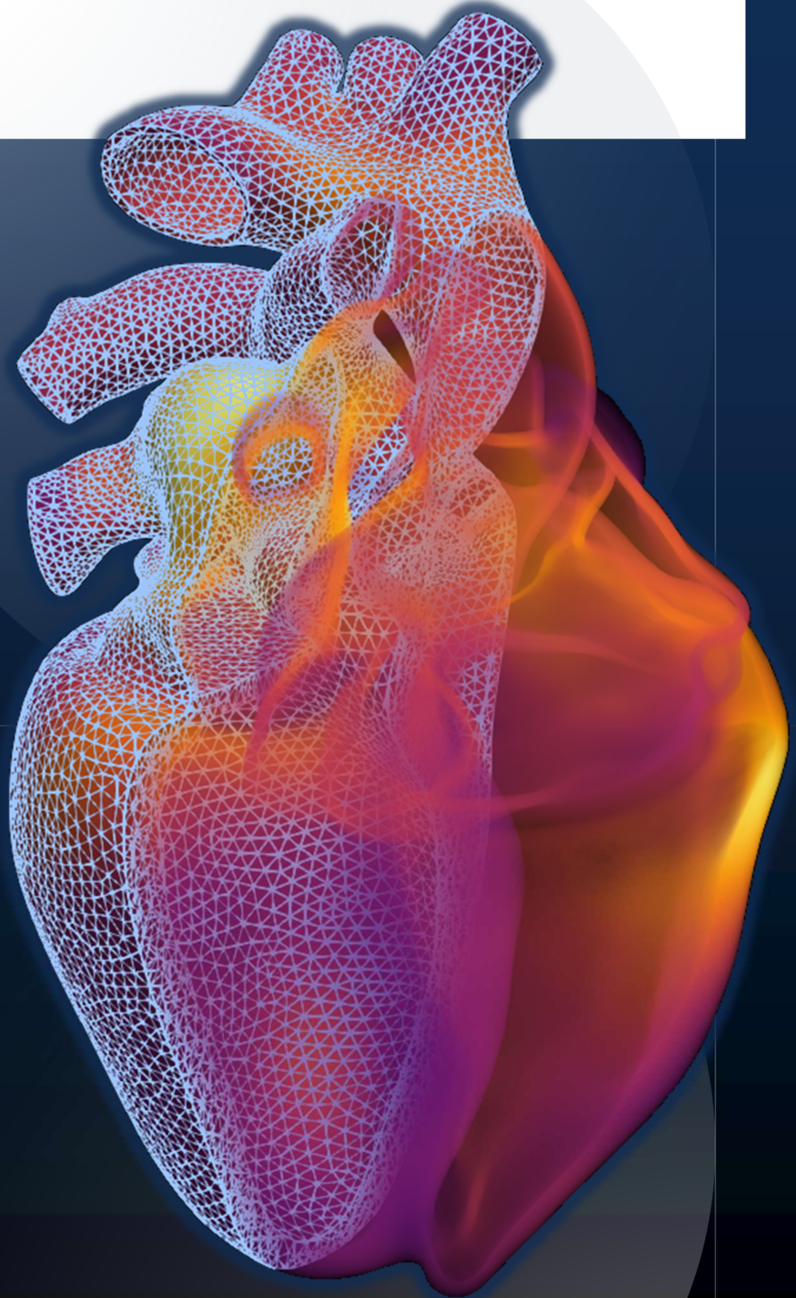
Friedrich-Alexander-Universität
Research Center for
Mathematics of Data | MoD

Machine Learning and PDEs (MLPDES25) Workshop
Erlangen, Bavaria (Germany)
April 28-30, 2025

Discovering the hidden low-dimensional dynamics of time-dependent PDEs with Latent Dynamics Networks



Francesco Regazzoni

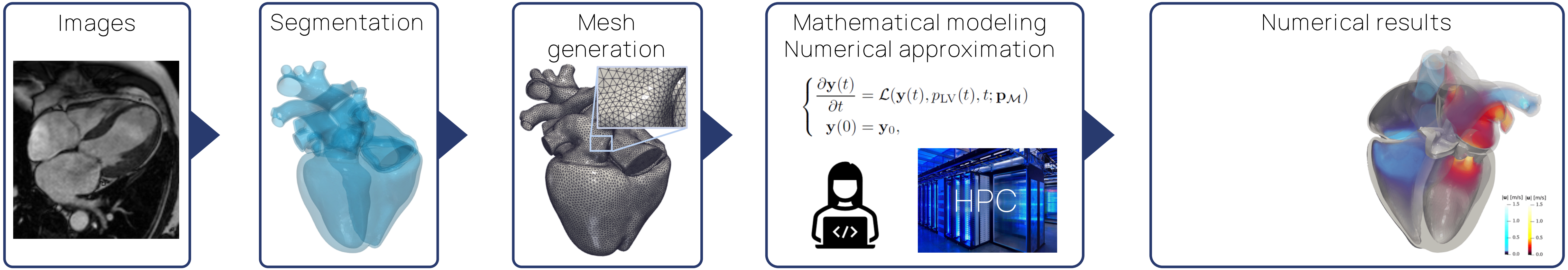
MOX – Department Of Mathematics
Politecnico di Milano (Italy)

POLITECNICO MILANO 1863

POLITECNICO MILANO 1863
DIPARTIMENTO DI MATEMATICA
DEPARTMENT OF EXCELLENCE 2023-2027

Motivations

Traditional modeling and simulation approach



Zingaro et al, JCP (2024), Fedele et al. CMAME (2024)

Preprocessing steps (including meshing) are time-consuming and require often manual interventions

Computational costs are often prohibitive

Example: cardiac electromechanics on 1.34 M Tetrahedra, 1152 cores on GALILEO100@Cineca:

- Takes 4 hours
- Costs about 2000 €
- Consumes 100 kWh of energy, producing 35 kg of CO_2

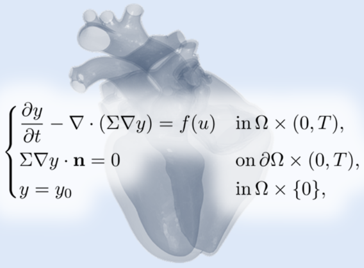
This hinders many-query procedures, such as Uncertainty Quantification or Global Sensitivity Analysis

Building surrogate models through Operator Learning

Surrogate models

Physics-based model

$\mathcal{M} : U \to Y$



$$\begin{cases} \frac{\partial y}{\partial t} - \nabla \cdot (\Sigma \nabla y) = f(u) & \text{in } \Omega \times (0,T), \\ \Sigma \nabla y \cdot \mathbf{n} = 0 & \text{on } \partial\Omega \times (0,T), \\ y = y_0 & \text{in } \Omega \times \{0\}, \end{cases}$$

$u \in U$

input: parameters, functions, shapes, ...

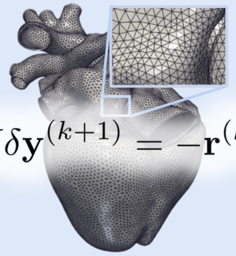
$y \in Y$

output: scalars, functions, ...

Numerical Approximation

High-fidelity solver

$\mathcal{M}_h : U \to Y_h$



$$J\delta\mathbf{y}^{(k+1)} = -\mathbf{r}^{(k)}$$

$u^1 \in U \rightarrow$ $\rightarrow y_h^1 = \mathcal{M}_h(u^1)$

$u^2 \in U \rightarrow$ $\rightarrow y_h^2 = \mathcal{M}_h(u^2)$

$\vdots$

$u^N \in U \rightarrow$ $\rightarrow y_h^N = \mathcal{M}_h(u^N)$

training data generation

Operator Learning

$\{(u^i, y_h^i)\}_{i=1}^N \subset U \times Y$ training dataset

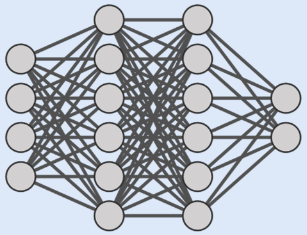
$\mathcal{H} \subset Y^U = \{\widetilde{\mathcal{M}} : U \to Y\}$ hypothesis space

$$\mathcal{M}_{SM}^* = \underset{\mathcal{M}_{\mathrm{SM}} \in \mathcal{H}}{\operatorname{argmin}} \sum_{i=1}^{N} \| y_h^i - \mathcal{M}_{\mathrm{SM}}(u^i) \|_Y^2$$

Training

Surrogate model

$\mathcal{M}_{\mathrm{SM}} : U \to Y$



Approximate
(typical accuracy: $10^{-2} - 10^{-3}$)

Fast
(often: milliseconds-seconds)

surrogates

Surrogate models: is it worth it?

Real-time solutions

e.g. robotic-assisted surgery, real-time control of industrial plants, …

Many-query scenarios

Suppose we need n_{query} evaluations of the data-to-solution map:

Using the surrogate model: Using the high-fidelity solver:

$$\underbrace{N\,T_{\mathrm{HF}} + T_{\mathrm{training}}}_{\text{offline phase}} + \underbrace{n_{\mathrm{query}} T_{\mathrm{SM}}}_{\text{online phase}} \quad < \quad n_{\mathrm{query}} T_{\mathrm{HF}}$$

We need: $N \ll n_{\mathrm{query}}$ ➡

The surrogate model must generalize well for unseen inputs even when using a small number of training samples

PART I

Operator Learning for time-dependent problems

PART II

Operator Learning in variable domain

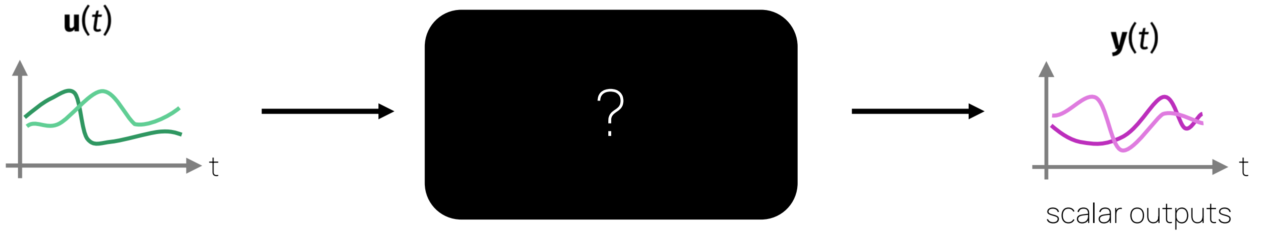
PART I

Operator Learning for time-dependent problems

PART II

Operator Learning in variable domain

Operator learning for of time-dependent problems



- Displacement of a control point
- Generated power of a plant
- Revenues of a company
- Number of infected people
- Wrist blood pressure
- ...

- Control of an engine
- Wind strength
- Price of an asset
- Strength of a social measure
- Dose of a farmakon
- ...

- Displacement field of a body
- Temperature field in a pump
- Concentration of a pollutant
- Spread of an epidemic
- Blood velocity in an aneurysm
- ...

Model Learning

1. Training input-output pairs:

$\hat{\mathbf{u}}_j \in \mathcal{U} = \mathcal{C}^0([0,T];U)$ input space, where $U \subset \mathbb{R}^{N_u}$

$\hat{\mathbf{y}}_j \in \mathcal{Y} = \mathcal{C}^0([0,T];Y)$ output space, where $Y \subset \mathbb{R}^{N_y}$

$j = 1, \dots, N_s$

2. Hypothesis space (class of candidate models):

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), & t \in (0,T] \\ \mathbf{x}(0) = \mathbf{x}_0 \end{cases}$$

$$\mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t)), \qquad t \in (0,T],$$

$n \in \mathbb{R}$

$\mathbf{f} \in \widehat{\mathcal{F}} \subset \mathcal{F}_n := \{\mathbf{f} \in \mathcal{C}^0(\mathbb{R}^n \times U; \mathbb{R}^n)$, Lipschitz cont. in $\mathbf{x}$ uniformly in $\mathbf{u}\}$

$\mathbf{g} \in \widehat{\mathcal{G}} \subset \mathcal{G}_n := \mathcal{C}^0(\mathbb{R}^n; Y)$

$\mathbf{x}_0 \in \widehat{\mathcal{X}} \subset \mathcal{X}_n \equiv \mathbb{R}^n$

This system of ODEs uniquely identifies a map: $\varphi_{\mathbf{f},\mathbf{g},\mathbf{x}_0} : \mathcal{U} \to \mathcal{Y}$

$$\Phi^{\widehat{\mathcal{F}},\widehat{\mathcal{G}},\widehat{\mathcal{X}}} = \left\{ \varphi_{\mathbf{f},\mathbf{g},\mathbf{x}_0} \in \Phi \text{ s.t. } \mathbf{f} \in \widehat{\mathcal{F}}, \mathbf{g} \in \widehat{\mathcal{G}}, \mathbf{x}_0 \in \widehat{\mathcal{X}} \right\}$$

Theorem (R., Dede', Quarteroni, *JCP* 2019)

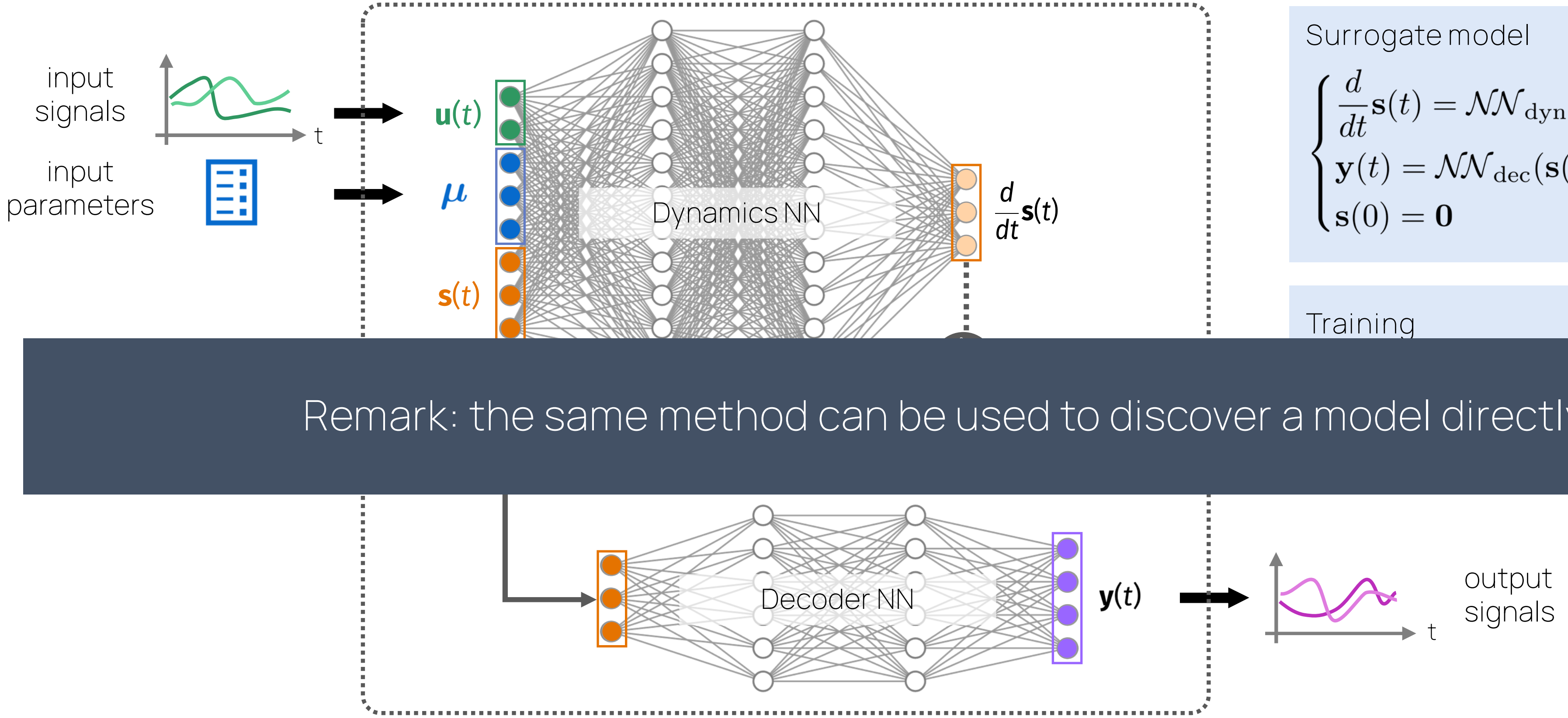
The set of ODE models whose r.h.s. and observation function are Neural Networks is dense in the set of all ODE models w.r.t. the L∞ norm.

3. Empirical risk minimzation (training):

$$\varphi^* = \underset{\varphi \in \Phi^{\widehat{\mathcal{F}},\widehat{\mathcal{G}},\widehat{\mathcal{X}}}}{\operatorname{argmin}} \frac{1}{2} \sum_{j=1}^{N_s} \int_0^T |\hat{\mathbf{y}}_j(t) - (\varphi \hat{\mathbf{u}}_j)(t)|^2 dt$$

F. Regazzoni, L. Dede', A. Quarteroni, Journal of Computational Physics (2019)

Model Learning: architecture and training



Surrogate model

$$\begin{cases} \dfrac{d}{dt}\mathbf{s}(t) = \mathcal{NN}_{\mathrm{dyn}}(\mathbf{s}(t), \mathbf{u}(t), \boldsymbol{\mu}; \mathbf{w}_{\mathrm{dyn}}) & t \in (0, T) \\ \mathbf{y}(t) = \mathcal{NN}_{\mathrm{dec}}(\mathbf{s}(t); \mathbf{w}_{\mathrm{dec}}) & t \in (0, T) \\ \mathbf{s}(0) = \mathbf{0} \end{cases}$$

Training

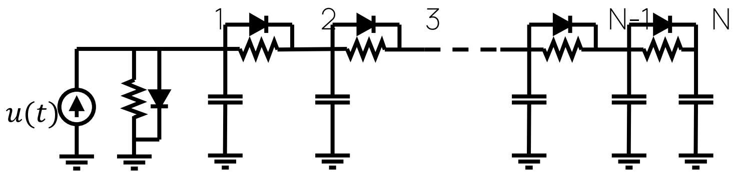
Remark: the same method can be used to discover a model directly from data!

- The recurrent part is formally similar to an Neural ODE. However the dynamics of s(t) is not known a-priori.
- The latent state s(t) provides a compact encoding of the high-fidelity model state. However, the mapping is never explicitly constructed!
- The two ANNs are trained simultaneously: the training algorithm discovers a latent space to
  - Predict the system dynamics
  - Reconstruct the output

F. Regazzoni, L. Dede', A. Quarteroni, Journal of Computational Physics (2019)

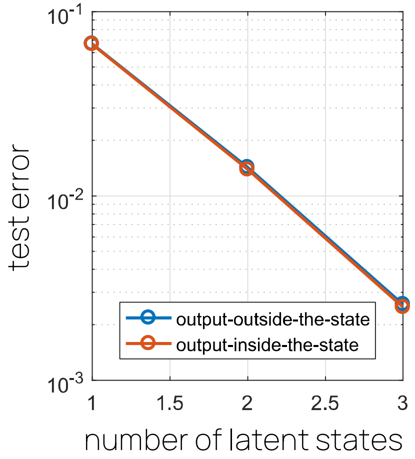
Benchmark test cases

Systems of nonlinear ODEs

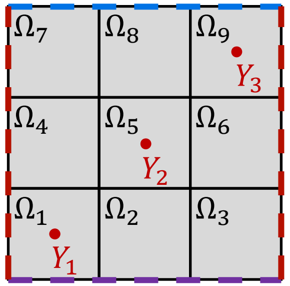


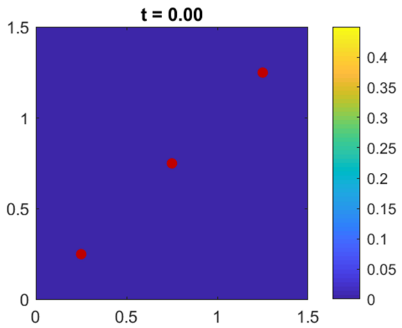
$$
\begin{cases}
\dot{v}_1(t) = -2\,v_1(t) + v_2(t) + 2 - e^{40\,v_1(t)} - e^{40(v_1(t)-v_2(t))} + u(t)\\
\dot{v}_i(t) = -2\,v_i(t) + v_{i-1}(t) + v_{i+1}(t) + e^{40(v_{i-1}(t)-v_i(t))} - e^{40(v_i(t)-v_{i+1}(t))},\\
\dot{v}_N(t) = -v_N(t) + v_{N-1}(t) - 1 + e^{40(v_{N-1}(t)-v_N(t))},
\end{cases}
$$

$$y(t) = v_1(t)$$



Parabolic PDEs

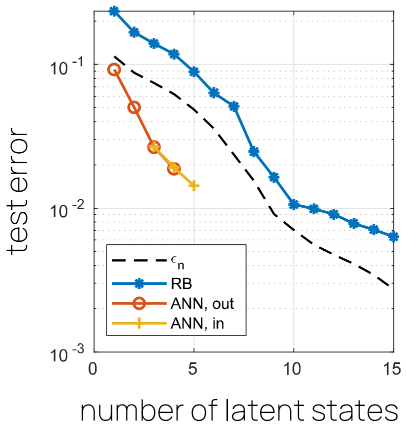




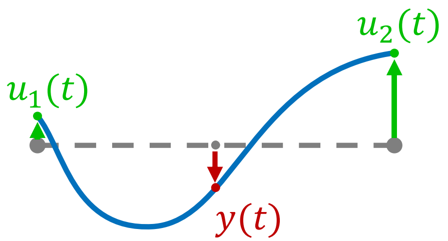
$$
\begin{cases}
\frac{\partial}{\partial t}\psi(\mathbf{p},t) - \nabla\cdot\left(k(\mathbf{p},\mathbf{u}(t))\,\nabla\psi(\mathbf{p},t)\right) = 0 & \text{for } \mathbf{p}\in\Omega,\ t>0\\
k(\mathbf{p},\mathbf{u}(t))\,\nabla\psi(\mathbf{p},t)\cdot\mathbf{n} = 0 & \text{for } \mathbf{p}\in\Gamma_w,\ t>0\\
k(\mathbf{p},\mathbf{u}(t))\,\nabla\psi(\mathbf{p},t)\cdot\mathbf{n} = 1 & \text{for } \mathbf{p}\in\Gamma_b,\ t>0\\
\psi(\mathbf{p},t) = 0 & \text{for } \mathbf{p}\in\Gamma_t,\ t>0\\
\psi(\mathbf{p},0) = 0 & \text{for } \mathbf{p}\in\Omega.
\end{cases}
$$

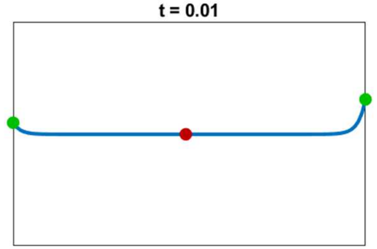
$$k(\mathbf{p},\mathbf{u}) = \sum_{i=1}^{9} u_i\,\mathbb{1}_{\Omega_i}(\mathbf{p})$$

$$y_i(t) = |Y_i|^{-1}\int_{Y_i}\psi(\mathbf{p},t)d\mathbf{p}$$



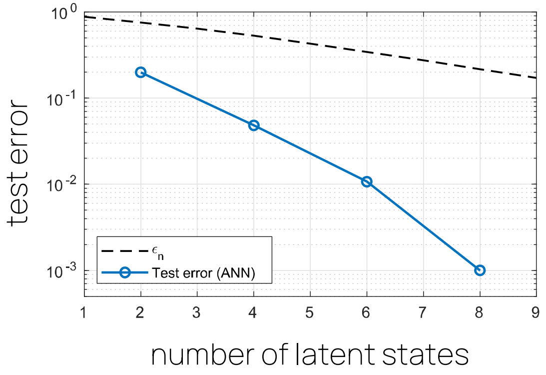
Hyperbolic PDEs





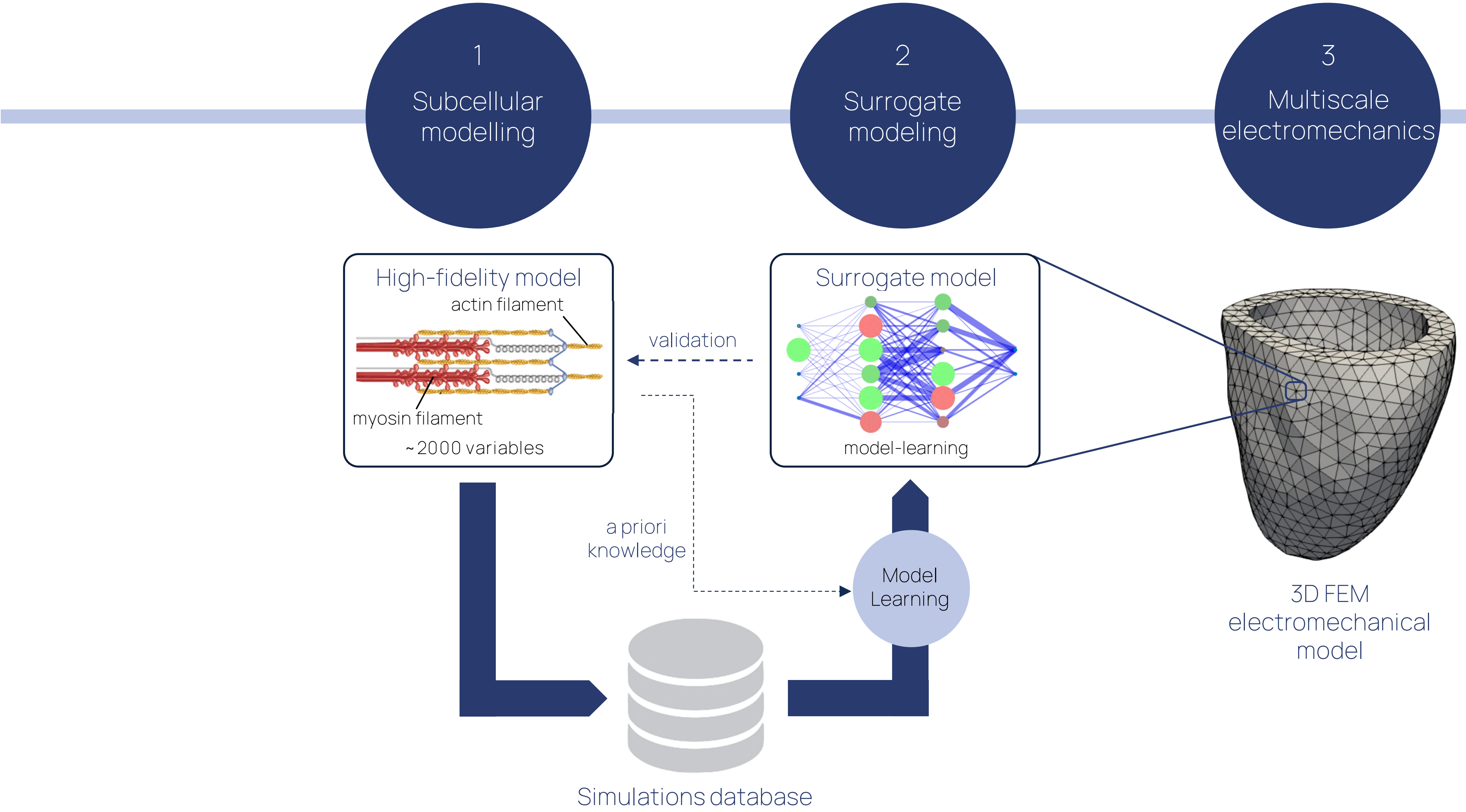
$$
\begin{cases}
\frac{\partial^2}{\partial t^2}\psi(z,t) - c^2\frac{\partial^2}{\partial z^2}\psi(z,t) = 0 & \text{for } z\in(0,L),\ t>0\\
\psi(0,t) = u_1(t) & \text{for } t>0\\
\psi(L,t) = u_2(t) & \text{for } t>0\\
\psi(z,0) = 0 & \text{for } z\in(0,L)\\
\frac{\partial}{\partial t}\psi(z,0) = 0 & \text{for } z\in(0,L),
\end{cases}
$$

$$y(t) = \psi\left(\tfrac{L}{2},t\right)$$



Applications to cardiac modeling

First application: learning the microscopic scale

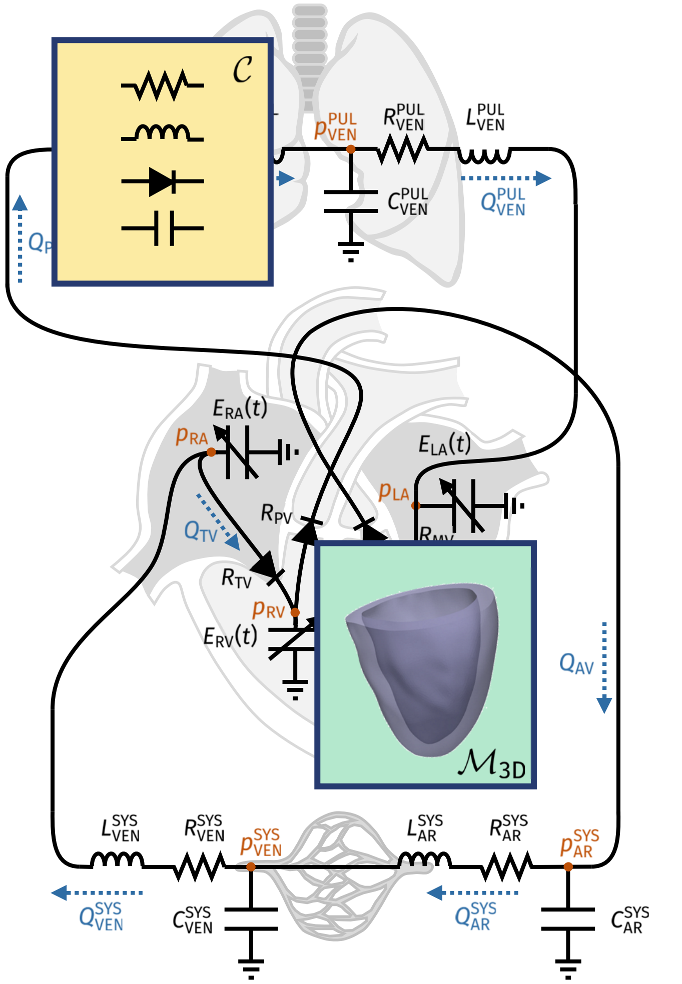


Results:

- ✓ 10^{-3} testing accuracy
- ✓ 400 x speedup
- ✓ 1000 x memory saving

F. Regazzoni, L. Dede', A. Quarteroni Computer Methods in Applied Mechanics and Engineering (2020)

Second application: learning the coupled 3D model



$\mathbf{F} = \mathbf{I} + \nabla \mathbf{d}$

Electrophysiology

$$\begin{cases} J\chi_m \left[C_m \dfrac{\partial v}{\partial t} + \mathcal{I}^{\mathrm{ion}}(v, \mathbf{w}) \right] - \nabla \cdot (J\mathbf{F}^{-1}\mathbf{D}\mathbf{F}^{-T}\nabla v) = I^{\mathrm{app}} & \text{in } \Omega_0 \times (0,T) \\ \dfrac{\partial \mathbf{w}}{\partial t} = \mathbf{h}(v, \mathbf{w}) & \text{in } \Omega_0 \times (0,T) \\ + \text{ boundary conditions} \end{cases}$$

$[\mathrm{Ca}^{2+}]_i$

$SL = SL_0 \sqrt{\mathcal{I}_{4f}}$

Active force generation

$$\frac{\partial \mathbf{r}}{\partial t} = \mathbf{\Phi}\left(\mathbf{r}, [\mathrm{Ca}^{2+}]_i, SL, \frac{\partial SL}{\partial t}\right) \quad \text{in } \Omega_0 \times (0,T)$$

$$\mathbf{P}_a = T_a(\mathbf{r}) \frac{\mathbf{F}\mathbf{f}_0 \otimes \mathbf{f}_0}{\sqrt{\mathcal{I}_{4,f}}}$$

Tissue mechanics

$$\begin{cases} \rho \dfrac{\partial^2 \mathbf{d}}{\partial t^2} - \nabla \cdot \mathbf{P} = 0 & \text{in } \Omega_0 \times (0,T) \\ \mathbf{P} = \dfrac{\partial \mathcal{W}}{\partial \mathbf{F}} + \mathbf{P}_a & \text{in } \Omega_0 \times (0,T) \\ \mathbf{P}\mathbf{N} = -p_{\mathrm{LV}} J \mathbf{F}^{-T}\mathbf{N} & \text{on } \Gamma_0^{\mathrm{endo}} \times (0,T) \\ + \text{ other boundary conditions} \end{cases}$$

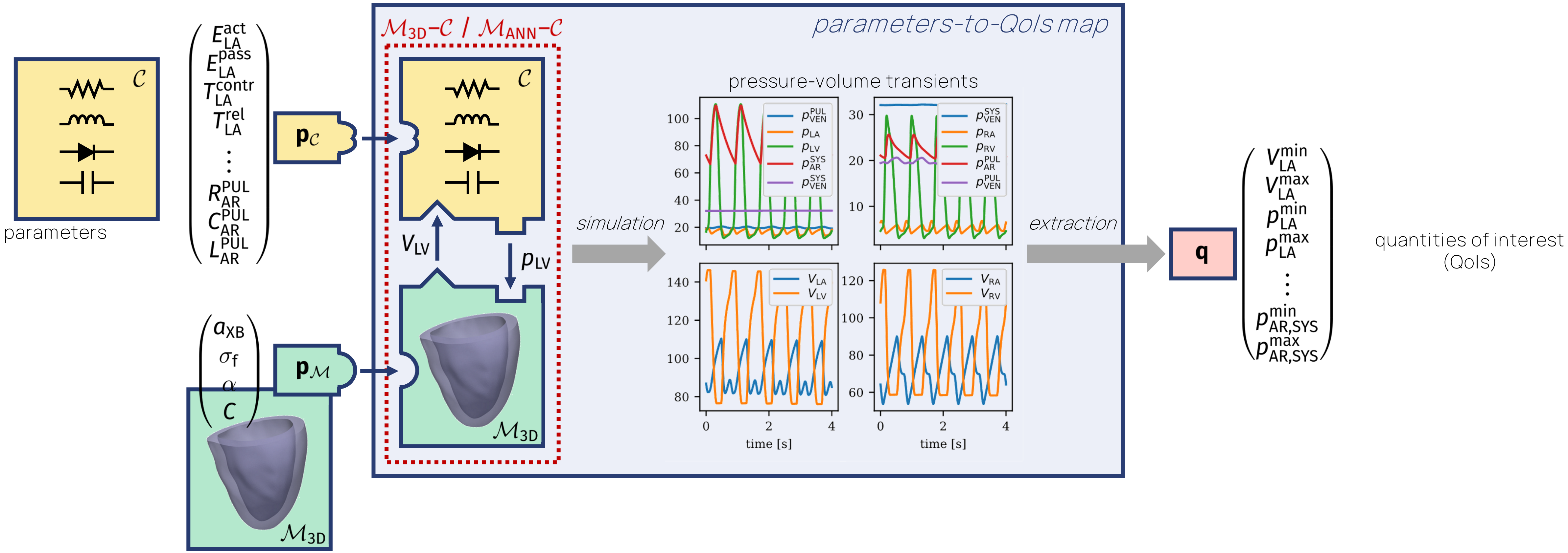
$$V_{LV}^{3D}(\mathbf{d}) = V_{LV}^{0D}(\mathbf{c})$$

Blood circulation

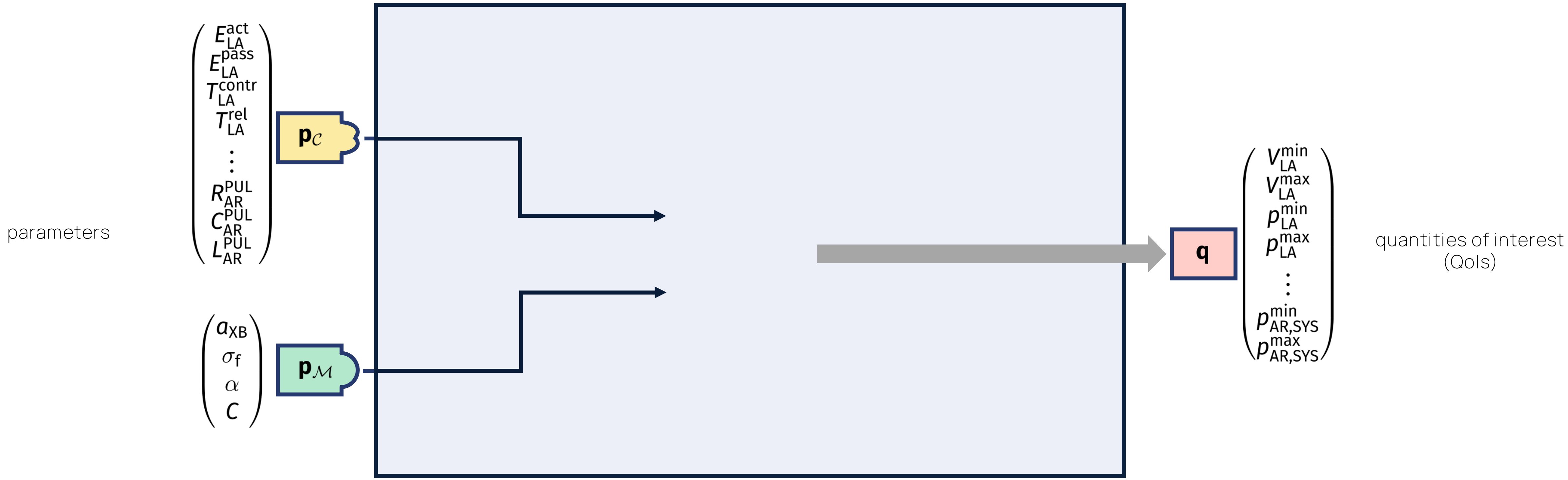
$$\frac{\partial \mathbf{c}}{\partial t} = \mathbf{H}(t, \mathbf{c}, p_{\mathrm{LV}}) \quad t \in (0,T)$$

F. Regazzoni, M. Salvador, P. C. Africa, et al. Journal of Computational Physics (2022)
M. Fedele, R. Piersanti, F. Regazzoni, et al. Computer Methods in Applied Mechanics and Engineering (2022)

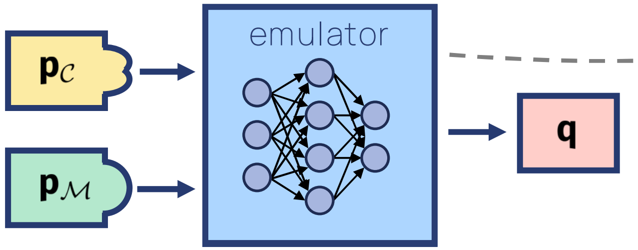
Second application: learning the coupled 3D model



Second application: learning the coupled 3D model



Standard approach: surrogating the whole parameter to QoI map

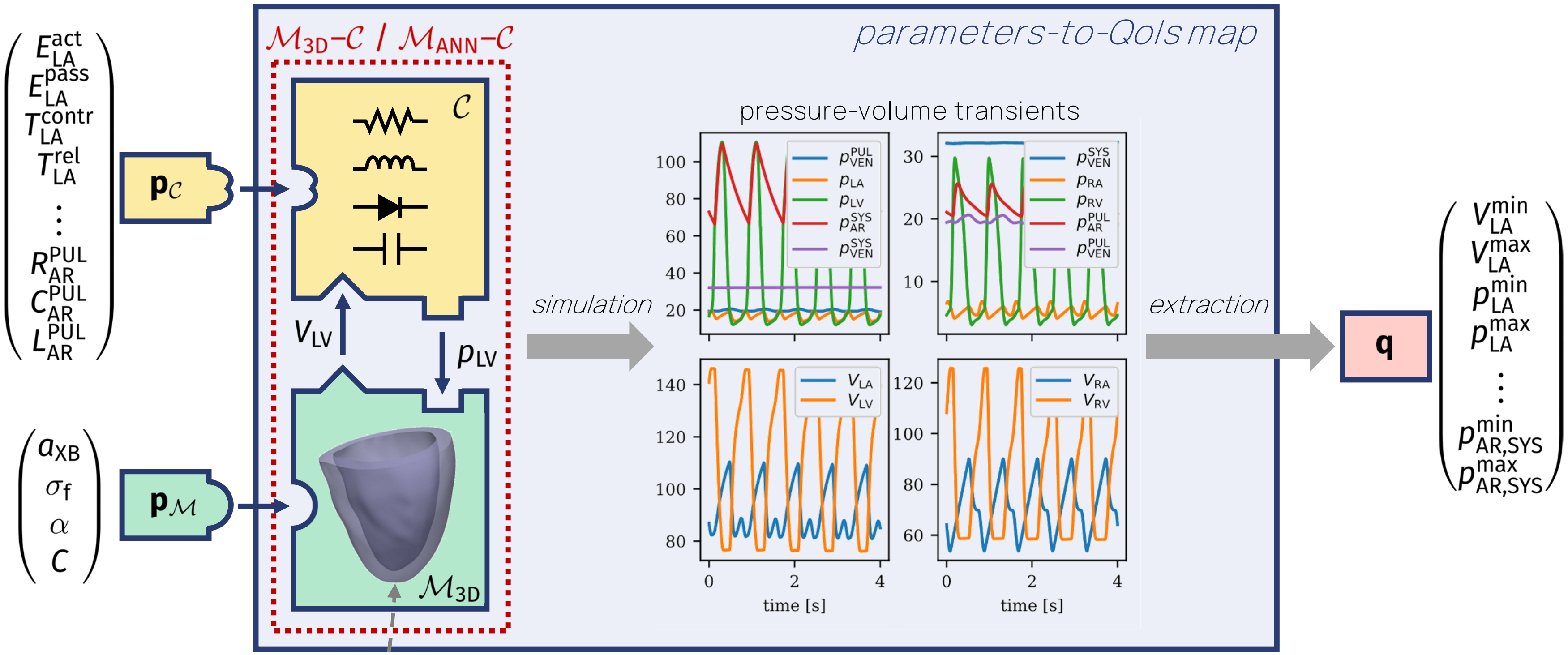


- Computationally inexpensive
- Large parameter space to explore (need for large training datasets)
- Only scalar QoIs (no transients)
- Fixed time horizon (cannot «extrapolate in time»)

P. Di Achille, A. Harouni, S. Khamzin, O. Solovyova, et al., Frontiers in Physiology 9 (2018)
S. Longobardi, A. Lewalle, S. Coveney, et al., Phil. Transactions of the Royal Society (2020)

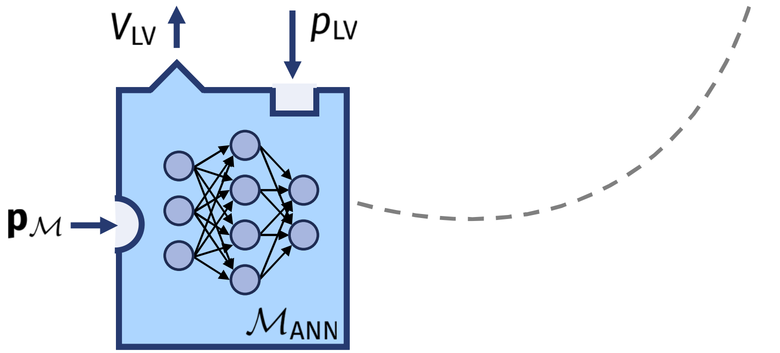
Second application: learning the coupled 3D model

parameters



quantities of interest (QoIs)

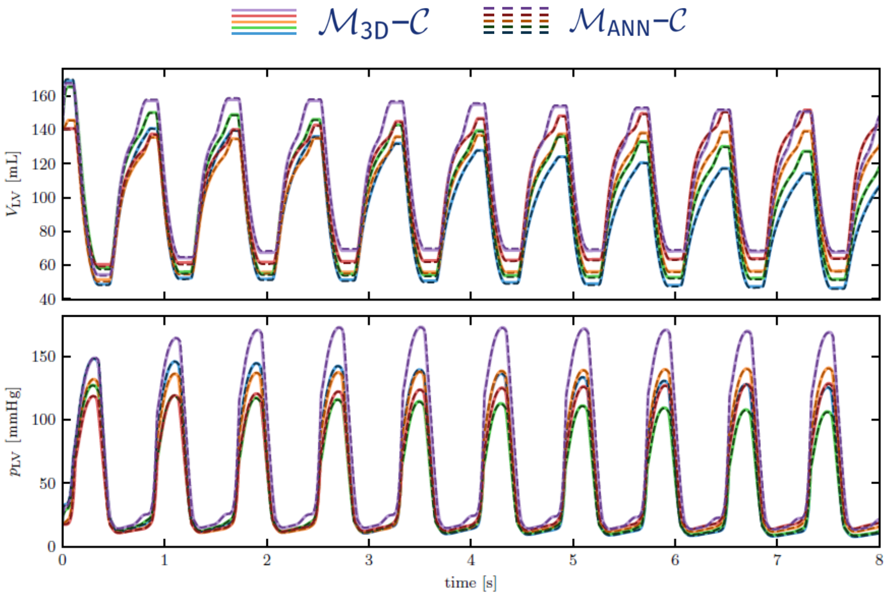
Proposed approach: surrogating just the 3D model

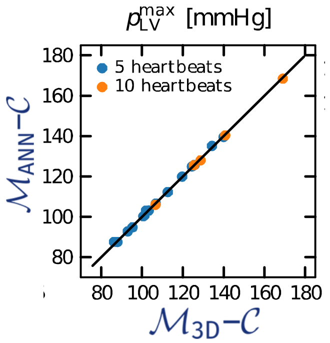


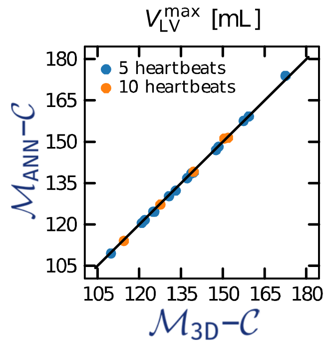
- Computationally inexpensive
- Provides pressures and volumes transients
- Only the parameter space of the mechanical model needs to be explored
- Possibility of performing simulations beyond the training time horizon

F. Regazzoni, M. Salvador, L. Dede', A. Quarteroni Computer Methods in Applied Mechanics and Engineering (2022)

Results







testing accuracy

	5 heartbeats			
	p_{LV}^{min}	p_{LV}^{max}	V_{LV}^{min}	V_{LV}^{max}
relative error	0.0097	0.0046	0.0139	0.0035
R^2	99.691	99.864	99.896	99.948
	10 heartbeats			
	p_{LV}^{min}	p_{LV}^{max}	V_{LV}^{min}	V_{LV}^{max}
relative error	0.0113	0.0037	0.0096	0.0031
R^2	99.924	99.980	99.851	99.944

Test dataset #1
Same time horizon as training dataset

Test dataset #2
Time horizon twice as long as in the training dataset

model	*task*	*computational platform*	*computational time*
$\mathcal{M}_{3D}$–$\mathcal{C}$	simulation of a heartbeat	32-core cluster	4 hours
$\mathcal{M}_{ANN}$–$\mathcal{C}$	simulation of a heartbeat	single core standard laptop	1 second
$\mathcal{M}_{ANN}$	training	single core standard laptop	18 hours
$\mathcal{M}_{3D}$–$\mathcal{C}$	training data generation	160-core cluster	~120 hours

460'000x speedup

F. Regazzoni, M. Salvador, L. Dede', A. Quarteroni Computer Methods in Applied Mechanics and Engineering (2022)

Use case: Bayesian Parameter Estimation

$\mathcal{F}: \mathbf{p} \mapsto \mathbf{q}$ parameters-to-QoIs map

$\mathbf{q}_{\text{obs}} = \mathcal{F}(\mathbf{p}) + \boldsymbol{\epsilon}$ observation

$\boldsymbol{\epsilon} \sim \mathcal{N}(\cdot|\mathbf{0}, \boldsymbol{\Sigma})$ $\boldsymbol{\Sigma}$ = noise covariance

$\pi_{\text{prior}}(\mathbf{p})$ prior distribution

Posterior distribution:

$$\pi_{\text{post}}(\mathbf{p}) = \frac{1}{Z} \mathcal{N}(\mathbf{q}_{\text{obs}}|\mathcal{F}(\mathbf{p}), \boldsymbol{\Sigma})\, \pi_{\text{prior}}(\mathbf{p})$$

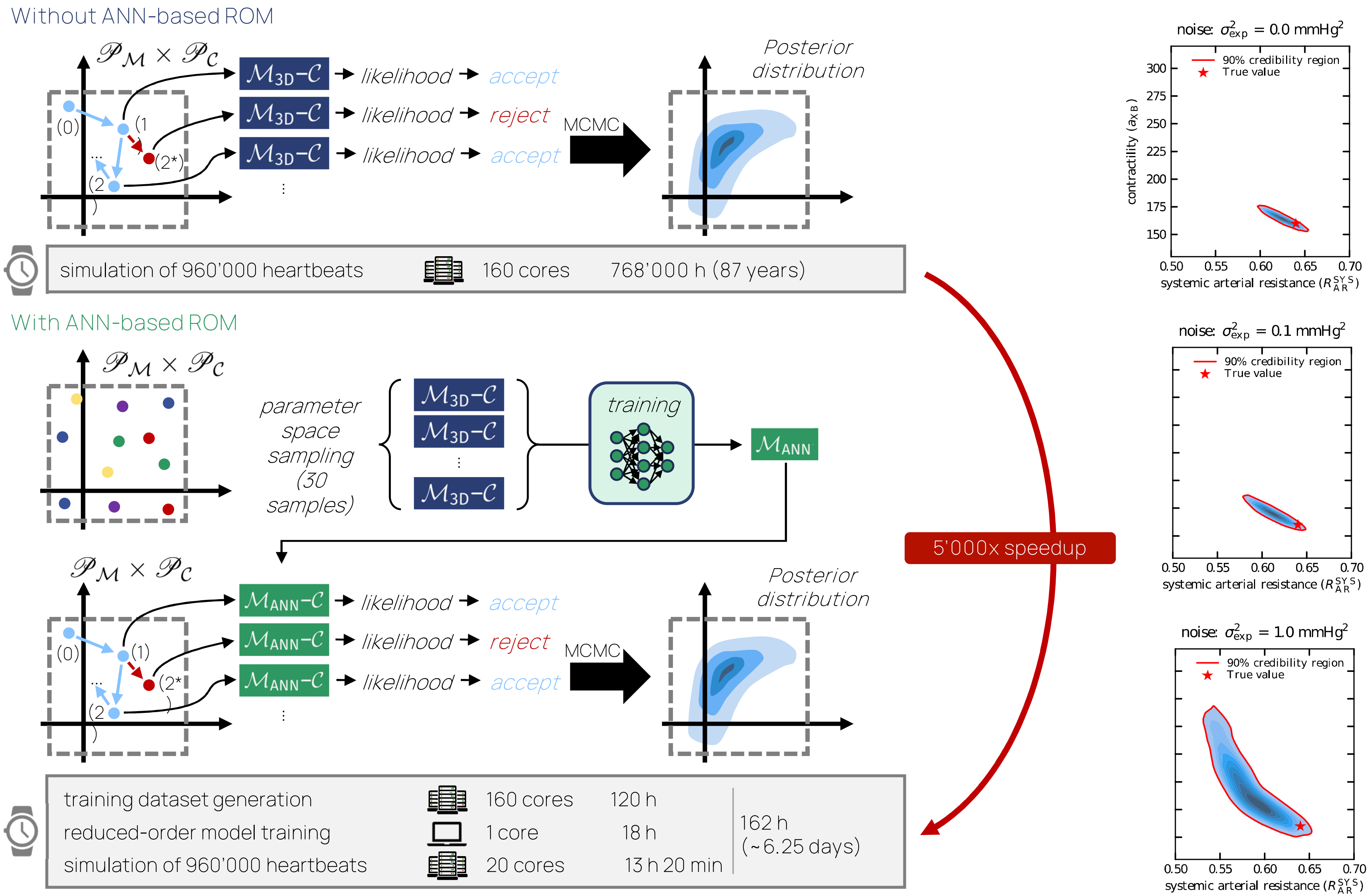
Accounting for the ROM error:

$$\mathcal{F}(\mathbf{p}) = \widetilde{\mathcal{F}}(\mathbf{p}) + \boldsymbol{\epsilon}_{\text{ROM}}$$

$$\mathbf{q}_{\text{obs}} = \widetilde{\mathcal{F}}(\mathbf{p}) + \boldsymbol{\epsilon}_{\text{ROM}} + \boldsymbol{\epsilon}_{\text{exp}}$$

Assuming independence:

$$\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_{\text{ROM}} + \boldsymbol{\Sigma}_{\text{exp}}$$



F. Regazzoni, M. Salvador, L. Dede', A. Quarteroni Computer Methods in Applied Mechanics and Engineering (2022)

Accounting for inter-sample variability

Model Learning: accounting for inter-sample variability

sample-specific latent parameter

Surrogate model

$$\begin{cases} \frac{d}{dt}\mathbf{s}(t) = \mathcal{NN}_{\text{dyn}}(\mathbf{s}(t), \mathbf{u}(t), \boldsymbol{\mu}; \mathbf{w}_{\text{dyn}}) & t \in (0,T) \\ \mathbf{y}(t) = \mathcal{NN}_{\text{dec}}(\mathbf{s}(t); \mathbf{w}_{\text{dec}}) & t \in (0,T) \\ \mathbf{s}(0) = \mathbf{0} \end{cases}$$

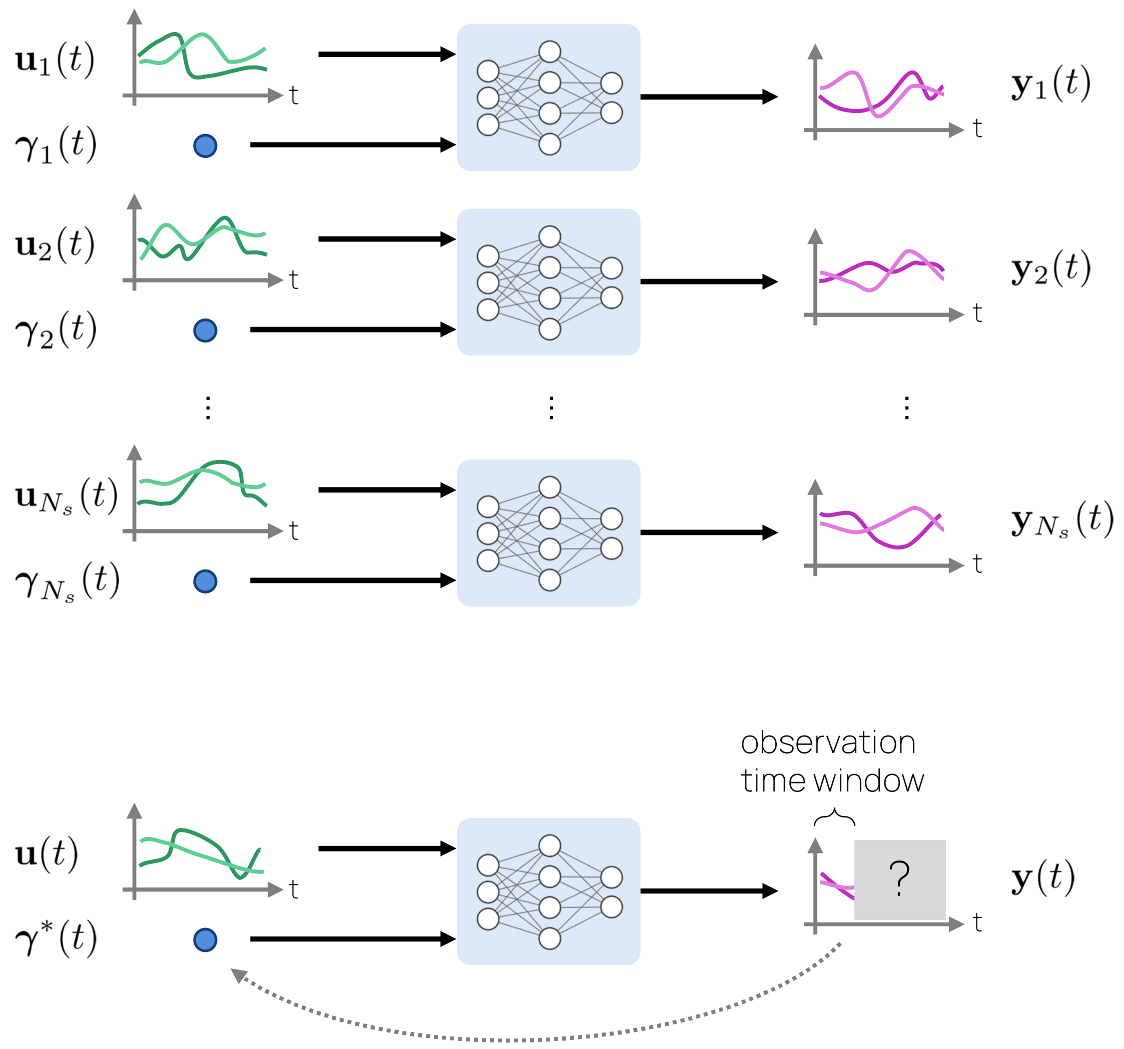
Training

$$\mathbf{w}^*_{\text{dyn}}, \mathbf{w}^*_{\text{dec}}, \{\boldsymbol{\gamma}^*_j\}_{j=1}^{N_s} = \underset{\mathbf{w}_{\text{dyn}}, \mathbf{w}_{\text{dec}}, \{\boldsymbol{\gamma}_j\}_{j=1}^{N_s}}{\text{argmin}} \sum_{j=1}^{N_s} \sum_{i=0}^{N_t} \|\widehat{\mathbf{y}}_j(t_i) - \mathbf{y}_j(t_i)\|^2$$

Inference

$$\boldsymbol{\gamma}^* = \underset{\boldsymbol{\gamma}}{\text{argmin}} \sum_{i=0}^{N_{\text{obs}}} \|\widehat{\mathbf{y}}(t_i) - \mathbf{y}(t_i)\|^2$$

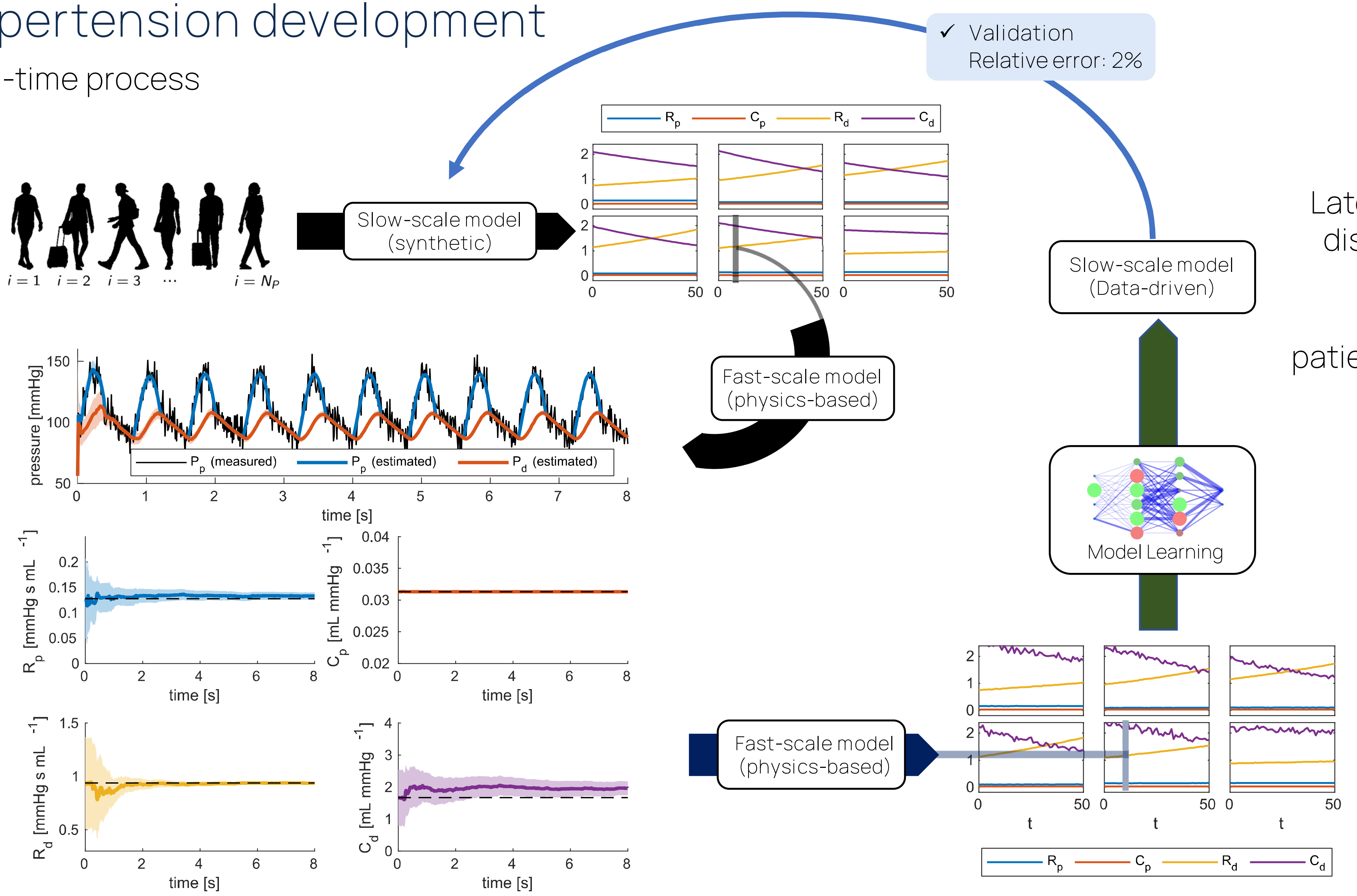
$$\begin{cases} \frac{d}{dt}\mathbf{s}(t) = \mathcal{NN}_{\text{dyn}}(\mathbf{s}(t), \mathbf{u}(t), \boldsymbol{\mu}, \boldsymbol{\gamma}^*; \mathbf{w}^*_{\text{dyn}}) & t \in (0,T) \\ \mathbf{y}(t) = \mathcal{NN}_{\text{dec}}(\mathbf{s}(t); \mathbf{w}^*_{\text{dec}}) & t \in (0,T) \\ \mathbf{s}(0) = \mathbf{0} \end{cases}$$



F. Regazzoni, D. Chapelle, P. Moireau International Journal for Numerical Methods in Biomedical Engineering (2021)

POC: hypertension development

Multiscale-in-time process



Latent parameter: disease severity

↓

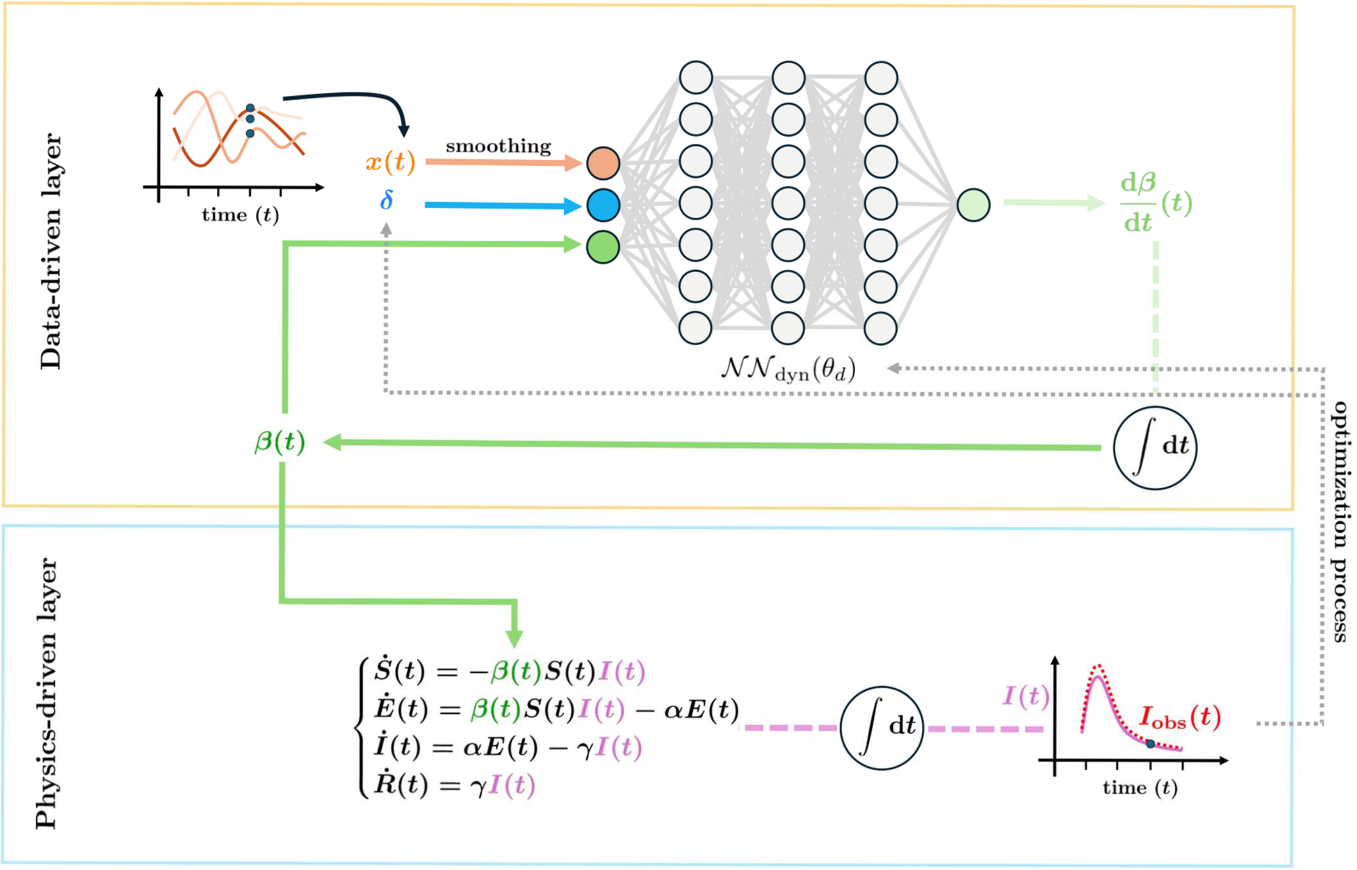
patient stratification

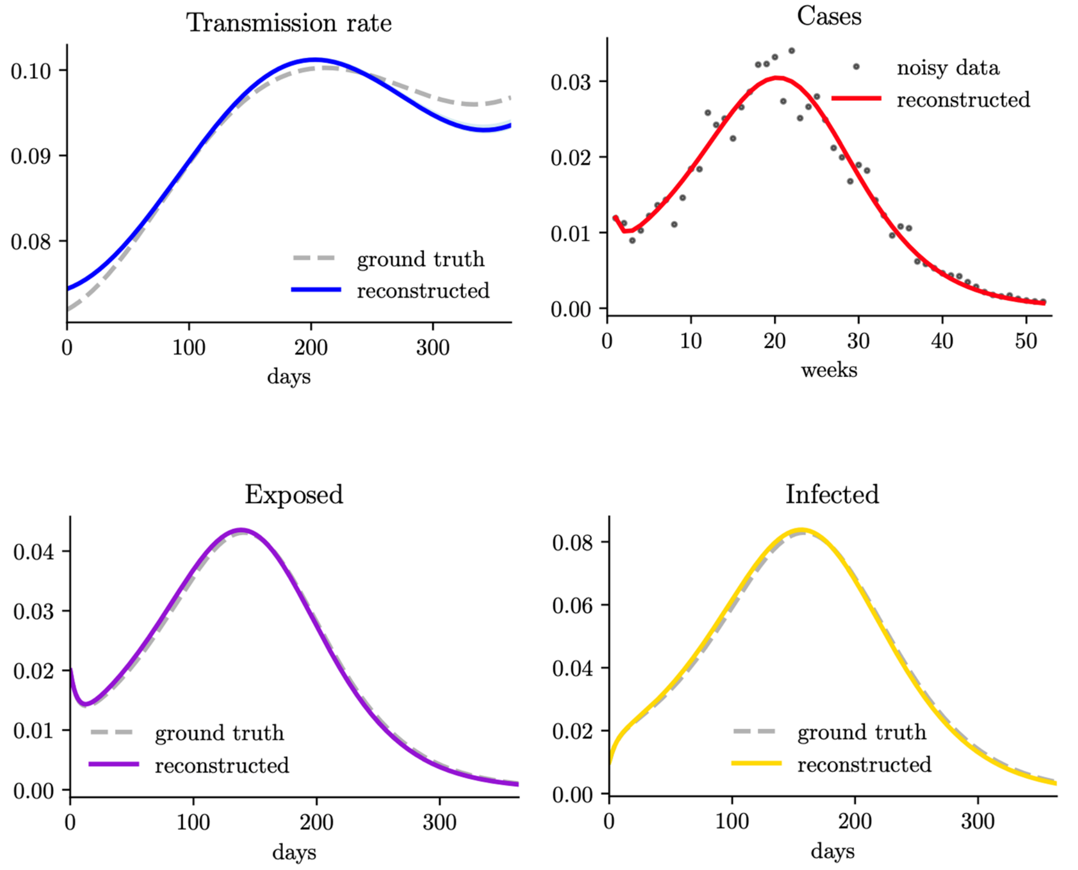
F. Regazzoni, D. Chapelle, P. Moireau International Journal for Numerical Methods in Biomedical Engineering (2021)

Application to epidemiology

Inferring the dynamics of transmission rate depending on exogenous variables (temperature, humidity) for epidemic forecasts, with application to influenza data from Italy between 2010 and 2020.

Latent parameter: unmodeled features of each virus strain





G. Ziarelli, S. Pagani, F. Regazzoni, N. Parolini, M. Verani, Computer Methods in Applied Mechanics and Engineering (2025)

Space-time Operator Learning

Operator learning for of time-dependent problems

- Displacement of a control point
- Generated power of a plant
- Revenues of a company
- Number of infected people
- Wrist blood pressure
- ...

- Control of an engine
- Wind strength
- Price of an asset
- Strength of a social measure
- Dose of a farmakon
- ...

discretization

- Displacement field of a body
- Temperature field in a pump
- Concentration of a pollutant
- Spread of an epidemic
- Blood velocity in an aneurysm
- ...

Latent Dynamics Networks (LDNets)

Full-order model (FOM)

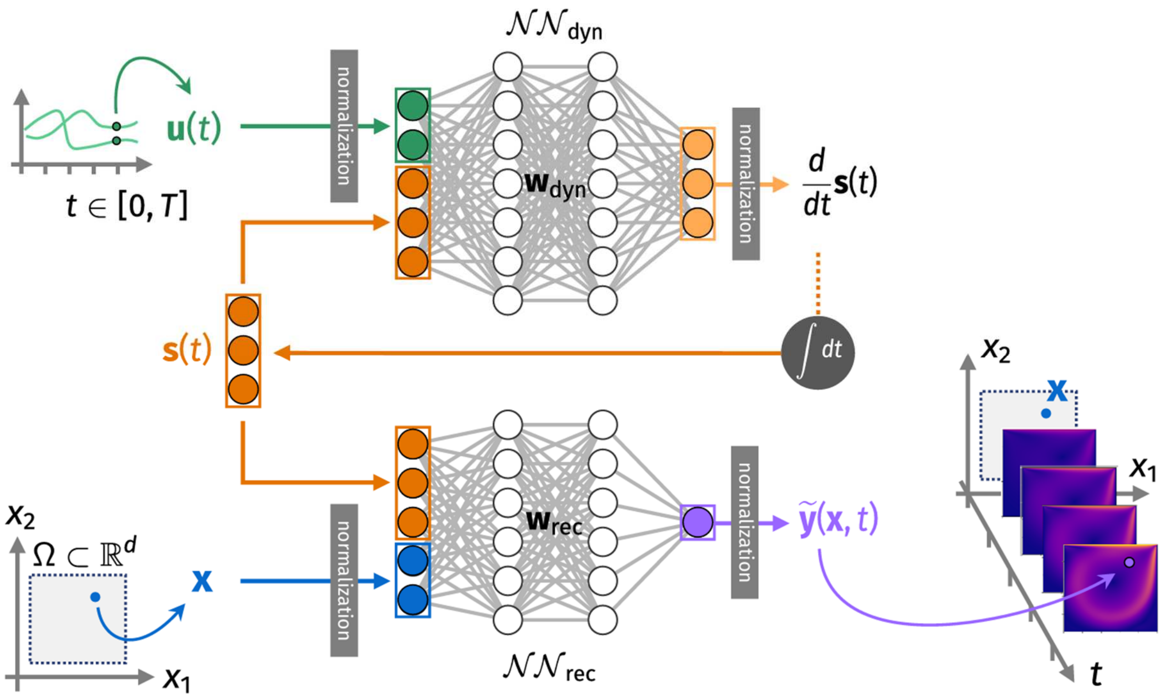
$$\begin{cases} \dfrac{d\mathbf{z}}{dt}(\mathbf{x},t) = \mathcal{F}(\mathbf{x},\mathbf{z},\mathbf{u}) & \mathbf{x}\in\Omega,\ t\in(0,T) \\ \mathbf{y}(\mathbf{x},t) = \mathcal{G}(\mathbf{x},\mathbf{z}) & \mathbf{x}\in\Omega,\ t\in(0,T) \\ \mathbf{z}(\mathbf{x},0) = \mathbf{z}_0(\mathbf{x}) \end{cases}$$

Surrogate model

$$\begin{cases} \dfrac{d}{dt}\mathbf{s}(t) = \mathcal{NN}_{\mathrm{dyn}}(\mathbf{s}(t),\mathbf{u}(t);\mathbf{w}_{\mathrm{dyn}}) & t\in(0,T) \\ \mathbf{y}(\mathbf{x},t) = \mathcal{NN}_{\mathrm{rec}}(\mathbf{x},\mathbf{s}(t);\mathbf{w}_{\mathrm{rec}}) & \mathbf{x}\in\Omega,\ t\in(0,T) \\ \mathbf{s}(0) = \mathbf{s}_0 \end{cases}$$

Training

$$\mathbf{w}^*_{\mathrm{dyn}},\mathbf{w}^*_{\mathrm{rec}} = \operatorname*{argmin}_{\mathbf{w}_{\mathrm{dyn}},\mathbf{w}_{\mathrm{rec}}} \sum_{j=1}^{N_s}\sum_{i=0}^{N_{obs}} \|\widehat{\mathbf{y}}_j(\mathbf{x}_i,t_i) - \mathbf{y}_j(\mathbf{x}_i,t_i)\|^2$$



- Latent state s(t): low-dimensional encoding of the high-dimensional HF model state z(x,t)
- Low-dimensional latent space discovered without the need of training an autoencoder
- Meshless representation of the space-dependent output: weights sharing

➡ LDNets never operate in the high-dimensional space

✓ Lightweight
✓ Easy to train
✓ Excellent generalization ability

F. Regazzoni, S. Pagani, M. Salvador, L. Dede', A. Quarteroni, Nature Communications (2024)

Test case #1: diffusion-reaction problem

Full-order model (FOM)

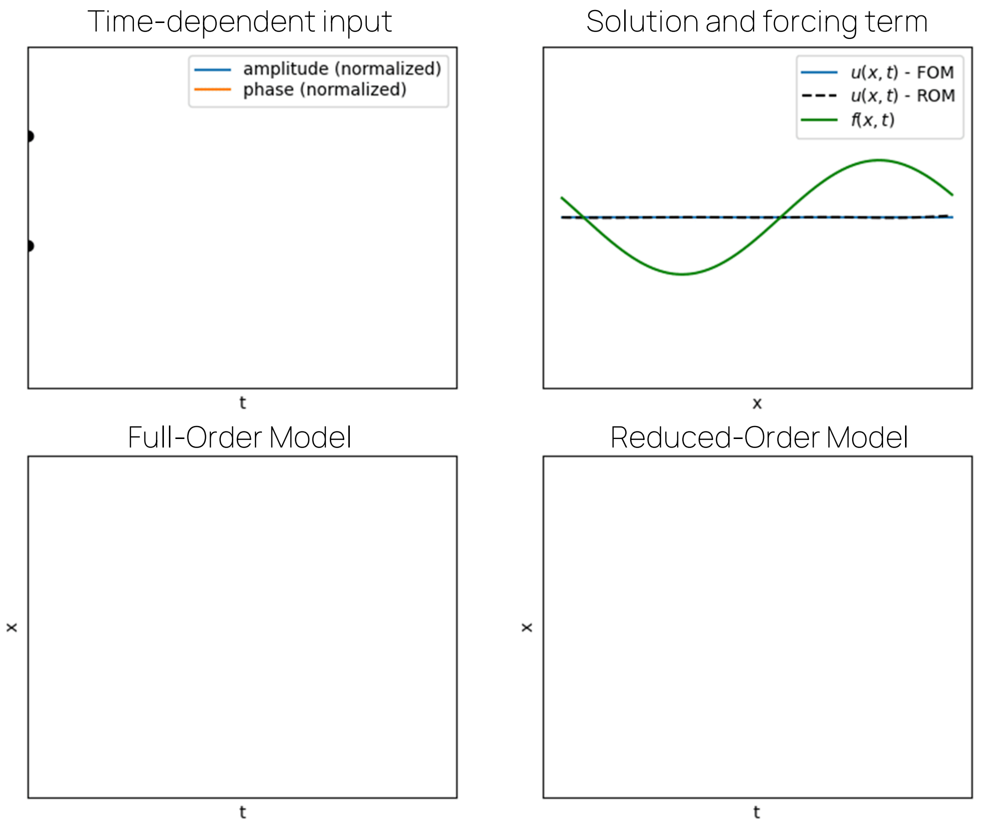
$$\begin{cases} \partial_t z(x,t) - \mu_1 \partial_{xx} z(x,t) + \mu_2 z(x,t) = f(x,t) & x \in (-1,1),\, t \in (0,T] \\ z(-1,t) = z(1,t) & t \in (0,T] \\ z(x,0) = 0 & x \in (-1,1) \end{cases}$$

$$f(x,t) = u_1(t)\cos(\pi\, x - u_2(t))$$

Goal: reconstruct $z(x,t)$

- The solution is a sine wave with the same period of $f(x,t)$
- At each time, the solution is fully determined by amplitude and phase
- The solution manifold dimension is exactly 2

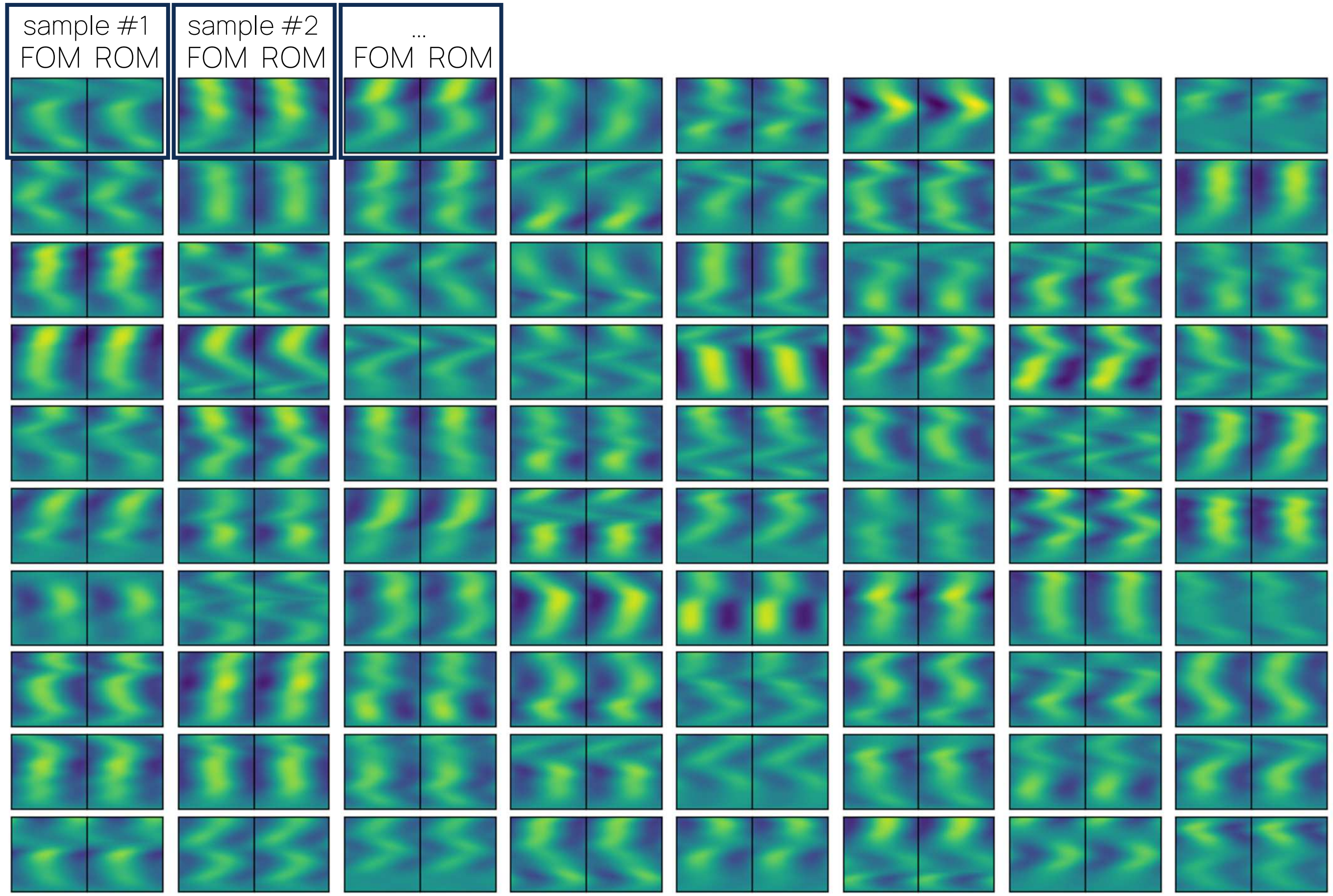
Are LDNets capable of learning a representation of the solution with 2 latent states?



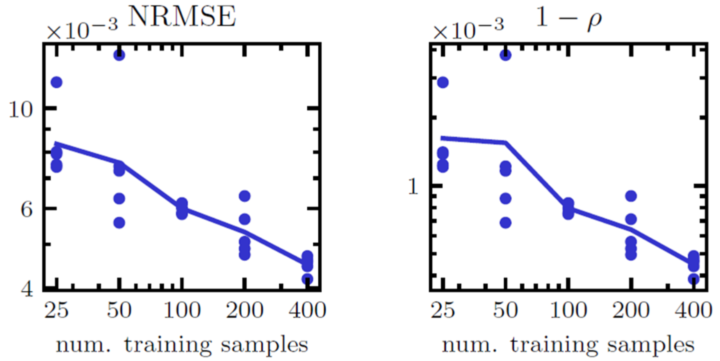
Test case #1: diffusion-reaction problem

Training/validation set:
- 100 samples
- 100 points in space
- 100 instants in time

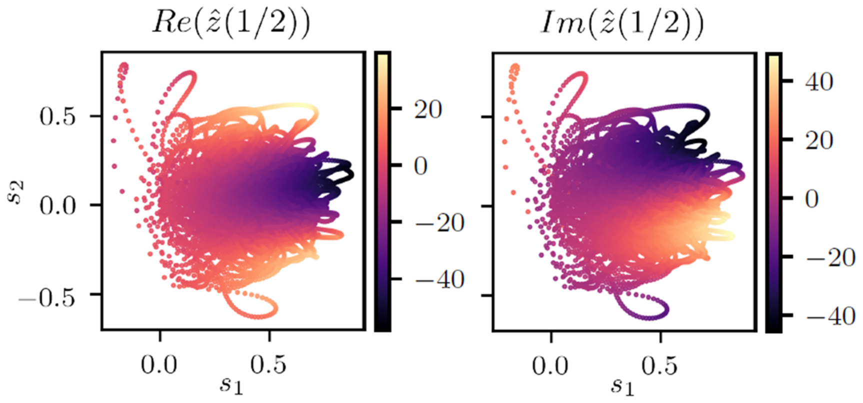
Testing set:
- 1000 samples
- 100 points in space
- 100 instants in time



Accuracy vs training set size



Discovered latent space vs Fourier space

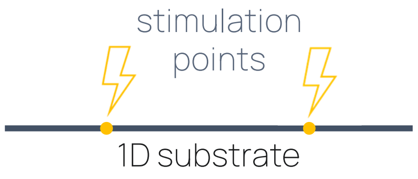


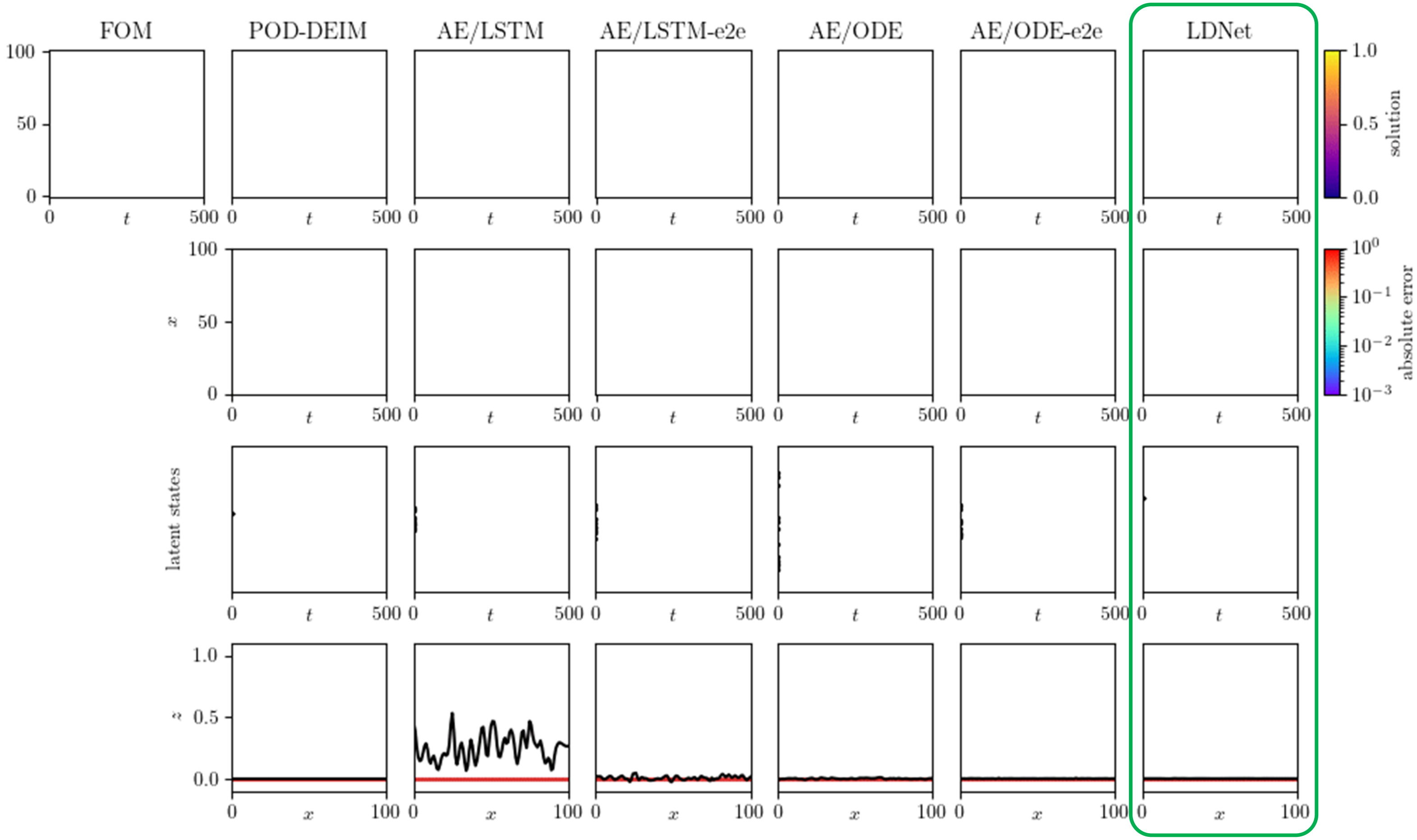
A compact encoding equivalent to the Fourier transform is automatically discovered

F. Regazzoni, S. Pagani, M. Salvador, L. Dede', A. Quarteroni, Nature Communications (2024)

Test case #2: cardiac electrophysiology

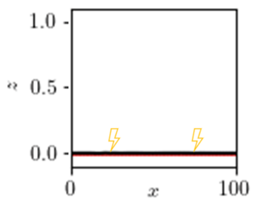
Goal: Learning the excitation-propagation dynamics of an excitable tissue





Aliev-Panfilov model

$$\frac{\partial z}{\partial t} - D\frac{\partial^2 z}{\partial x^2} = Kz(1-z)(z-\alpha) - zw + I_{\text{stim}}(x,t)$$
$$\frac{\partial w}{\partial t} = \left(\gamma + \frac{\mu_1 w}{\mu_2 + z}\right)(-w - Kz(z-b-1))$$

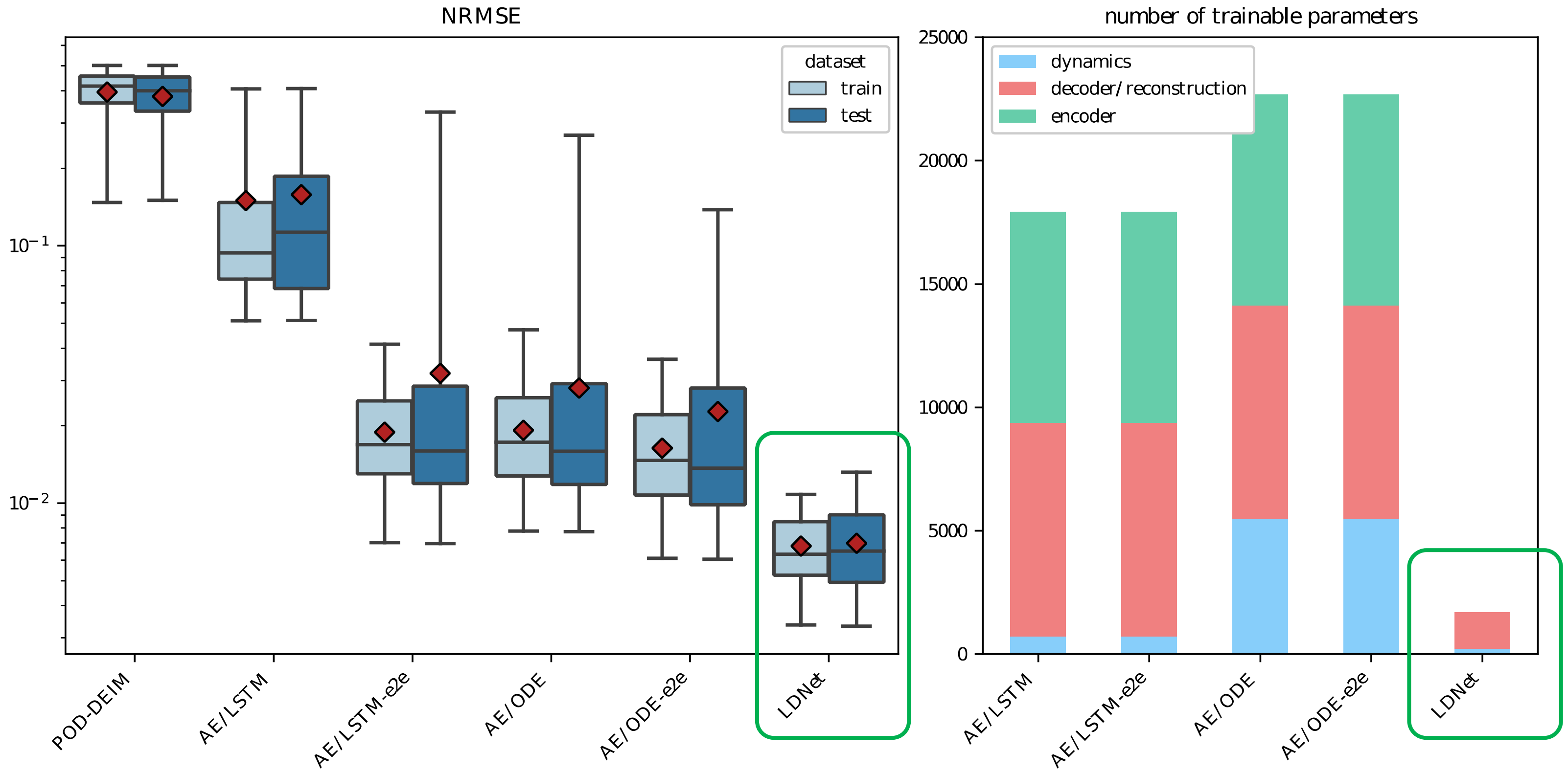


F. Regazzoni, S. Pagani, M. Salvador, L. Dede', A. Quarteroni, Nature Communications (2024)

Test case #2: cardiac electrophysiology

LDNets outperform state-of-the-art approaches to learn space-time dynamics:

- ✓ 5 times more accurate
- ✓ Better generalization (lower overfitting)
- ✓ 10 times fewer parameters



F. Regazzoni, S. Pagani, M. Salvador, L. Dede', A. Quarteroni, Nature Communications (2024)

PART I

Operator Learning for time-dependent problems

PART II

Operator Learning in variable domain

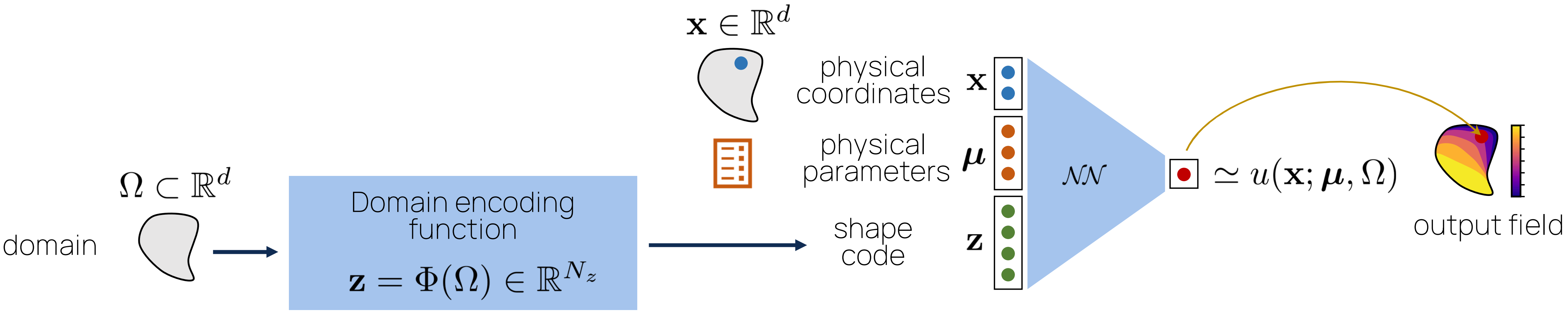
Universal Solution Manifold Network (USM-Net)

Operator learning method that learns the solution map underlying a PDE, in a universal manner w.r.t. the domain

Full-order model

$$\begin{cases} \mathcal{F}(u;\boldsymbol{\mu}) = 0, \text{ in } \Omega \\ \text{Boundary conditions} \end{cases}$$

→ Parametric PDE solution $u(\mathbf{x};\boldsymbol{\mu},\Omega)$



Physical-coordinate based USM-Net (PC-USM-Net)

Training: $$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{j=1}^{N_{\text{samples}}} \sum_{i=0}^{N_{\text{points}}} \left\| u(\mathbf{x}_i;\boldsymbol{\mu}_j,\Omega_j) - \mathcal{NN}(\boldsymbol{\mu}_j,\mathbf{x}_i,\Phi(\Omega_j);\mathbf{w}) \right\|^2$$

F. Regazzoni, S. Pagani, A. Quarteroni, ASME Journal of Biomechanical Engineering (2022)

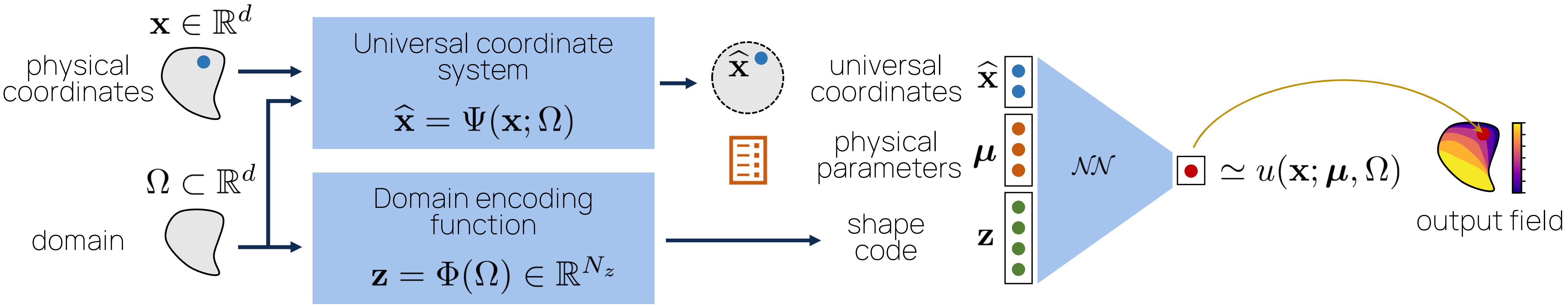
Universal Solution Manifold Network (USM-Net)

Operator learning method that learns the solution map underlying a PDE, in a universal manner w.r.t. the domain

Full-order model

$$\begin{cases} \mathcal{F}(u;\boldsymbol{\mu}) = 0, \text{ in } \Omega \\ \text{Boundary conditions} \end{cases}$$

→ Parametric PDE solution $u(\mathbf{x};\boldsymbol{\mu},\Omega)$



Universal-coordinate based USM-Net (UC-USM-Net)

Training: $$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{j=1}^{N_{\text{samples}}} \sum_{i=0}^{N_{\text{points}}} \left\| u(\mathbf{x}_i;\boldsymbol{\mu}_j,\Omega_j) - \mathcal{NN}(\boldsymbol{\mu}_j, \Psi(\mathbf{x}_i,\Omega_j), \Phi(\Omega_j);\mathbf{w}) \right\|^2$$

F. Regazzoni, S. Pagani, A. Quarteroni, ASME Journal of Biomechanical Engineering (2022)

Domain encoding functions

1. Parametrized domains

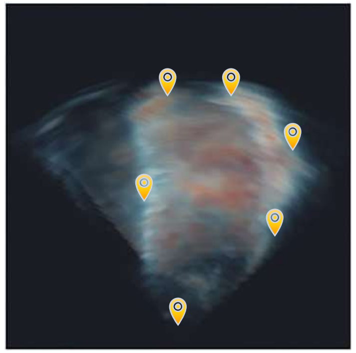
Define the shape code as the parameters defining the domain

$$\gamma \begin{pmatrix} r \\ \theta \end{pmatrix} = \begin{pmatrix} r\cos\theta \\ r\sin\theta \\ r^2 \end{pmatrix}$$

where $0 \leq \theta < 2\pi$ and $0 \leq r < 2.$

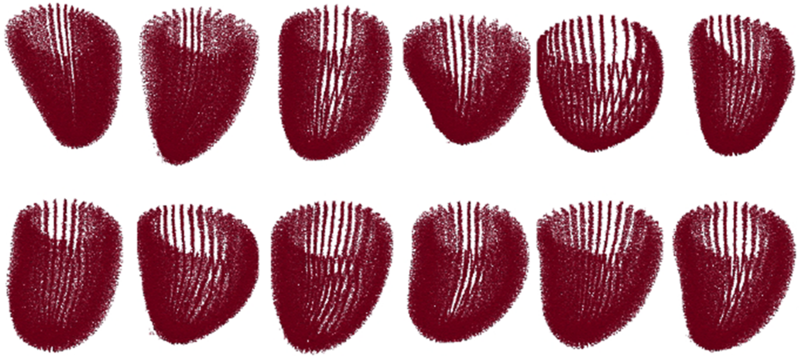
2. Landmarks (coordinates of key points)

Shape codes are defined as the coordinates of key points (automatically or manually annotated)



3. Statistical shape modelling coefficients

Perform Principal Component Analysis (PCA) on point clouds defining the training geometries, then define the shape codes as the coefficients associated with the first N_z principal directions



credits: D. Carrara, M. Hirschvogel

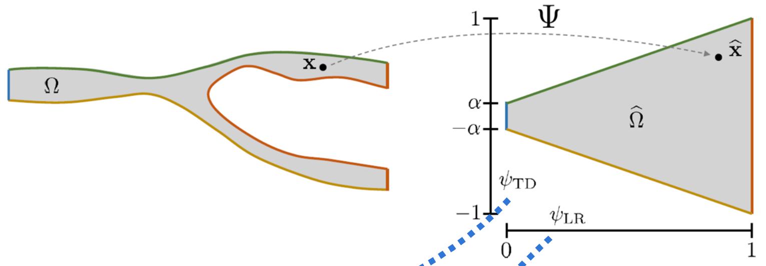
4. Signed distance function (SDF) encoding

(We will see later)

Test case: blood flow in a bifurcating domain

Goal: predict the velocity and pressure field in a coronary bifurcation for an unseen geometry

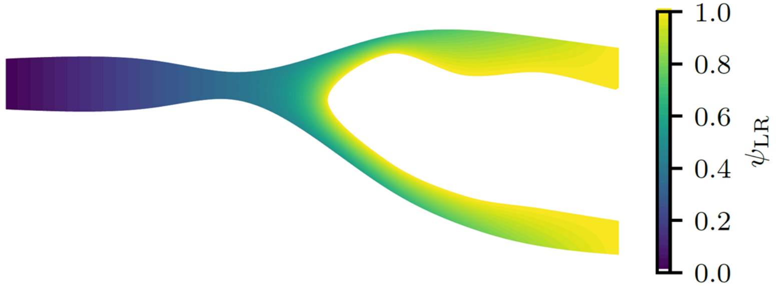
Universal coordinate System



Top-down coordinate

$$
\begin{cases}
-\Delta\psi_{\mathrm{TD}} = 0 & \text{in } \Omega, \\
\psi_{\mathrm{TD}} = +\alpha + (1-\alpha)\dfrac{x - x_{\min}}{x_{\max} - x_{\min}} & \text{on } \Gamma_{\mathrm{top}}, \\
\psi_{\mathrm{TD}} = -\alpha - (1-\alpha)\dfrac{x - x_{\min}}{x_{\max} - x_{\min}} & \text{on } \Gamma_{\mathrm{bottom}}, \\
\dfrac{\partial\psi_{\mathrm{TD}}}{\partial\mathbf{n}} = 0 & \text{on } \Gamma_{\mathrm{in}} \cup \Gamma_{\mathrm{out}} \cup \Gamma_{\mathrm{front}}
\end{cases}
$$

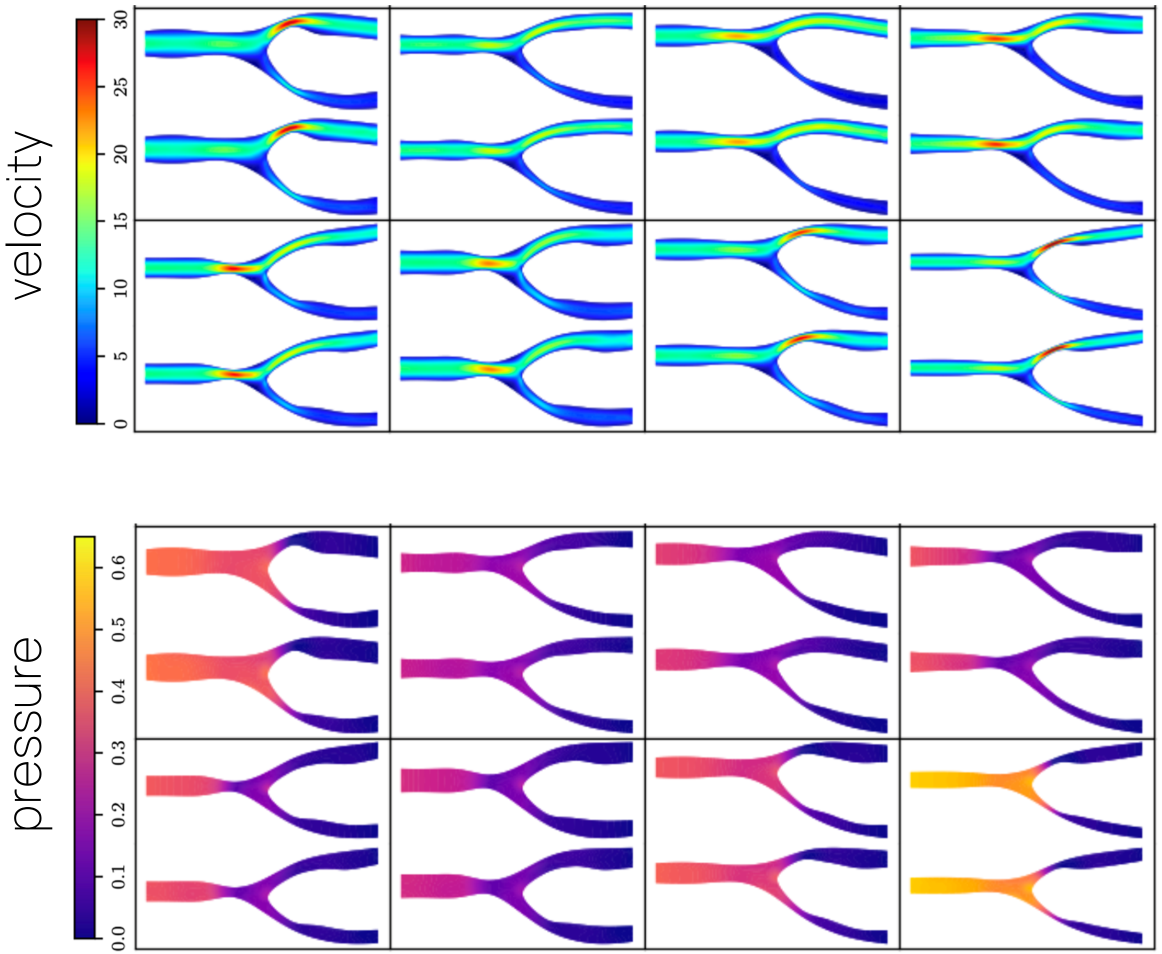
Left-right coordinate

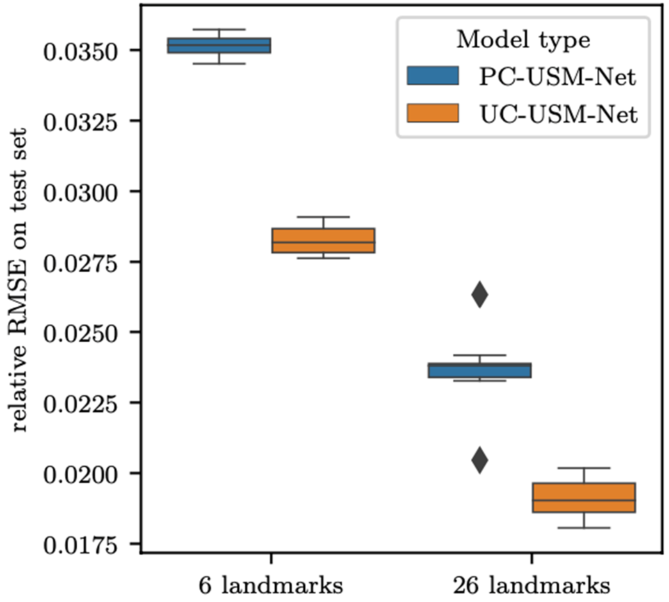


$$
\begin{cases}
-\Delta\psi_{\mathrm{LR}} = 0 & \text{in } \Omega, \\
\psi_{\mathrm{LR}} = 0 & \text{on } \Gamma_{\mathrm{in}}, \\
\psi_{\mathrm{LR}} = 1 & \text{on } \Gamma_{\mathrm{out}} \cup \Gamma_{\mathrm{front}}, \\
\psi_{\mathrm{LR}} = \dfrac{x - x_{\min}}{x_{\max} - x_{\min}} & \text{on } \Gamma_{\mathrm{top}} \cup \Gamma_{\mathrm{bottom}}
\end{cases}
$$

ld):

Results





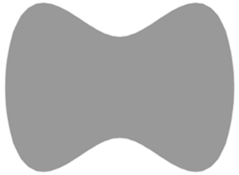
- Good generalization even with few landmarks
- Universal coordinate system allows to increase the accuracy

F. Regazzoni, S. Pagani, A. Quarteroni, ASME Journal of Biomechanical Engineering (2022)

Signed Distance Function encoding

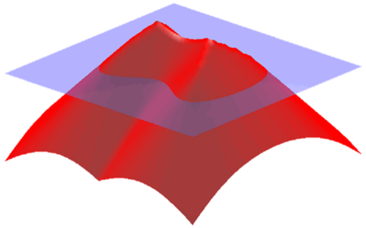
Encoding domains through the associated signed distance function

$\Omega \subset \mathbb{R}^d$



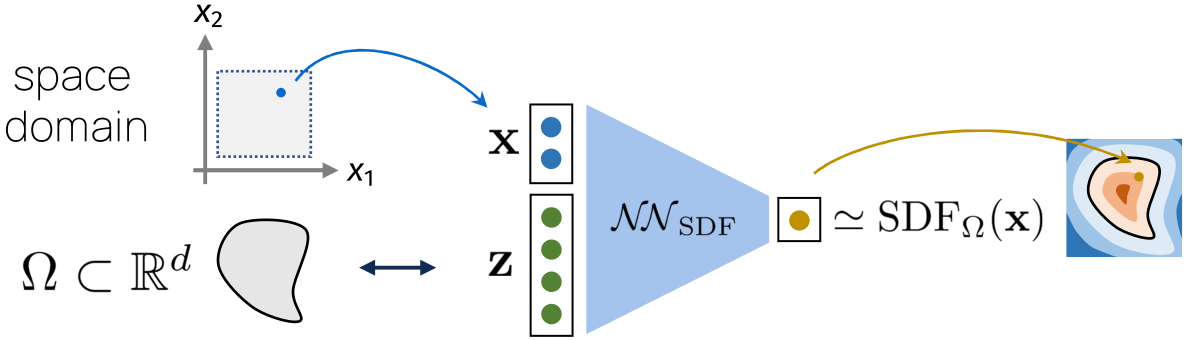
Signed distance function (SDF)

$$\mathrm{SDF}_\Omega(\mathbf{x}) = \begin{cases} d(\mathbf{x}, \partial\Omega) & \text{if } \mathbf{x} \in \Omega, \\ -d(\mathbf{x}, \partial\Omega) & \text{if } \mathbf{x} \notin \Omega, \end{cases}$$

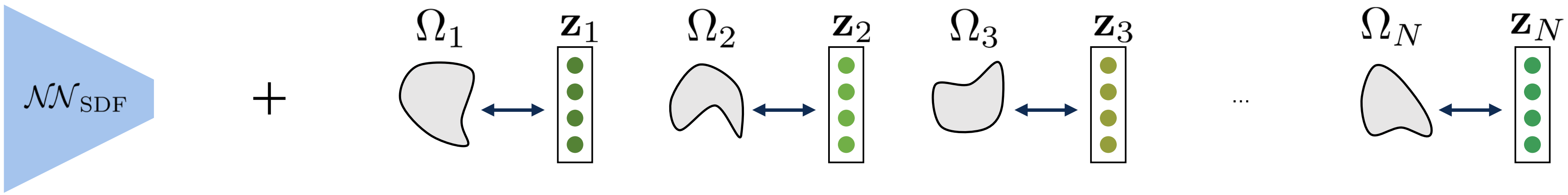


Encoding domains through the associated signed distance function

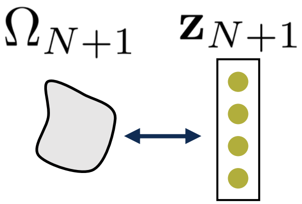
Deep SDFs [J.J. Park et al. Proceedings of the IEEE/CVF (2019)]



Training: $$\{\mathbf{z}_j^*\}_{j=1}^N, \mathbf{w}_{\mathrm{SDF}}^* = \operatorname*{argmin}_{\{\mathbf{z}_j\}_j, \mathbf{w}_{\mathrm{SDF}}} \sum_{j=1}^{N} \sum_{i=0}^{N_{\mathrm{points}}} \left|\mathrm{SDF}_{\Omega_j}(\mathbf{x}_i) - \mathcal{NN}_{\mathrm{SDF}}(\mathbf{x}_i, \mathbf{z}_j; \mathbf{w}_{\mathrm{SDF}})\right|^2$$

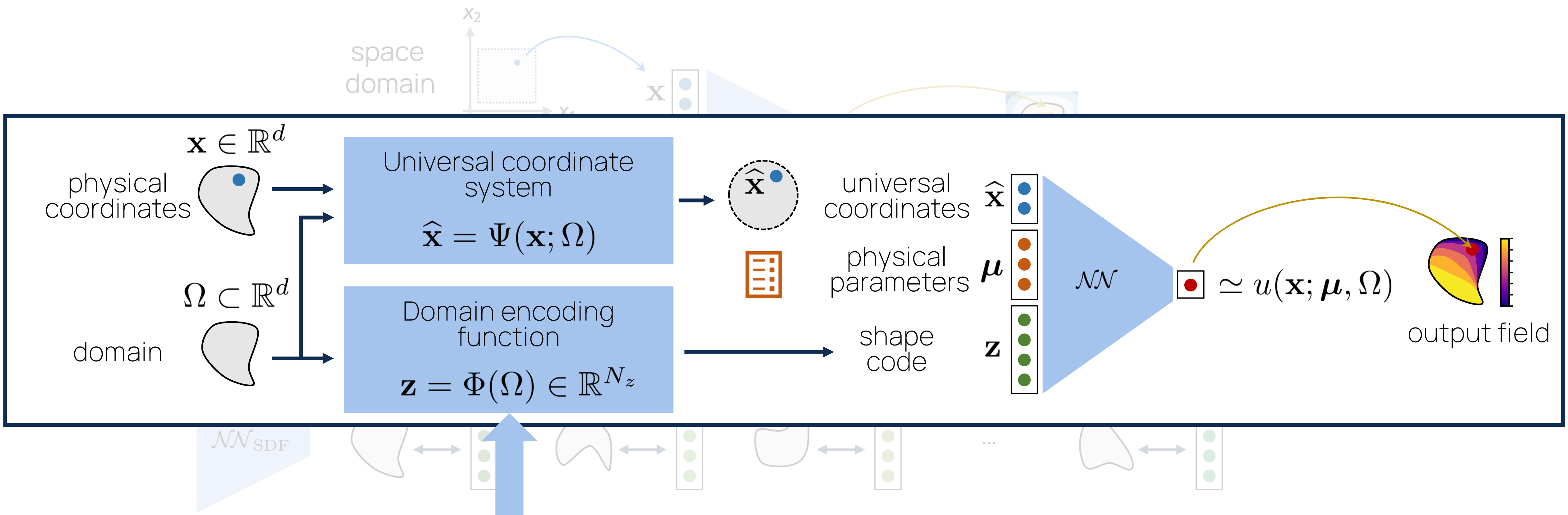


Inference: $$\mathbf{z}_{N+1}^* = \operatorname*{argmin}_{\mathbf{z}_{N+1}} \sum_{i=0}^{N_{\mathrm{points}}} \left|\mathrm{SDF}_{\Omega_{N+1}}(\mathbf{x}_i) - \mathcal{NN}_{\mathrm{SDF}}(\mathbf{x}_i, \mathbf{z}_{N+1}; \mathbf{w}_{\mathrm{SDF}}^*)\right|^2$$

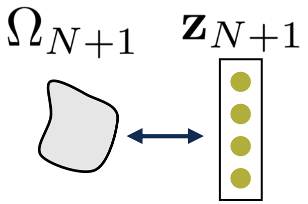


Encoding domains through the associated signed distance function

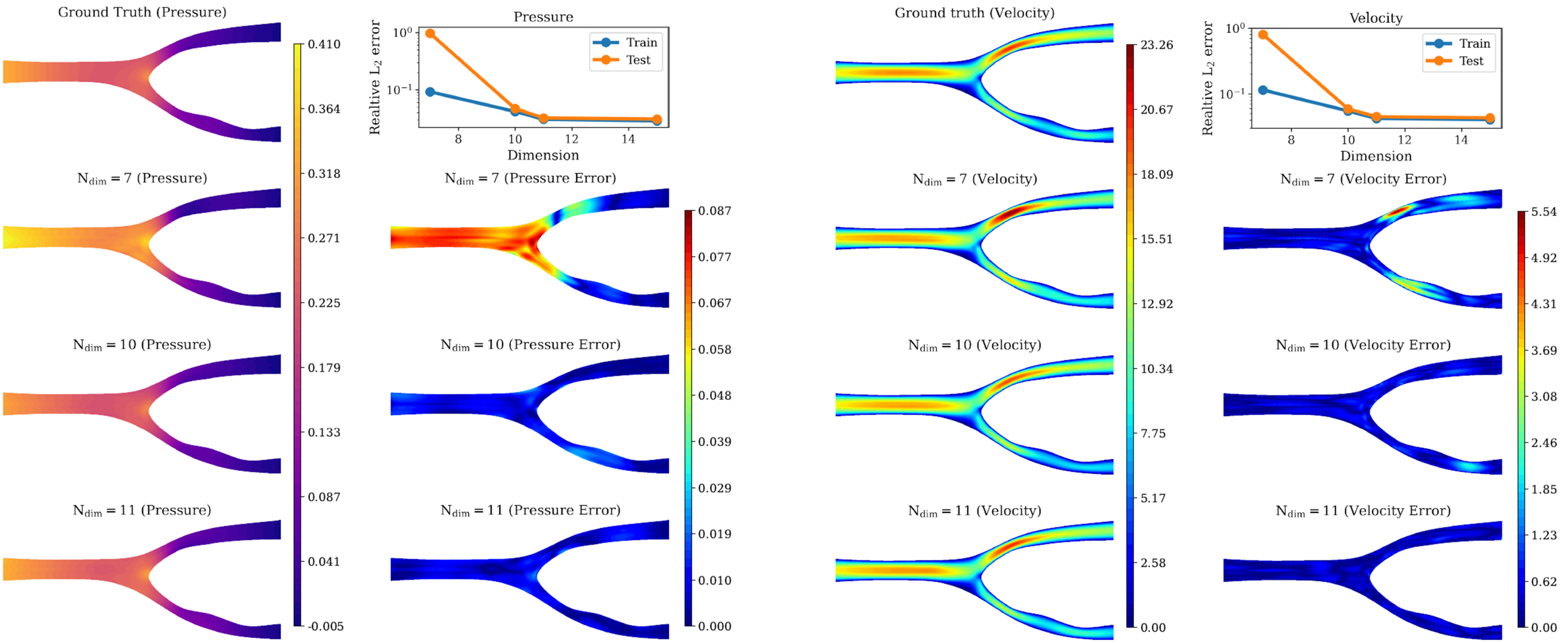
Deep SDFs [J.J. Park et al. Proceedings of the IEEE/CVF (2019)]



Inference: $$\mathbf{z}^*_{N+1} = \operatorname*{argmin}_{\mathbf{z}_{N+1}} \sum_{i=0}^{N_{\text{points}}} \left| \mathrm{SDF}_{\Omega_{N+1}}(\mathbf{x}_i) - \mathcal{NN}_{\mathrm{SDF}}(\mathbf{x}_i, \mathbf{z}_{N+1}; \mathbf{w}^*_{\mathrm{SDF}}) \right|^2$$

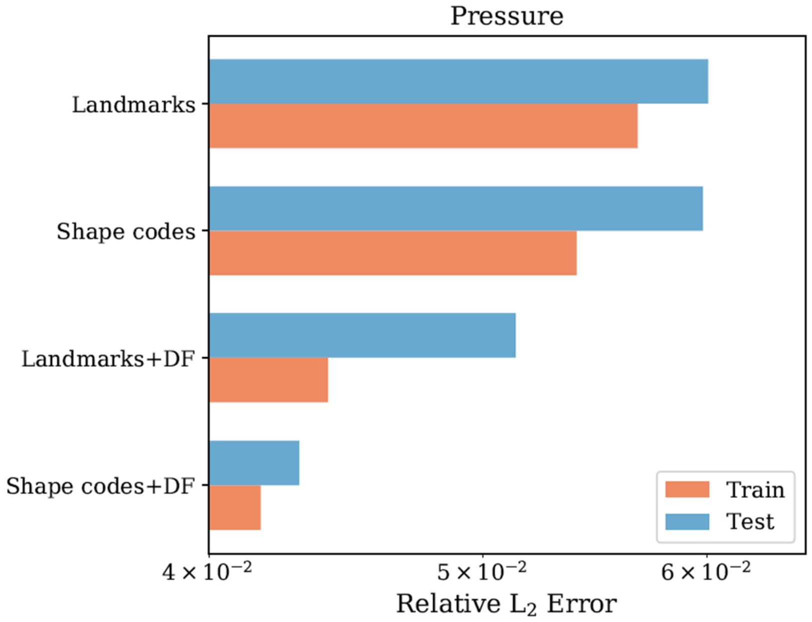


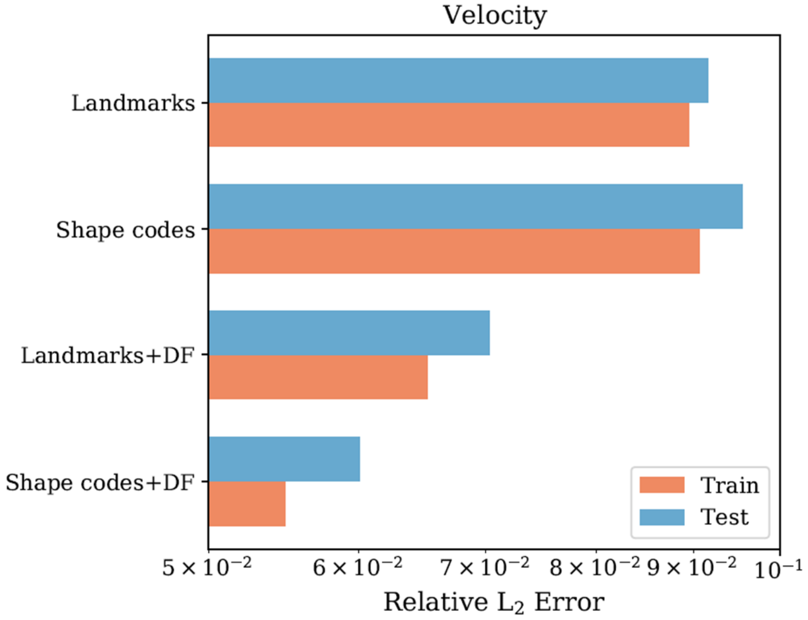
Results (SDF-USM-Net)



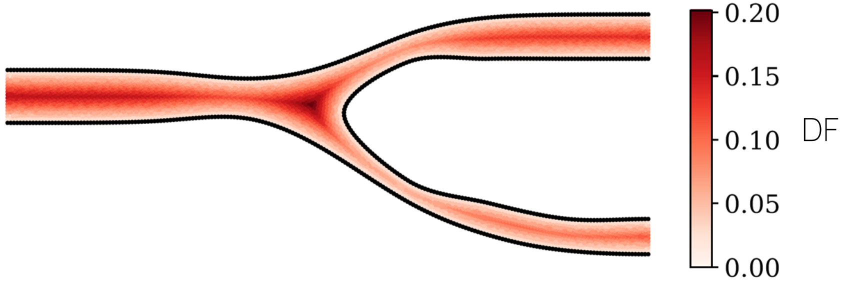
L. Zhang, S. Pagani, J. Zhang, F. Regazzoni, arXiv (2024)

Landmarks vs SDF-based shape codes



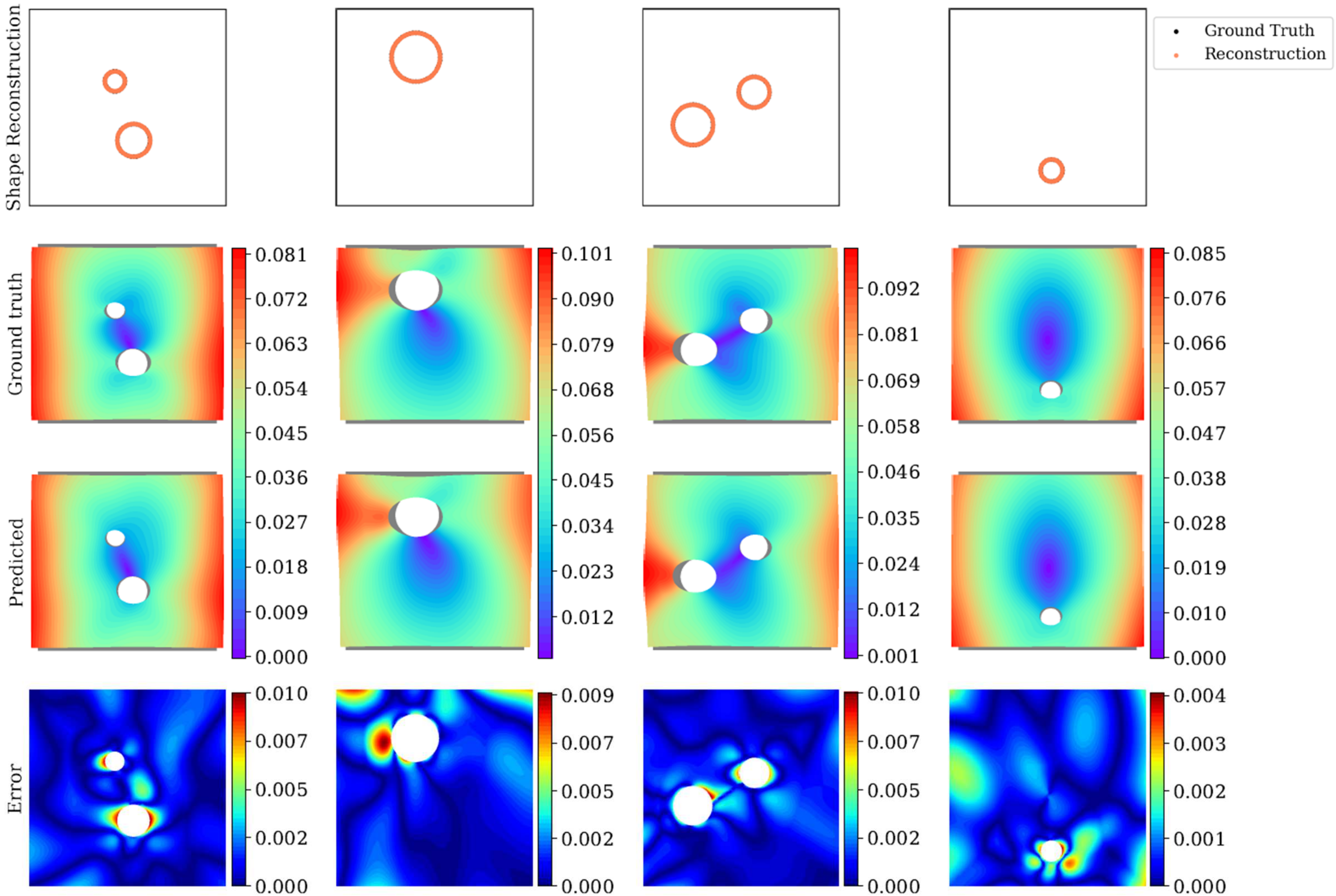


We provide as additional feature the distance function (DF) from the Dirichlet boundary:

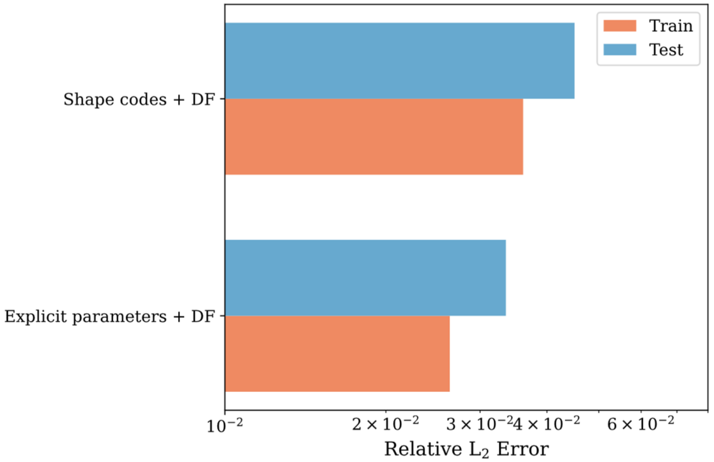


L. Zhang, S. Pagani, J. Zhang, F. Regazzoni, arXiv (2024)

Test case: perforated plate under load (variable topology)



We benchmark SDF-based shape codes against the exact parametrization (coordinates and radii of the holes)



L. Zhang, S. Pagani, J. Zhang, F. Regazzoni, arXiv (2024)

Thank you for your attention

PostDoc positions opening soon

FIS (Italian Science Fund) Starting Grant (1.3 M€)

“ *SYNERGIZE*: Synergizing Numerical Methods and Machine Learning for a new generation of computational models”

If interested, reach out at francesco.regazzoni@polimi.it

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA

Financed by the PRIN 2022 PNRR Project - Prot. P2022N5ZNP
“SIDDMs: shape-informed data-driven models for parametrized PDEs, with application to computational cardiology”

POLITECNICO MILANO 1863
DIPARTIMENTO DI MATEMATICA
DEPARTMENT OF EXCELLENCE 2023-2027

With the support of MUR, grant Dipartimento di Eccellenza 2023-2027