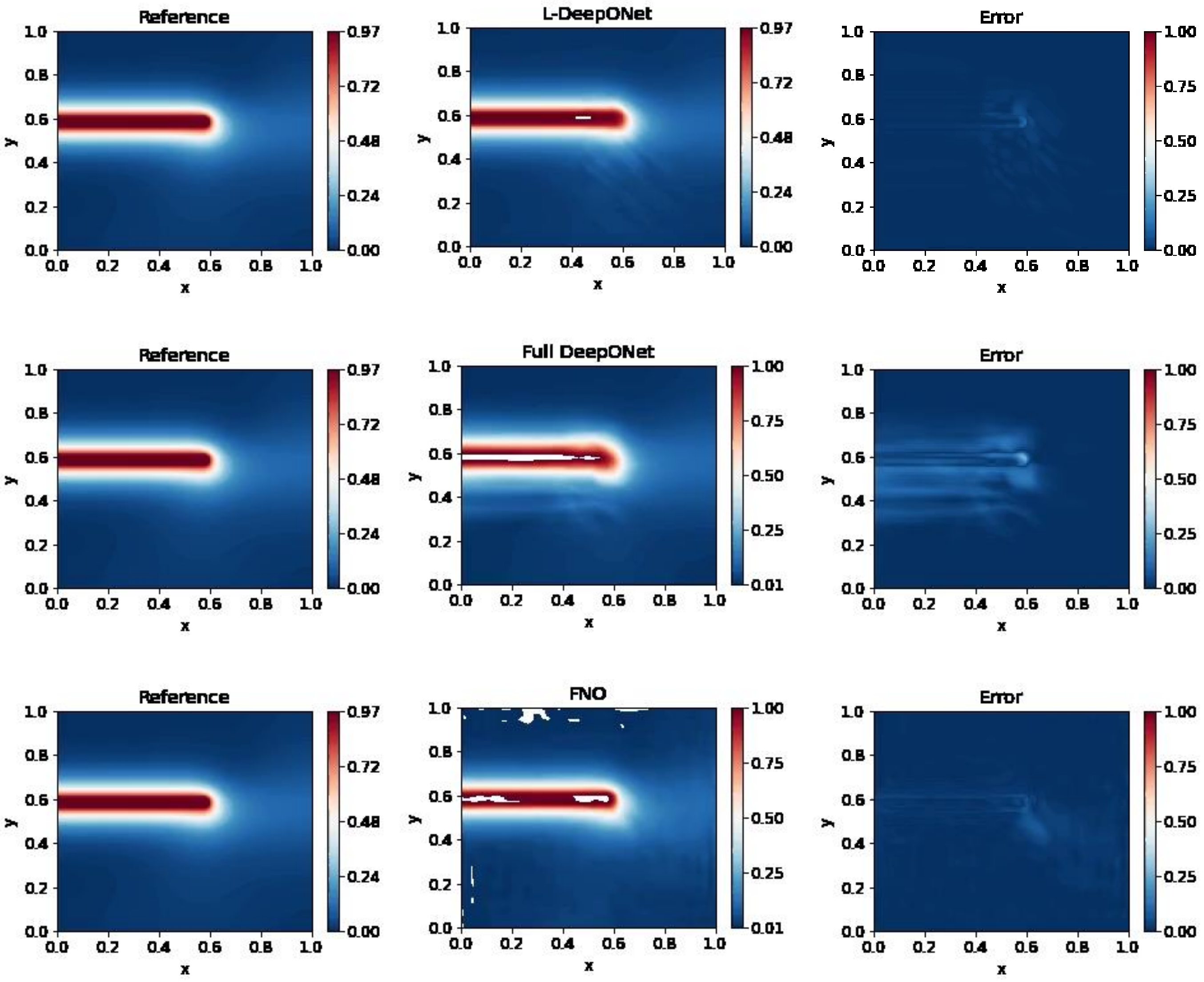


Comparison with Benchmark DeepONet

L-DeepONet

Full DeepONet

FNO



Outline

Physics-Informed Operator Learning on Latent Spaces

Part – I: Data-driven operator learning on reduced spaces

Part – II: Integrating physics and data to learn operator on reduced spaces

To incorporate the governing physics, we have to introduce a one step learning process (latent encoding + operator learning).

Our Proposed framework

Physics-Informed Latent Neural Operator: Integrating Physics and Data using Reduced Order Modeling



Manuscript in preparation

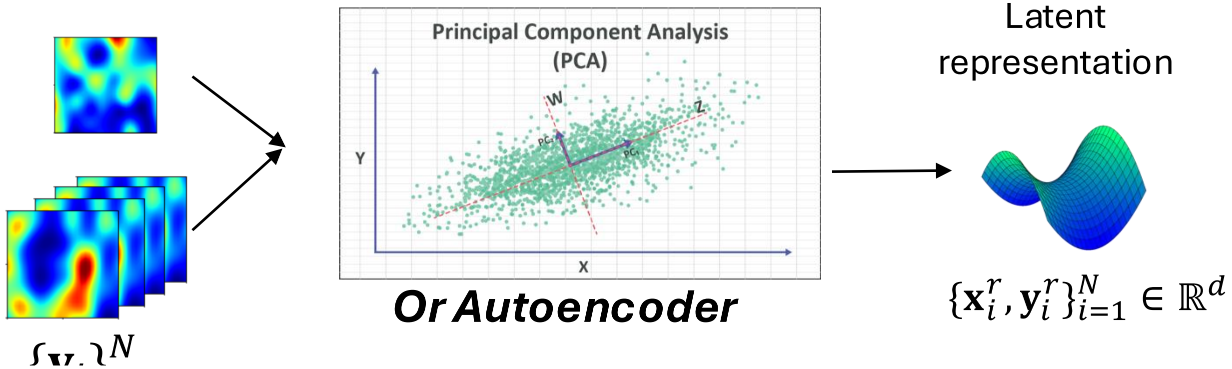
Framework I: Physics Informed Latent Neural Operator

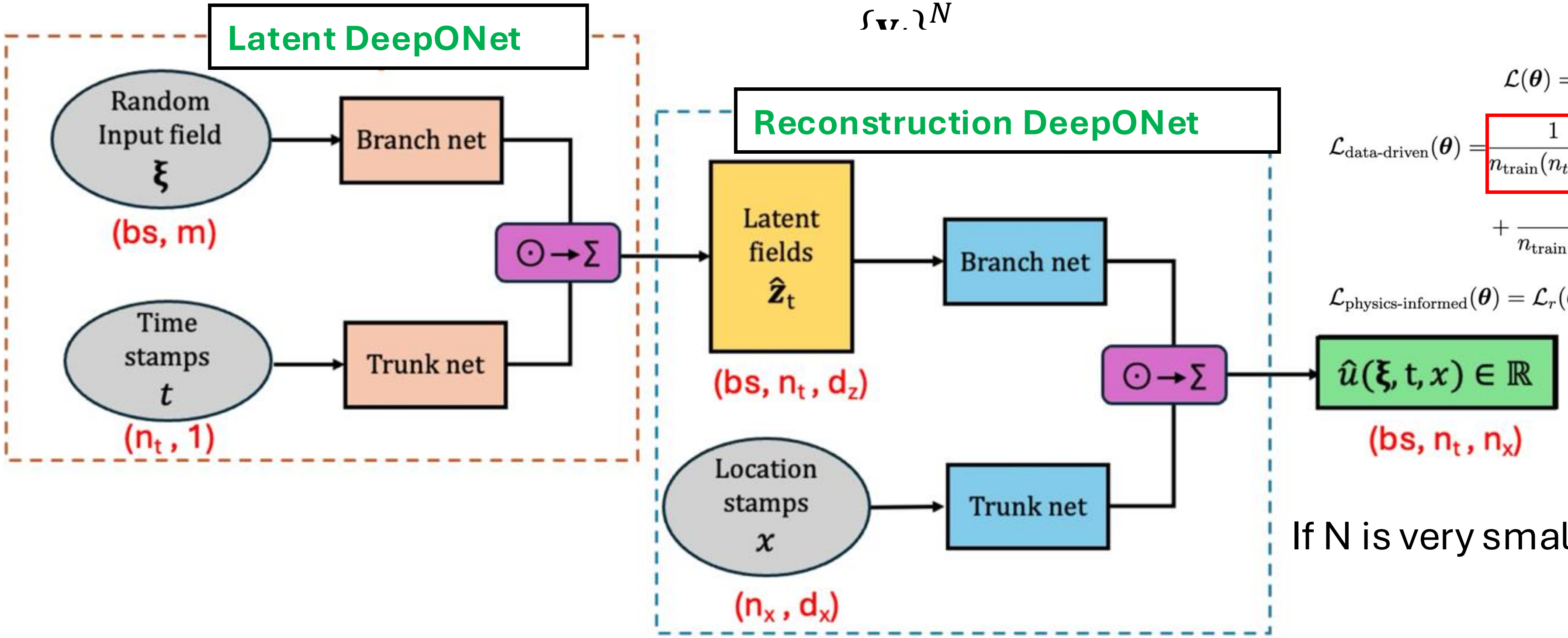
Operator $\mathcal{G} : \mathcal{X} \to \mathcal{Y}$

$\mathcal{G}_\theta : \mathcal{X} \to \mathcal{Y}, \quad \theta \in \Theta$

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

N is very small





$$\mathcal{L}(\boldsymbol{\theta}) = \mathcal{L}_{\text{data-driven}}(\boldsymbol{\theta}) + \mathcal{L}_{\text{physics-informed}}(\boldsymbol{\theta}),$$

$$\mathcal{L}_{\text{data-driven}}(\boldsymbol{\theta}) = \boxed{\frac{1}{n_{\text{train}}(n_t+1)} \sum_{i=1}^{n_{\text{train}}} \sum_{j=0}^{n_t} \left\| \mathbf{z}(\boldsymbol{\xi}^{(i)}, j\Delta t) - \hat{\mathbf{z}}(\boldsymbol{\xi}^{(i)}, j\Delta t) \right\|_2^2} + \frac{1}{n_{\text{train}}(n_t+1)n_{\boldsymbol{x}}} \sum_{i=1}^{n_{\text{train}}} \sum_{j=0}^{n_t} \sum_{k=1}^{n_{\boldsymbol{x}}} \left(u(\boldsymbol{\xi}^{(i)}, j\Delta t, \boldsymbol{x}^{(k)}) - \hat{u}(\boldsymbol{\xi}^{(i)}, j\Delta t, \boldsymbol{x}^{(k)}) \right)^2,$$

$$\mathcal{L}_{\text{physics-informed}}(\boldsymbol{\theta}) = \mathcal{L}_r(\boldsymbol{\theta}) + \mathcal{L}_{bc}(\boldsymbol{\theta}) + \mathcal{L}_{ic}(\boldsymbol{\theta}).$$

If N is very small, wrong latent spaces are identified.

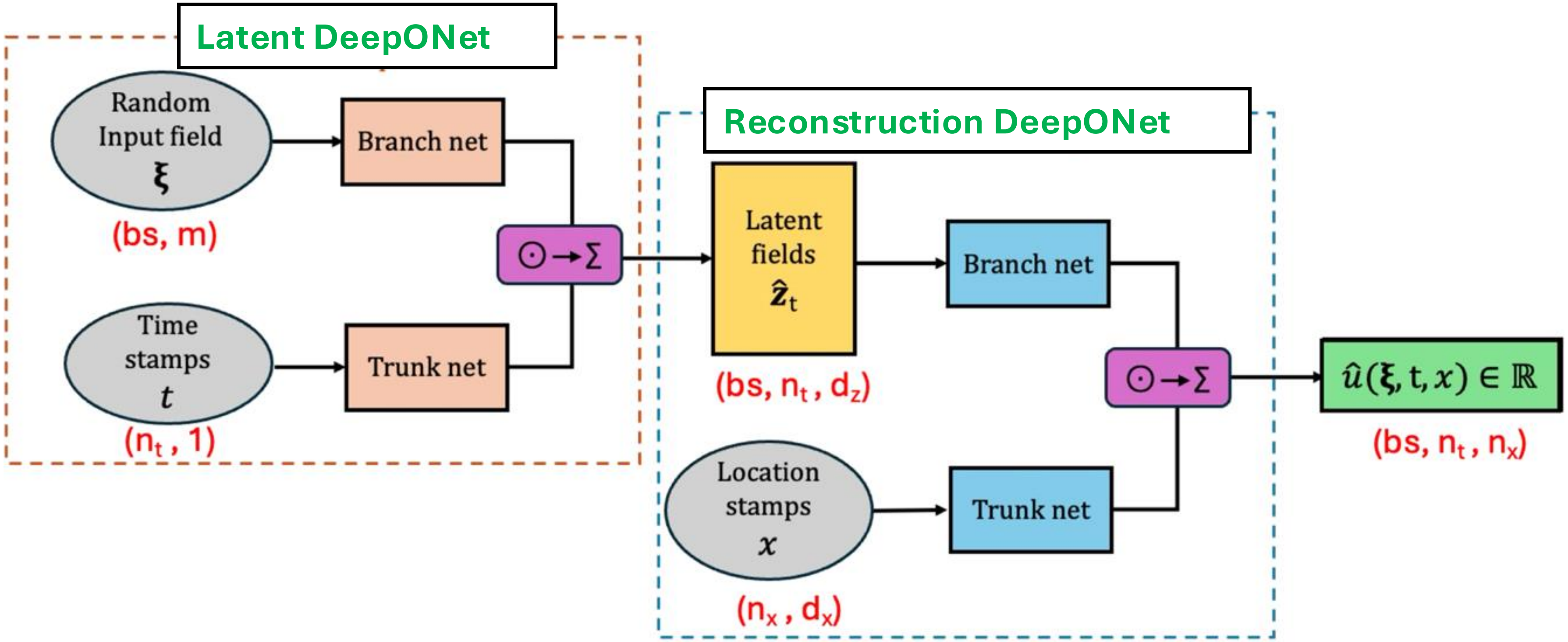
Framework II: Physics Informed Latent Neural Operator

Operator $\mathcal{G} : \mathcal{X} \to \mathcal{Y}$

$\mathcal{G}_\theta : \mathcal{X} \to \mathcal{Y}, \quad \theta \in \Theta$

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

N is either zero or very small



$$\mathcal{L}(\boldsymbol{\theta}) = \mathcal{L}_{\text{data-driven}}(\boldsymbol{\theta}) + \mathcal{L}_{\text{physics-informed}}(\boldsymbol{\theta}),$$

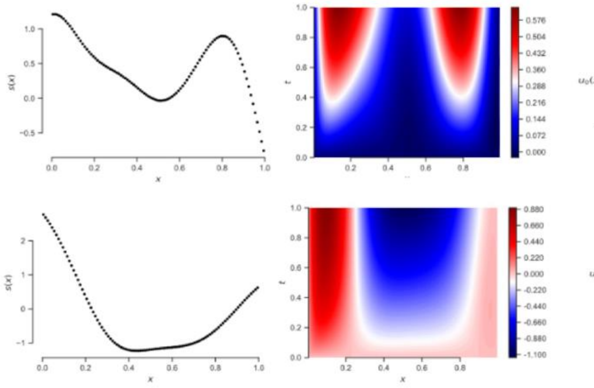
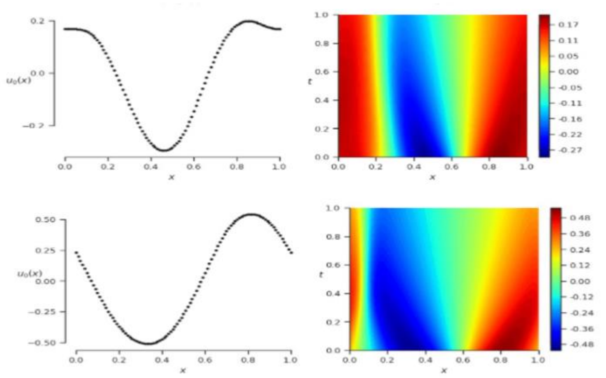
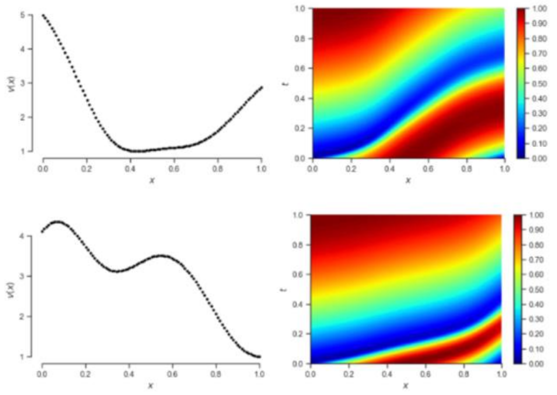
$$\mathcal{L}_{\text{data-driven}}(\boldsymbol{\theta}) = \frac{1}{n_{\text{train}}(n_t+1)n_{\boldsymbol{x}}} \sum_{i=1}^{n_{\text{train}}} \sum_{j=0}^{n_t} \sum_{k=1}^{n_{\boldsymbol{x}}} \left(u(\boldsymbol{\xi}^{(i)}, j\Delta t, \boldsymbol{x}^{(k)}) - \hat{u}(\boldsymbol{\xi}^{(i)}, j\Delta t, \boldsymbol{x}^{(k)})\right)^2,$$

$$\mathcal{L}_{\text{physics-informed}}(\boldsymbol{\theta}) = \mathcal{L}_r(\boldsymbol{\theta}) + \mathcal{L}_{bc}(\boldsymbol{\theta}) + \mathcal{L}_{ic}(\boldsymbol{\theta}).$$

Advantages

Allows for temporal and spatial interpolation

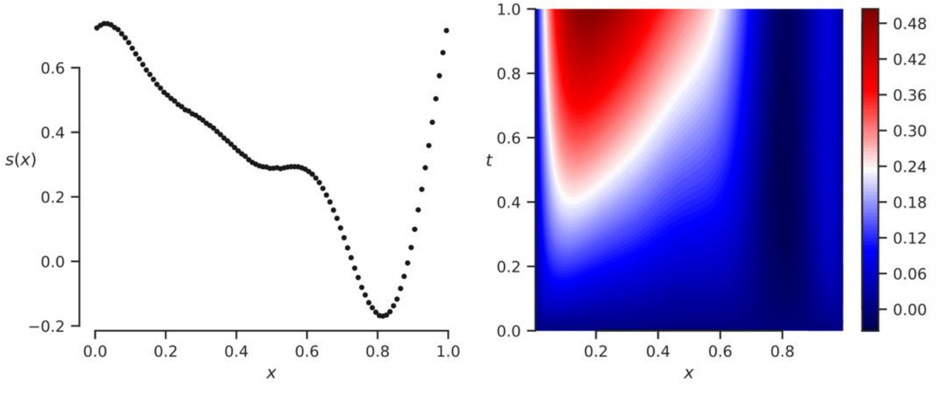
Introduces separability and accelerates training

Case	Diffusion-reaction dynamics	Burgers' transport dynamics	Advection
PDE	$\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2} + ku^2 + s(x),$ $D = 0.01,\ k = 0.01,$ $(t,x) \in (0,1] \times (0,1],$ $u(0,x) = 0,\ x \in (0,1)$ $u(t,0) = 0,\ t \in (0,1)$ $u(t,1) = 0,\ t \in (0,1)$ $\mathcal{G}_{\boldsymbol{\theta}} : s(x) \to u(t,x).$	$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - \nu\frac{\partial^2 u}{\partial x^2} = 0,$ $\nu = 0.01,$ $(t,x) \in (0,1] \times (0,1],$ $u(0,x) = g(x),\ x \in (0,1)$ $u(t,0) = u(t,1)$ $\frac{\partial u}{\partial x}(t,0) = \frac{\partial u}{\partial x}(t,1)$ $\mathcal{G}_{\boldsymbol{\theta}} : g(x) \to u(t,x).$	$\frac{\partial u}{\partial t} + s(x)\frac{\partial u}{\partial x} = 0,$ $(t,x) \in (0,1] \times (0,1],$ $u(0,x) = \sin(\pi x)\ \forall\ x \in (0,1),$ $u(t,0) = \sin(0.5\pi t)\ \forall\ t \in (0,1),$ $s(x) = v(x) - \min_x v(x) + 1$ $\mathcal{G}_{\boldsymbol{\theta}} : v(x) \to u(t,x).$
Input Function	$s(x) \sim \mathrm{GP}(0, k(x,x')),$ $\ell_x = 0.2,\ \sigma^2 = 1.0,$ $k(x,x') = \sigma^2 \exp\left\{-\frac{\\|x-x'\\|^2}{2\ell_x^2}\right\}.$	$g(x) \sim \mathcal{N}\left(0, 25^2\left(-\Delta + 5^2 I\right)^{-4}\right),$	$v(x) \sim \mathrm{GP}(0, k(x,x')),$ $\ell_x = 0.2,\ \sigma^2 = 1.0,$ $k(x,x') = \sigma^2 \exp\left\{-\frac{\\|x-x'\\|^2}{2\ell_x^2}\right\}.$
Samples			

Reaction Diffusion Dynamics

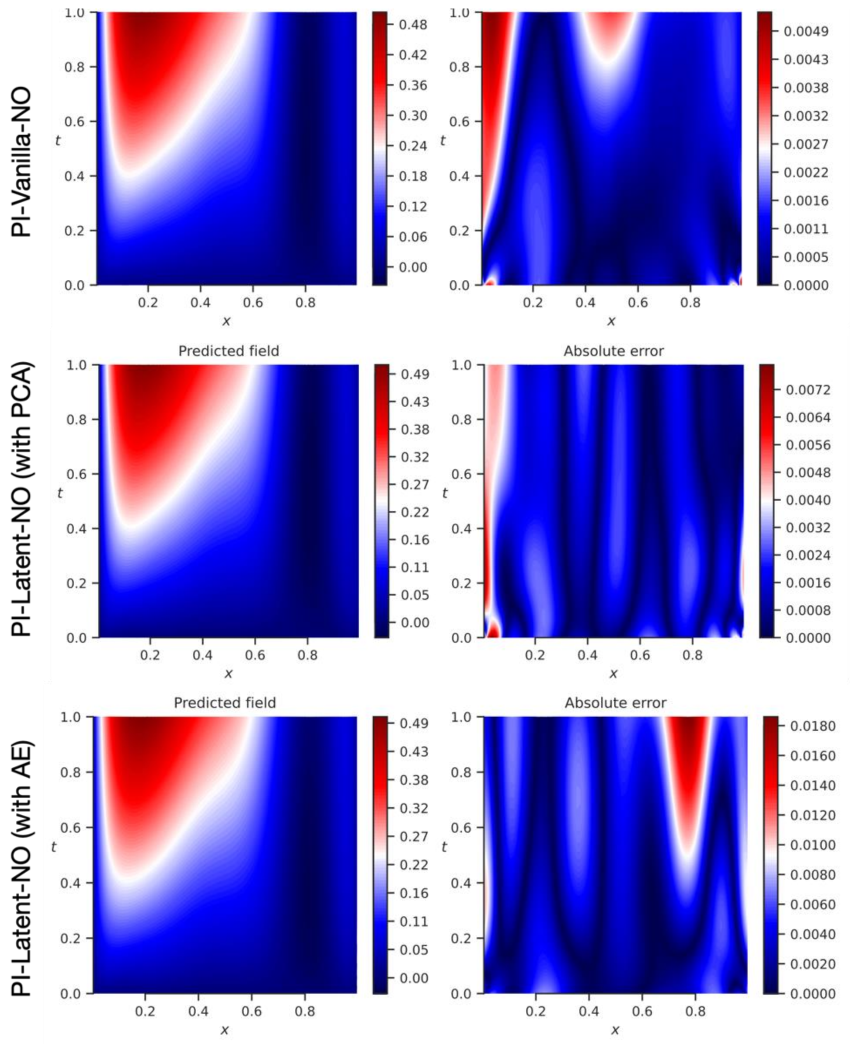
Source
(Input function)

Solution
(Ground Truth)

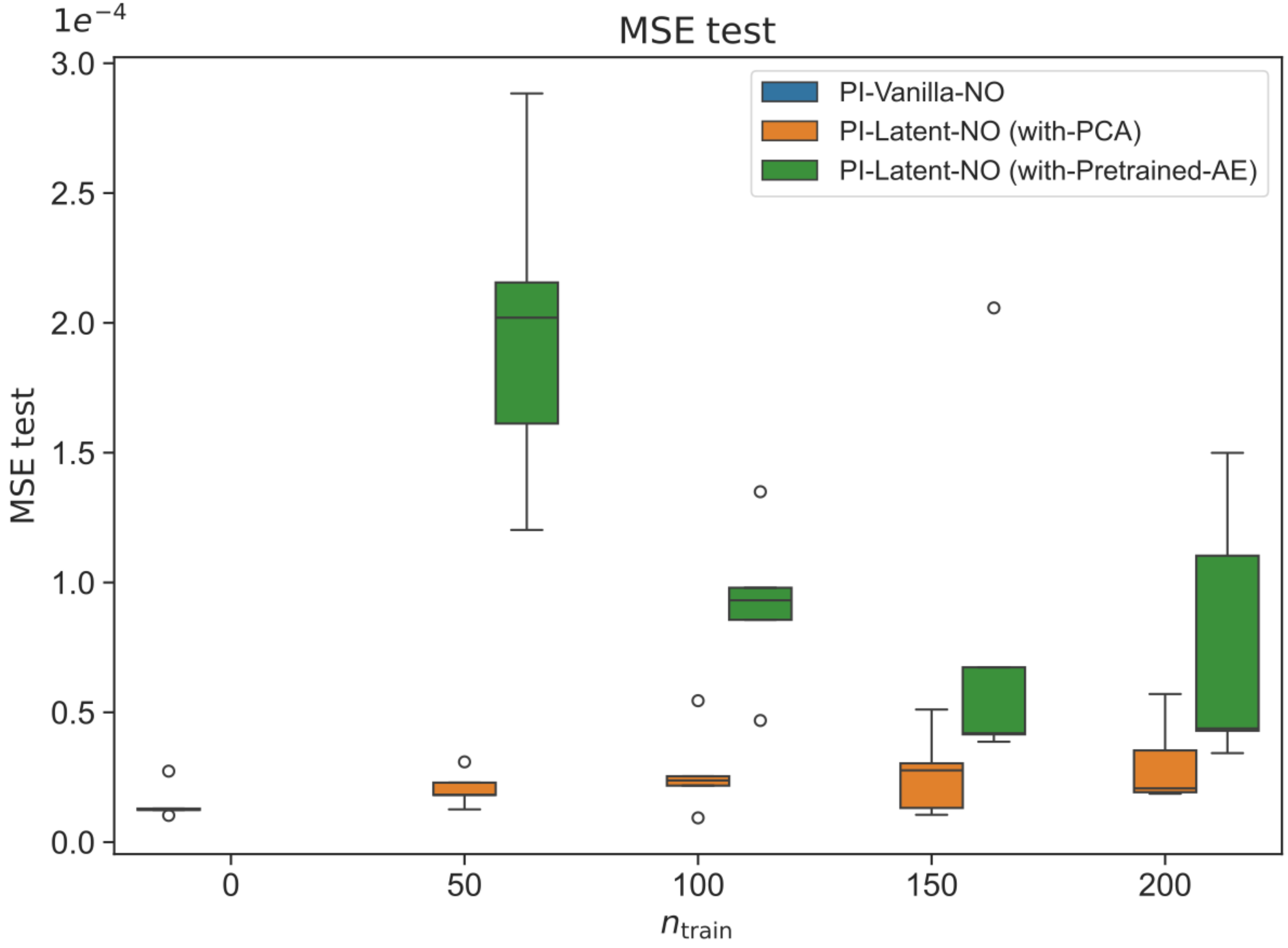


Prediction

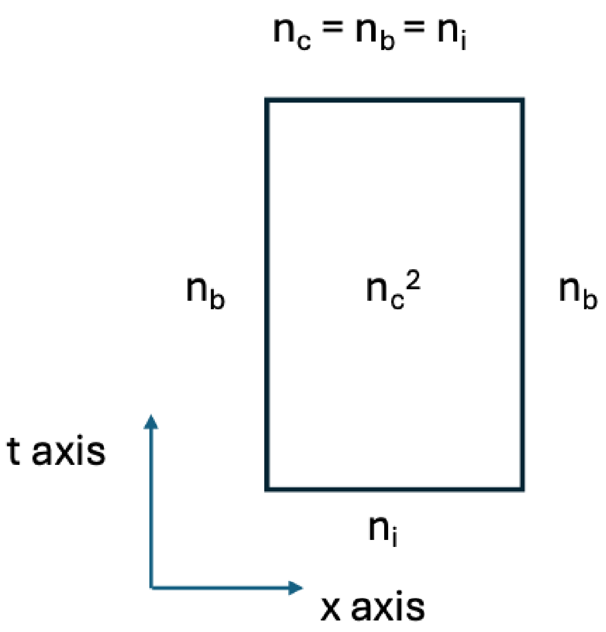
Error

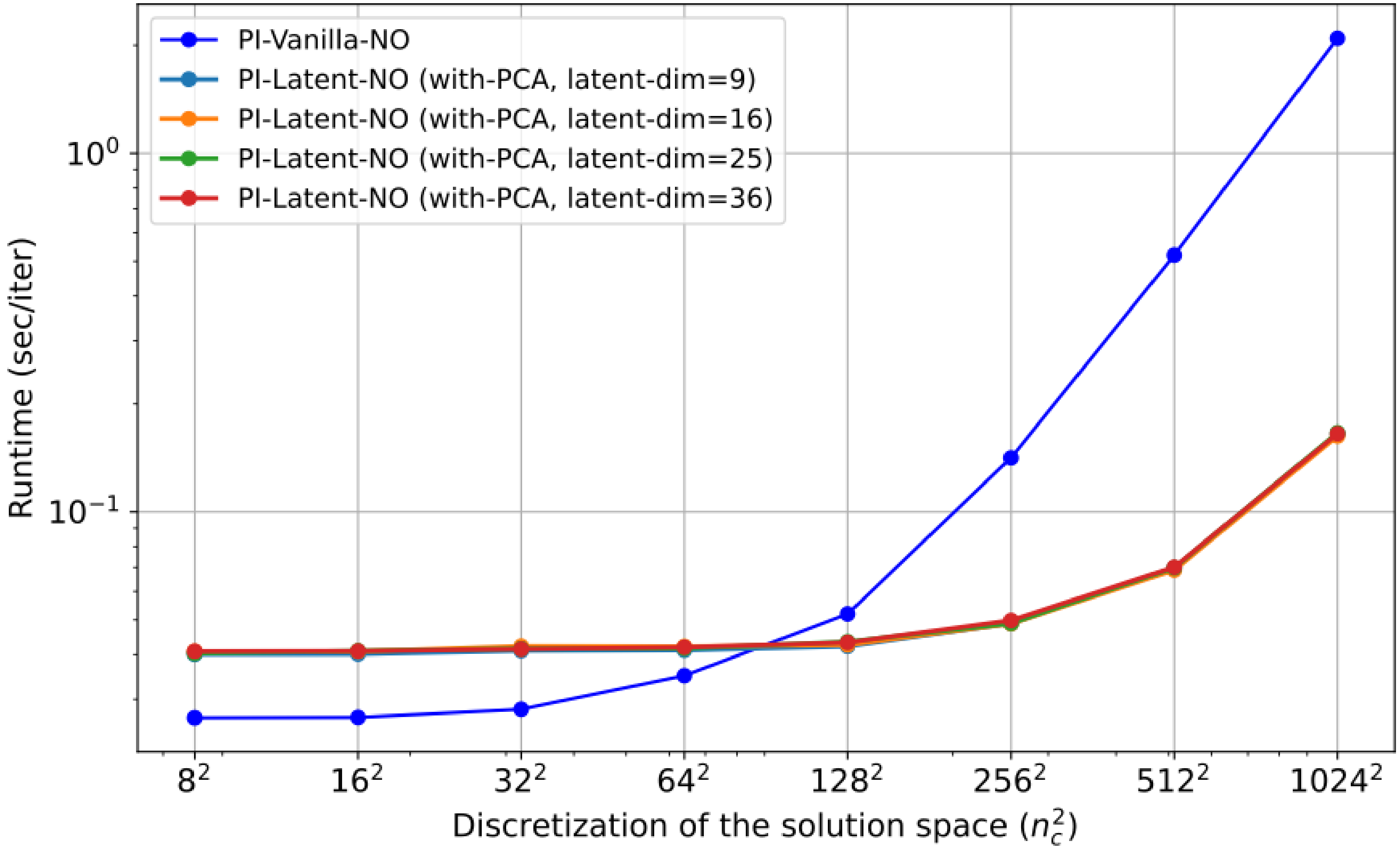


Accuracy Comparison

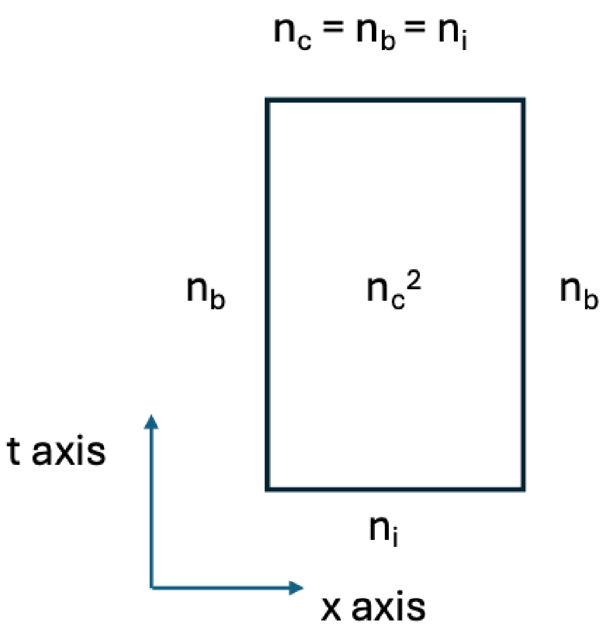


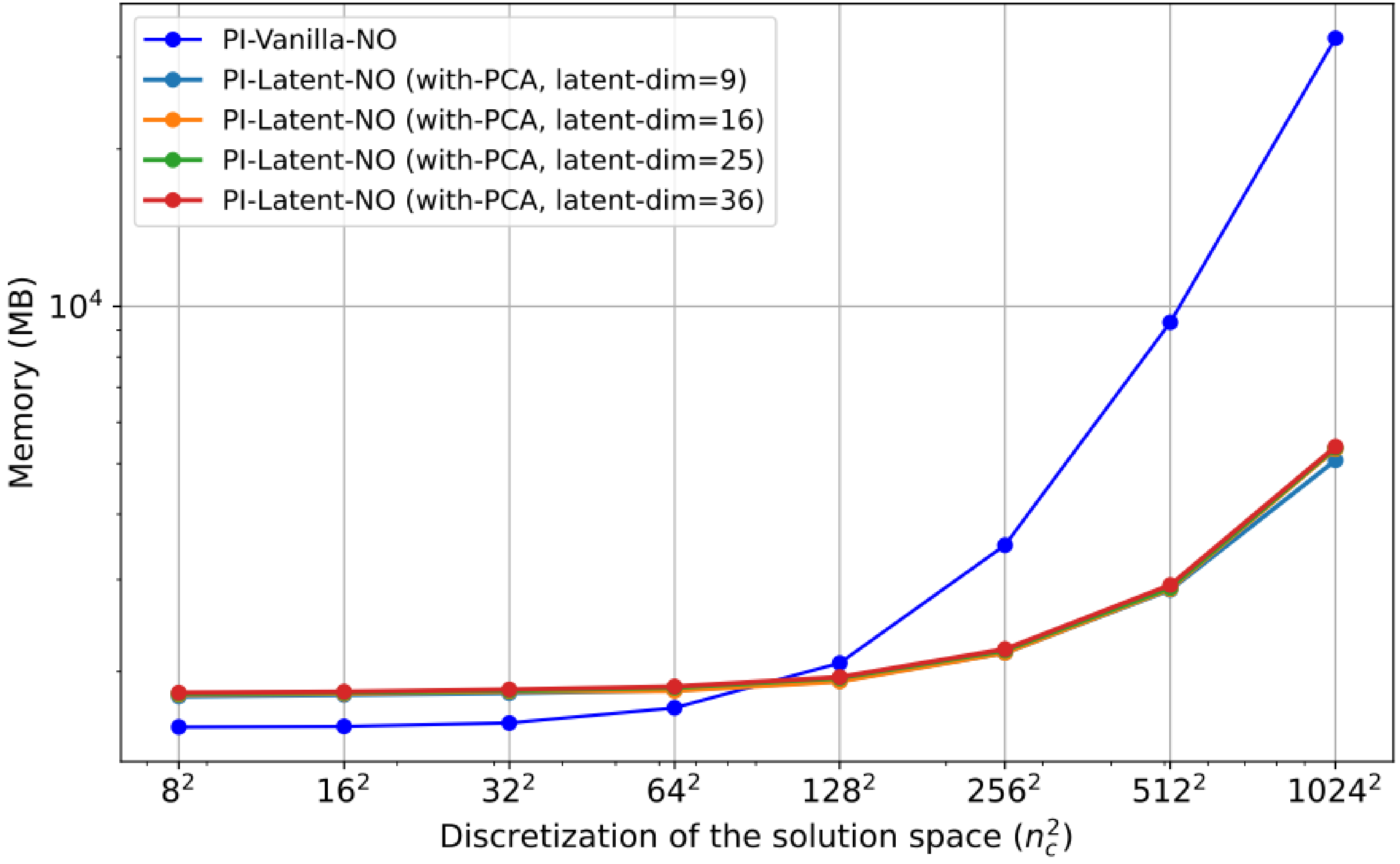
Runtime Scaling





Memory Scaling





Learning the heat equation with PI-Latent Neural Operator

Operator $\mathcal{G} : \mathcal{X} \rightarrow \mathcal{Y}$

$\mathcal{G}_\theta : \mathcal{X} \rightarrow \mathcal{Y}, \quad \theta \in \Theta$

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

$$\frac{\partial T}{\partial t} = D\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right) + \mathrm{s}(\mathrm{x}, \mathrm{y}, \mathrm{a}) \quad \forall\, (x, y) \in [-L, L]^2, t \in [0, T]$$

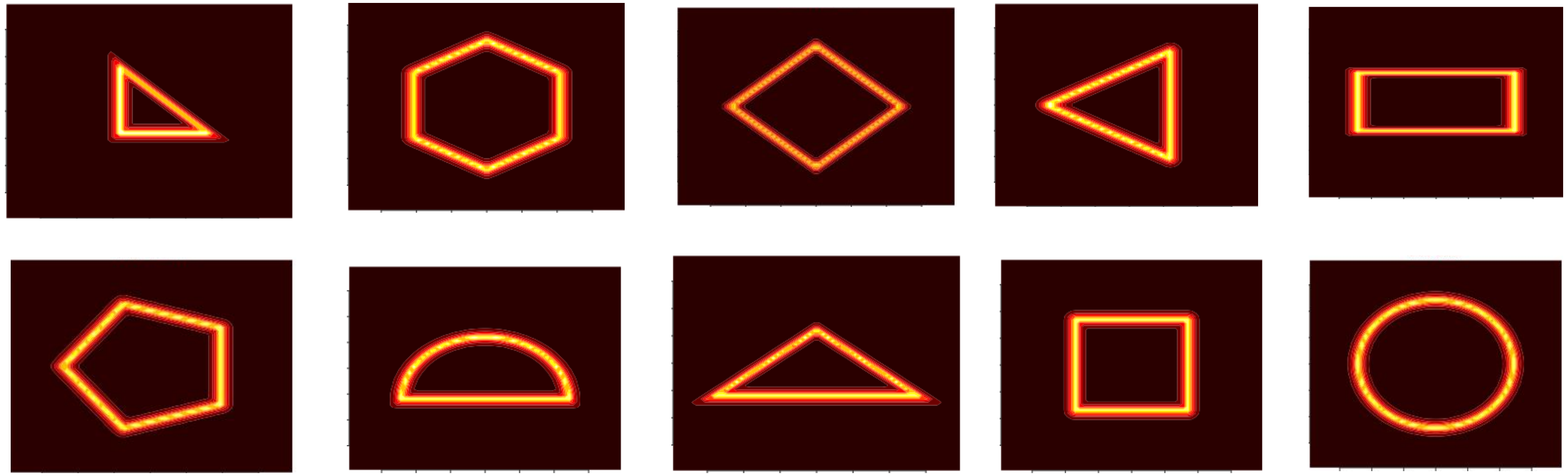
$L = 2, T = 1, D = 1$, and $u = u(t, x, y)$

$u_{\partial\Omega} = 0 \quad \forall\, (x, y) \in \partial\Omega$ and $u(0, x, y) = 0$

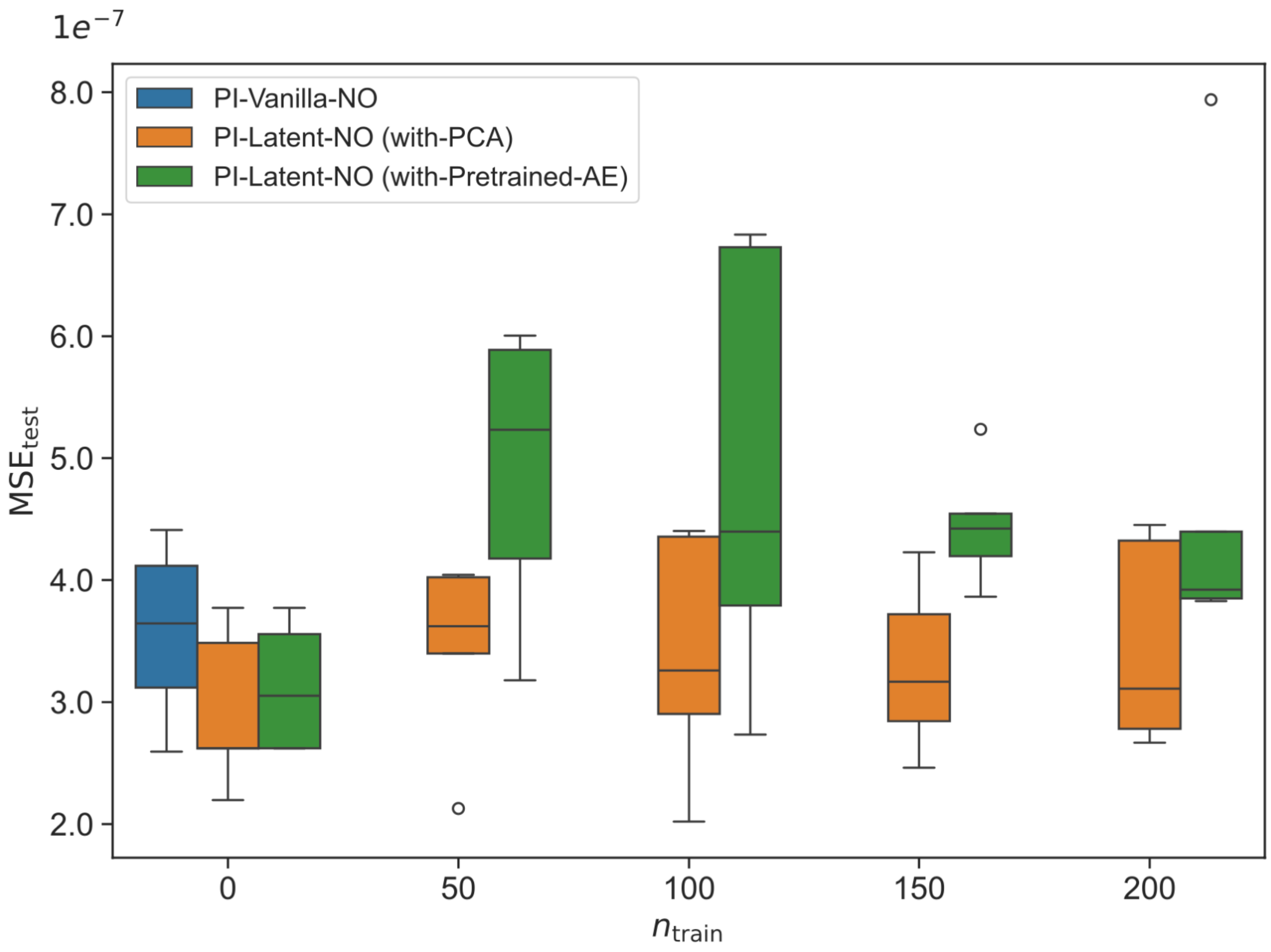
$\mathcal{X}$: the source field image, $\mathrm{s}(\mathrm{x}, \mathrm{y}, \mathrm{a})$ – varying geometries and the filament intensity

$\mathcal{Y}$: the temperature field, $T(t, x, y)$

Considered Geometries

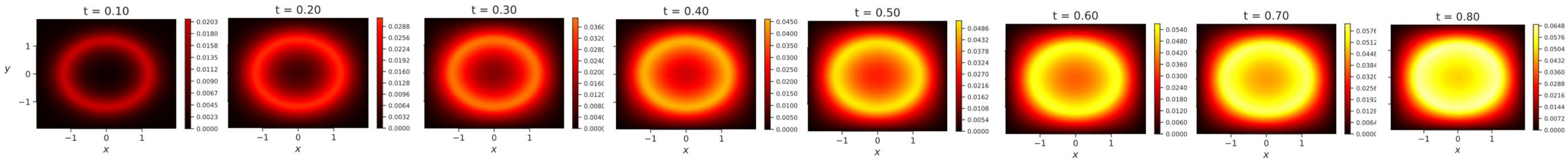


Error Comparison

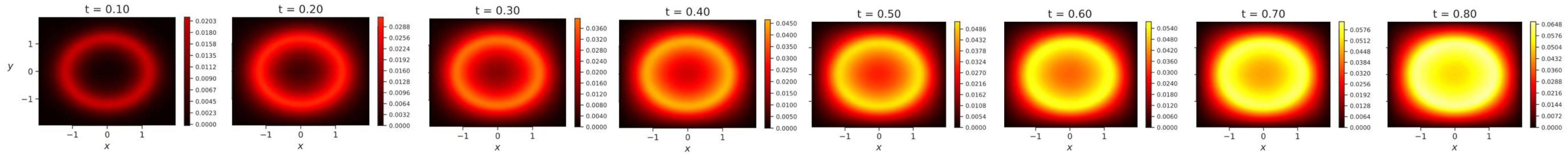


Results Comparison

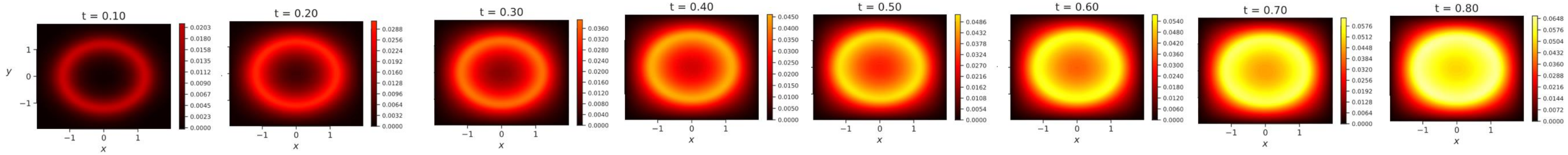
Ground Truth



Vanilla PI-DeepONet

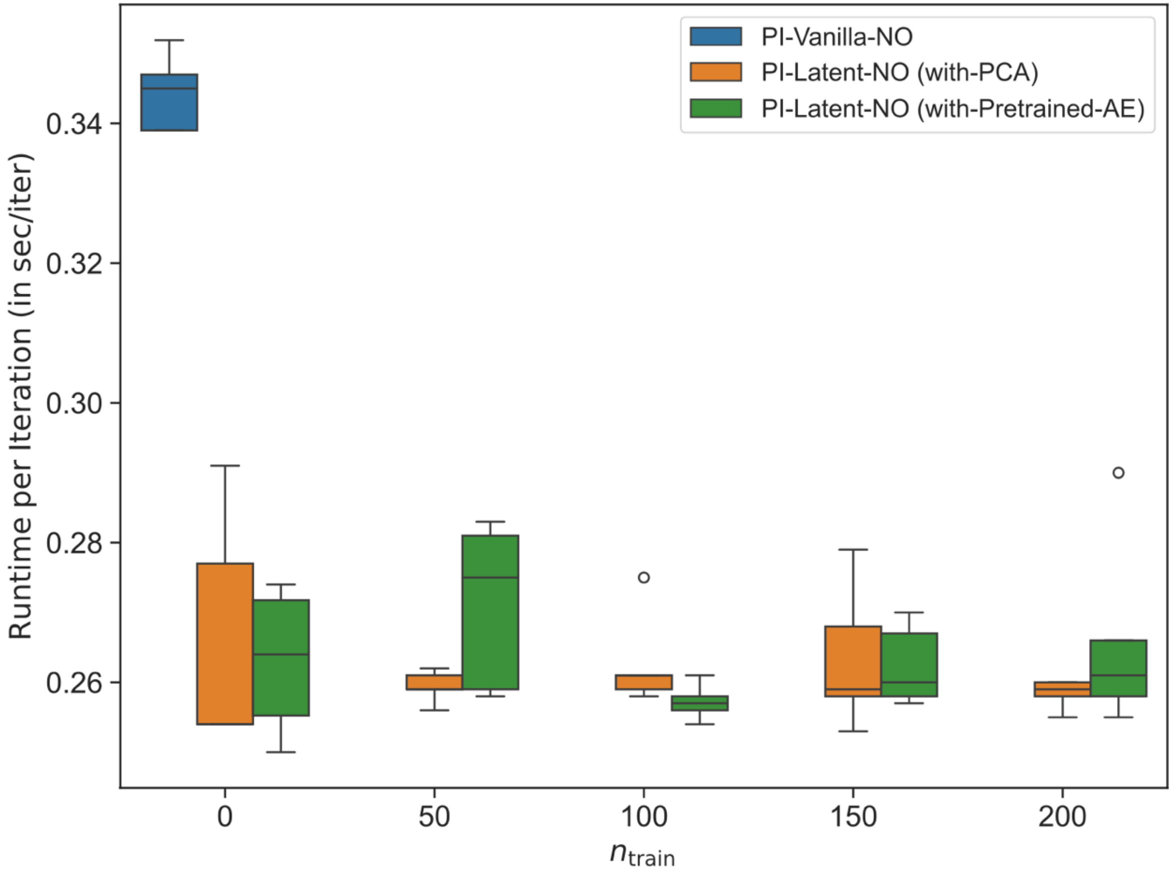


PI-Latent DeepONet with $\#N_{train}$=0

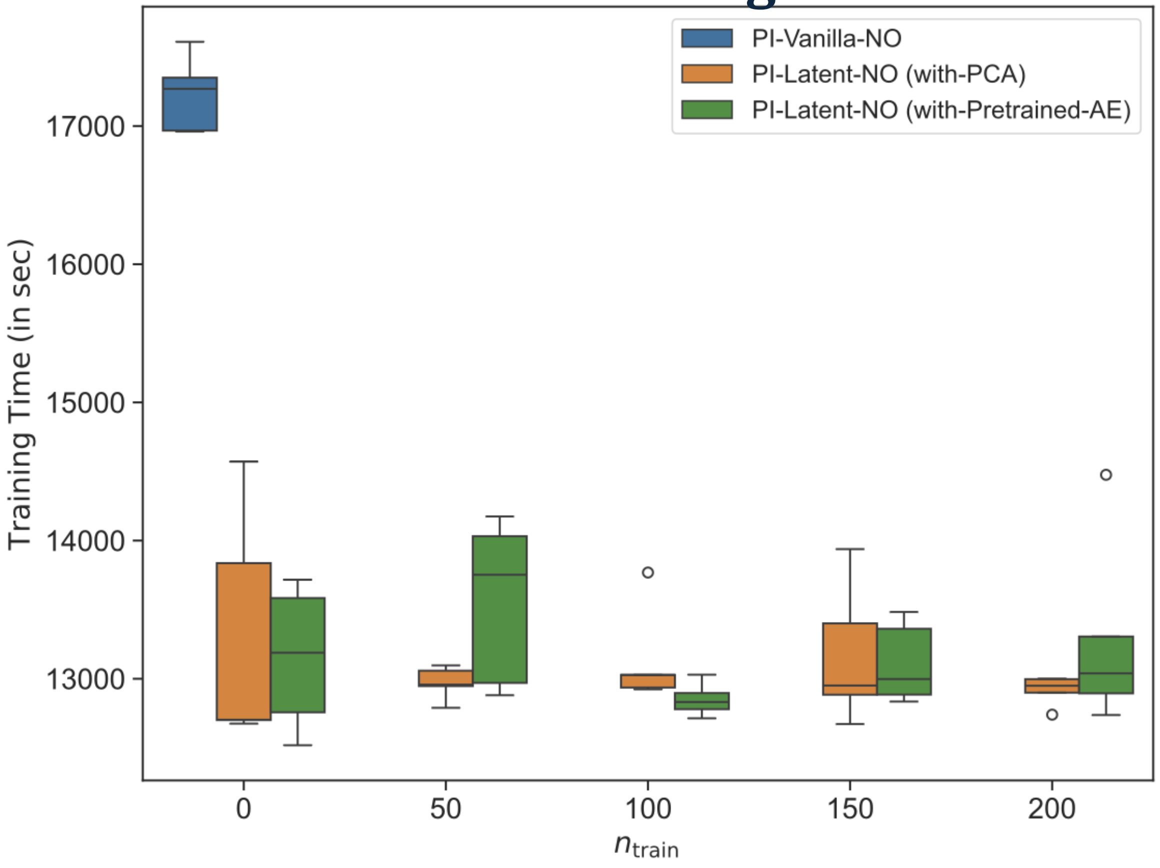


Runtime Comparison

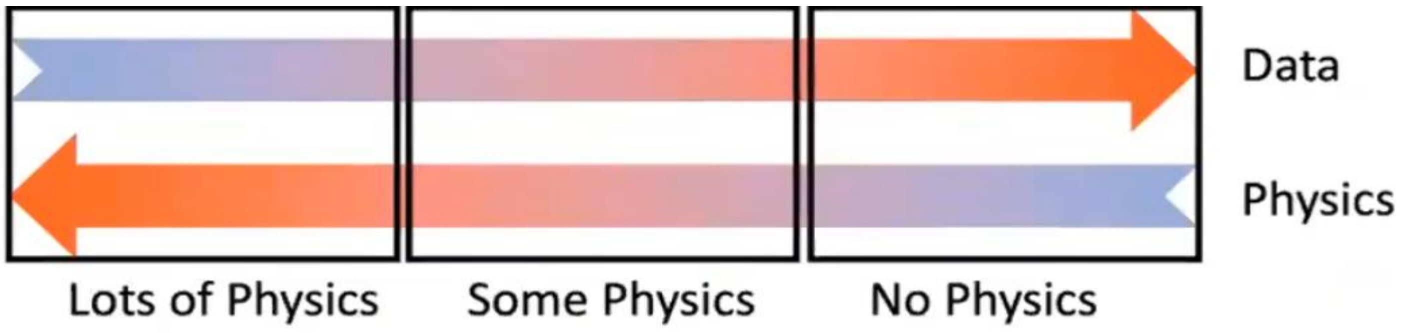
Per Iteration



Per Training Time

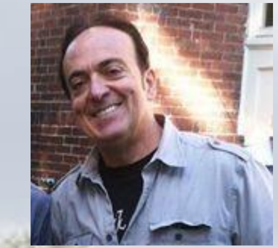


Key Takeaways



- These methods have a niche in real world problems, where partially physics in known and some measurements of quantities of interest are available.
- These methods are best implemented when complemented with mature methods like FEM.
  - ➢ Heterogenous multiscale modeling
  - ➢ Hybrid fast solvers
- These frameworks offer a possibility to seamlessly blend data and physics.

Acknowledgement

Thank you!

This work is funded by U.S. DEPARTMENT OF ENERGY