BUILD-IT2023 Workshop

October 20th, 2023 (Rome)

BUILD-IT Workshop 2023

Automatic discovery of low-dimensional dynamics underpinning time-dependent PDEs by means of Latent Dynamics Neural Networks

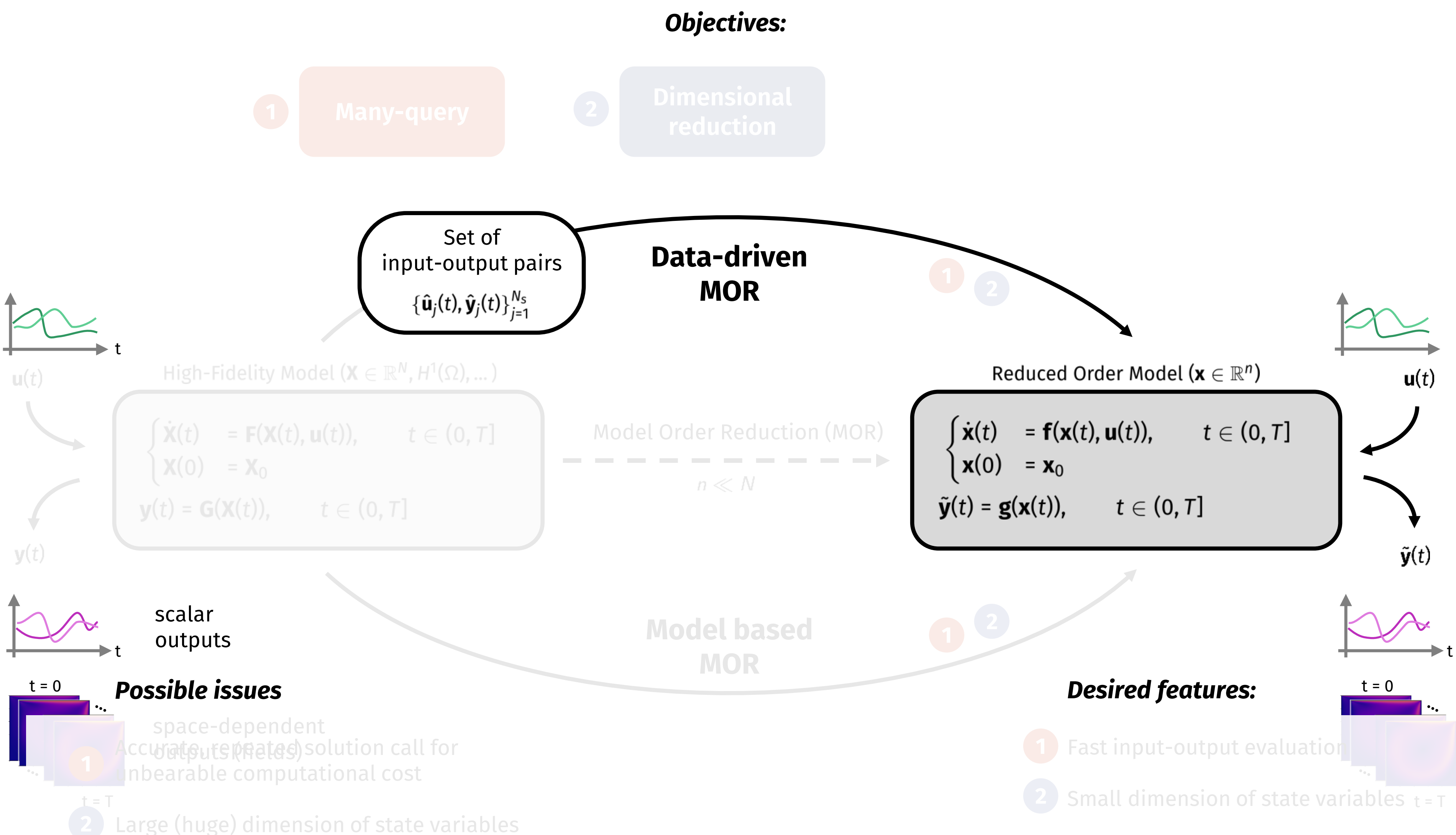
Francesco Regazzoni, Stefano Pagani, Matteo Salvador, Luca Dede', Alfio Quarteroni

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

MOX

MOX - Dipartimento di Matematica, Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

Model Order Reduction of time-dependent models

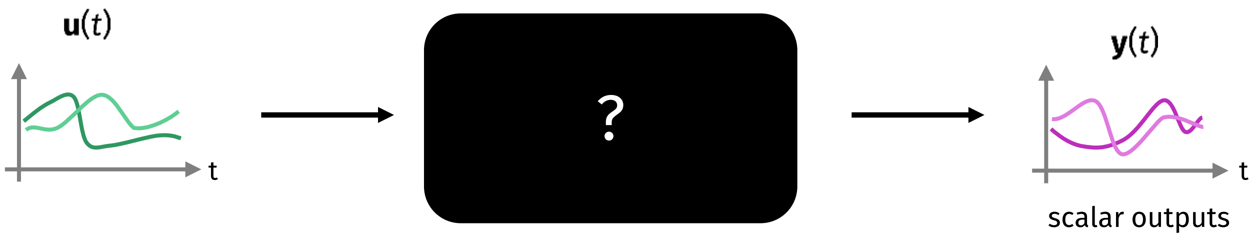


$$F = G\frac{m_1\, m_2}{r^2}$$

$$m\,\frac{d^2 s}{dt^2} = F$$

Challange: How to automate this process?

Data-driven modeling of time-dependent processes



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

Learning time-dependent differential equations

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. Candidate models class:

$n \in \mathbb{R}$

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

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

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

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

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

It uniquely identifies a map:

$$\varphi_{\mathbf{f},\mathbf{g},\mathbf{x}_0} : \mathcal{U} \to \mathcal{Y}$$

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

3. Best-approximation problem:

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

Or, equivalently:

$$\begin{cases} \displaystyle\min_{\mathbf{f}\in\widehat{\mathcal{F}}, \mathbf{g}\in\widehat{\mathcal{G}}, \mathbf{x}_0\in\widehat{\mathcal{X}}} & \frac{1}{2}\sum_{j=1}^{N_s} \int_0^T |\hat{\mathbf{y}}_j(t) - \mathbf{y}_j(t)|^2 dt \\ \text{s.t.} & \dot{\mathbf{x}}_j(t) = \mathbf{f}(\mathbf{x}_j(t), \hat{\mathbf{u}}_j(t)), \quad t \in (0,T], \quad j = 1,\dots,N_s \\ & \mathbf{x}_j(0) = \mathbf{x}_0, \quad j = 1,\dots,N_s, \\ & \mathbf{y}_j(t) = \mathbf{g}(\mathbf{x}_j(t)), \quad t \in (0,T], \quad j = 1,\dots,N_s \end{cases}$$

? How to select the sets of candidate functions?

? How to solve the best-approximation problem?

Universal approximation

Theorem (Regazzoni, Dede', Quarteroni, *JCP* 2019)

Let U be compact and suppose that $\widehat{\mathcal{F}} \subseteq \mathcal{F}_n$ and $\widehat{\mathcal{G}} \subseteq \mathcal{G}_n$ are such that for each compact set $E \subset \mathbb{R}^n$:

$$\forall \varepsilon > 0 \quad \forall \mathbf{f} \in \mathcal{F}_n \quad \exists \widehat{\mathbf{f}} \in \widehat{\mathcal{F}} \quad \forall \mathbf{x} \in E, \mathbf{u} \in U \quad \text{s.t.} \quad \left|\mathbf{f}(\mathbf{x}, \mathbf{u}) - \widehat{\mathbf{f}}(\mathbf{x}, \mathbf{u})\right| \leq \varepsilon$$

$$\forall \varepsilon > 0 \quad \forall \mathbf{g} \in \mathcal{G}_n \quad \exists \widehat{\mathbf{g}} \in \widehat{\mathcal{G}} \quad \forall \mathbf{x} \in E \quad \text{s.t.} \quad \left|\mathbf{g}(\mathbf{x}) - \widehat{\mathbf{g}}(\mathbf{x})\right| \leq \varepsilon.$$

Then, the subset of models $\Phi^{\widehat{\mathcal{F}}, \widehat{\mathcal{G}}, \mathcal{X}_n}$ is dense in the model space $\Phi^{\mathcal{F}_n, \mathcal{G}_n, \mathcal{X}_n}$:

$$\forall \varepsilon > 0 \quad \forall \varphi \in \Phi^{\mathcal{F}_n, \mathcal{G}_n, \mathcal{X}_n} \quad \exists \widehat{\varphi} \in \Phi^{\widehat{\mathcal{F}}, \widehat{\mathcal{G}}, \mathcal{X}_n} \quad \text{s.t.} \quad \|\varphi - \widehat{\varphi}\|_{\mathcal{C}^0(\mathcal{U};\mathcal{Y})} \leq \varepsilon.$$

Definition

An **Artifical Neural Network** (ANN) is a function: $\Phi = T_n \circ \sigma \circ T_{n-1} \cdots \circ T_1 \circ \sigma \circ T_0$
where T_j are affine functions and σ is a (prescribed) nonlinear function acting componentwise

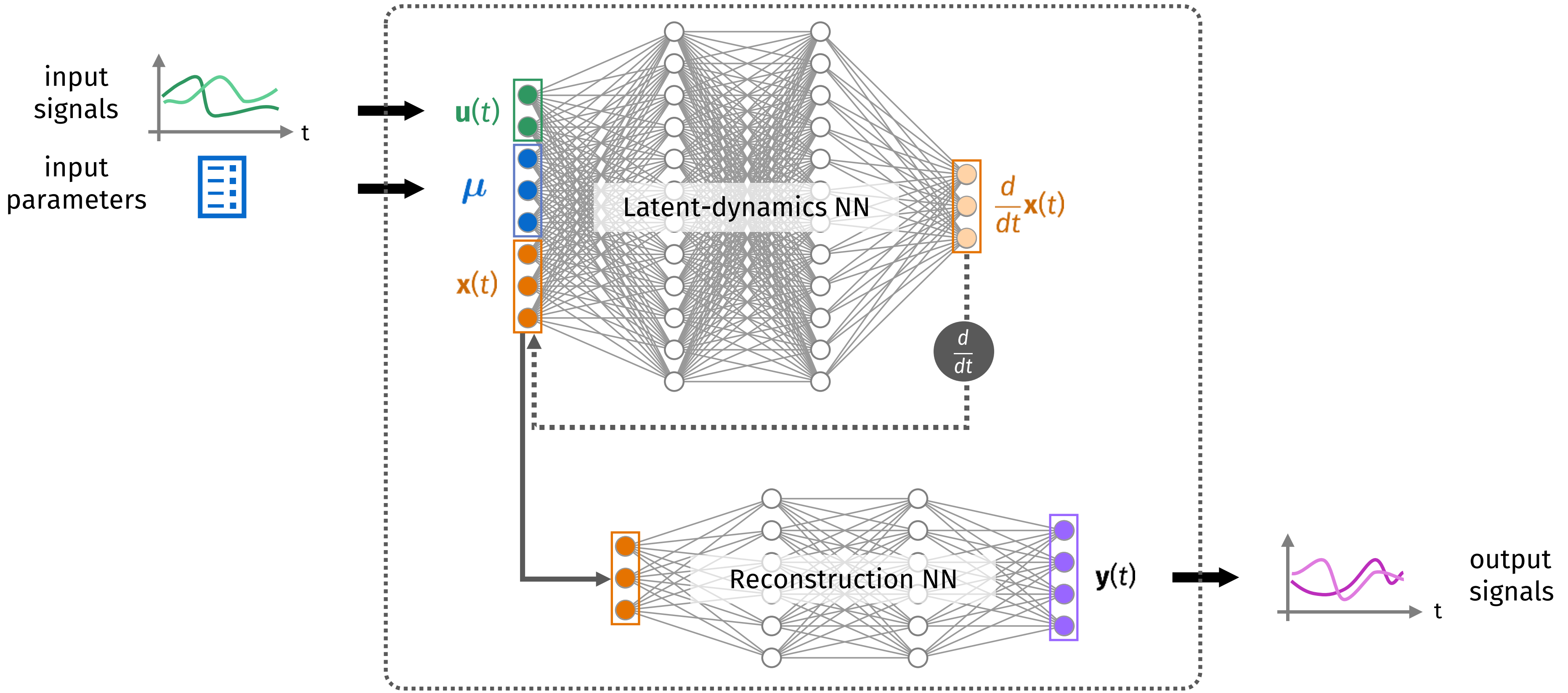
Define: $\mathcal{F}_n^{\mathrm{ANN}}$: the space of ANNs with $n + N_u$ input neurons and n output neurons

$\mathcal{G}_n^{\mathrm{ANN}}$: the space of ANNs with n and N_y input and output neurons

Corollary

If U is compact, the space of ANN models $\Phi^{\mathcal{F}_n^{\mathrm{ANN}}, \mathcal{G}_n^{\mathrm{ANN}}, \mathcal{X}_n}$ is dense in the model space $\Phi^{\mathcal{F}_n, \mathcal{G}_n, \mathcal{X}_n}$

Model-Learning: architecture

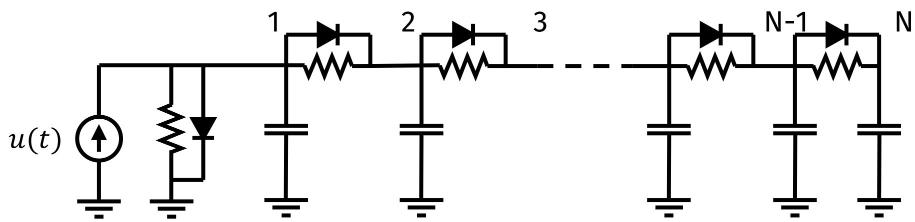


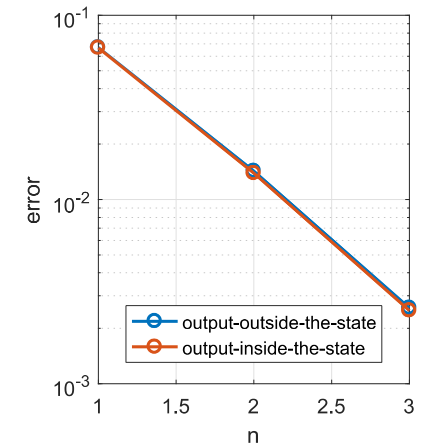
- The recurrent part is formally similar to an ODE-net. However the dynamics of **x**(t) is not known a-priori.
- The latent state **x**(t) provides a compact encoding of the high-fidelity model state. However, the mapping is never explicitly constructed!
- The two ANNs are trained simultaneously: the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

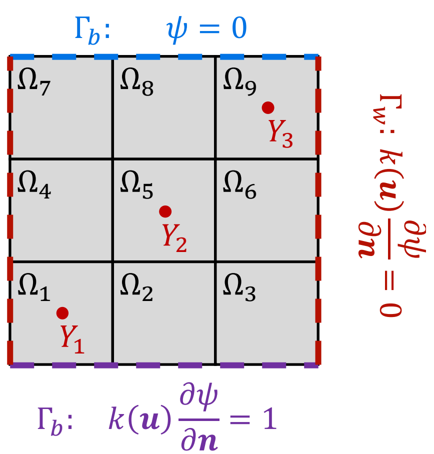
F. Regazzoni, L. Dede', A. Quarteroni "Machine learning for fast and reliable solution of time-dependent differential equations", *Journal of Computational Physics* (2019)

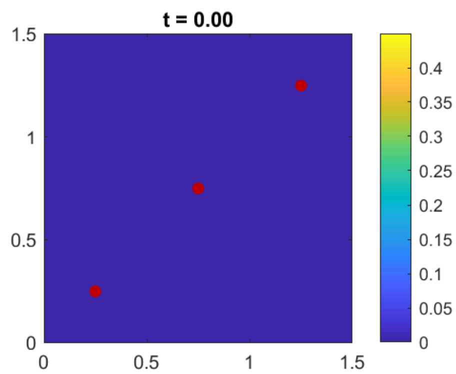
Test cases

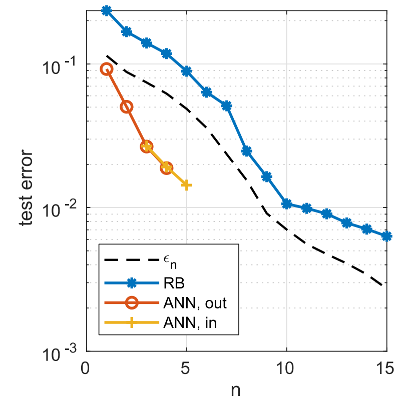
① Systems of nonlinear ODEs









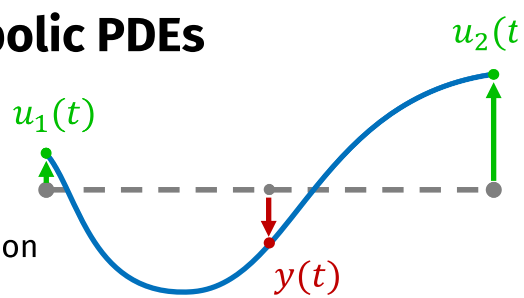


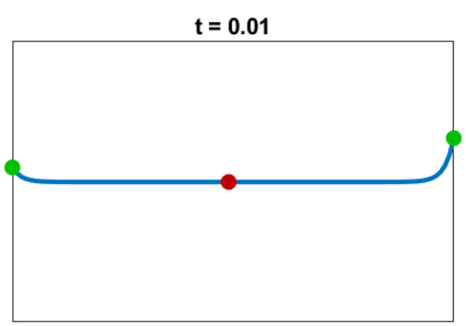
② Parabolic PDEs

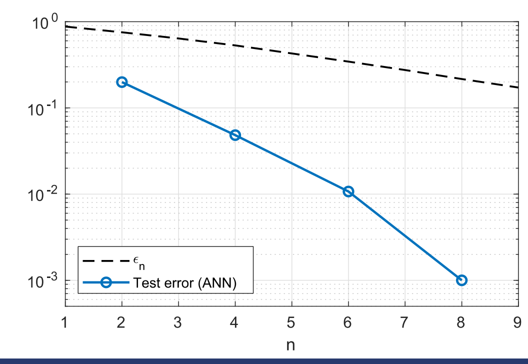
heat equation

③ Hyperbolic PDEs

wave equation







Data-driven modeling of time-dependent processes

- Displacement of a control point
- Generated power of a plant
- Revenues of a company
- Number of infected people
- 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

Full-order model (FOM)

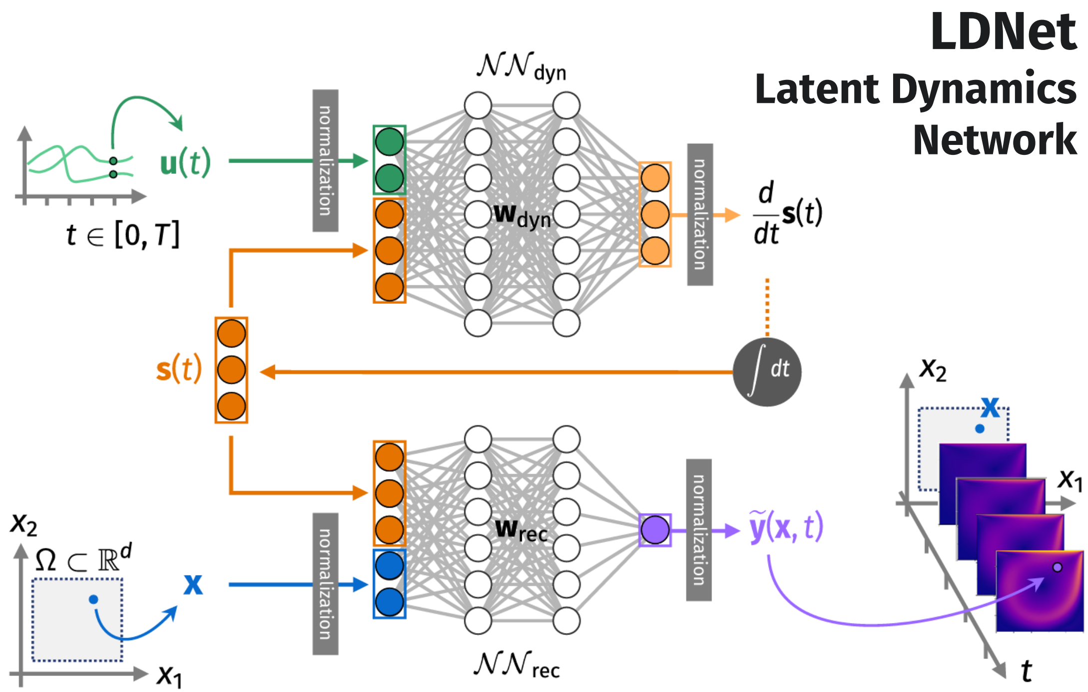
$$\begin{cases}\partial_t \mathbf{z}(\mathbf{x},t) = \mathcal{L}(\mathbf{z}(\mathbf{x},t),\mathbf{u}(t),\boldsymbol{\mu}) & \text{in } \Omega\times(0,T] \\ \mathbf{y}(\mathbf{x},t)=\mathcal{G}(\mathbf{z}(\mathbf{x},t),\mathbf{u}(t),\boldsymbol{\mu},\mathbf{x}) & \text{in } \Omega\times(0,T] \\ \mathbf{z}(\mathbf{x},0)=\mathbf{z}_0 & \text{in } \Omega\end{cases}$$

Reduced-order model (ROM)

$$\begin{cases}\dot{\mathbf{s}}(t) = \mathbf{f}(\mathbf{s}(t),\mathbf{u}(t),\boldsymbol{\mu};\mathbf{w}_\mathbf{f}) & \text{in } (0,T] \\ \tilde{\mathbf{y}}(\mathbf{x},t)=\mathbf{g}(\mathbf{s}(t),\mathbf{u}(t),\boldsymbol{\mu};\mathbf{x};\mathbf{w}_\mathbf{g}) & \text{in } (0,T] \\ \mathbf{s}(0)=\mathbf{0} & \end{cases}$$

Training

$$\underset{\mathbf{w}_\mathbf{f},\mathbf{w}_\mathbf{g}}{\operatorname{argmin}} \sum_{i=1}^{N_s} \sum_{t\in\mathscr{T}_\mathbf{y}} \sum_{\mathbf{x}\in\mathscr{P}_\mathbf{y}} \left|\tilde{\mathbf{y}}^i(\mathbf{x},t)-\mathbf{y}^i(\mathbf{x},t)\right|^2$$



- Latent state **s**(t): low-dimensional encoding of the high-dimensional HF model state
- Low-dimensional manifold 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, arXiv:2305.00094 (2023)

Diffusion-reaction: amplitude + phase

Full-order model (FOM)

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

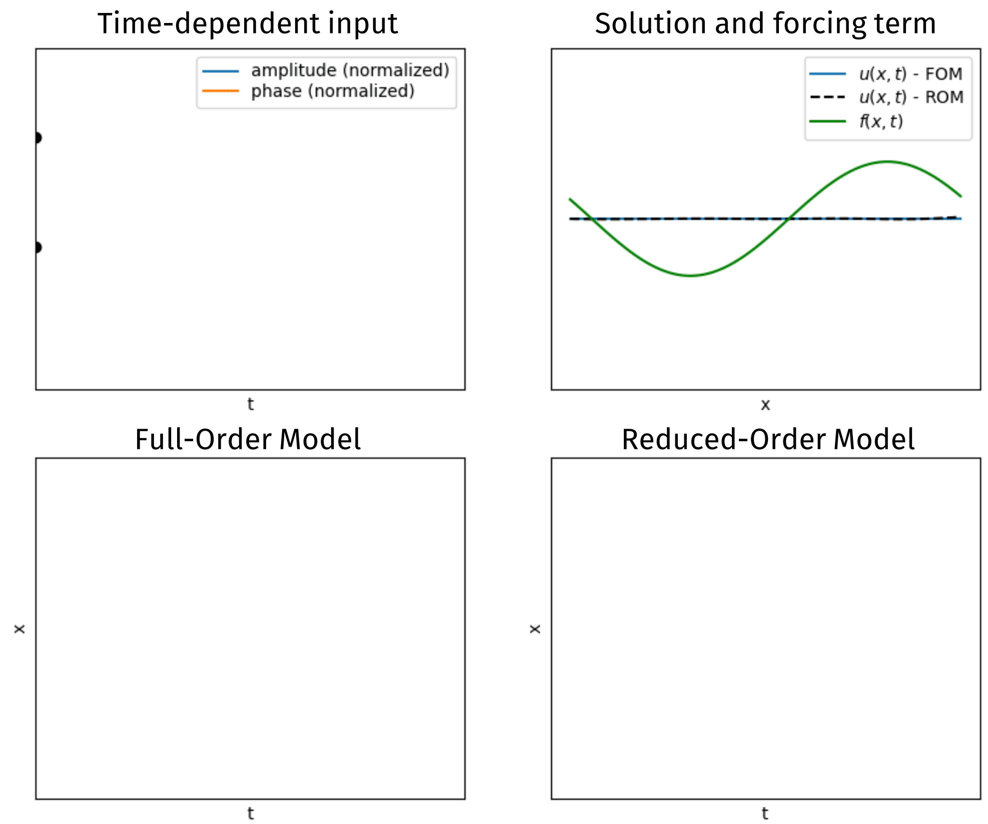
Goal: reconstruct $u(x,t)$

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

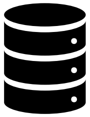
- The solution is a sinusoid with the same period of $f(x,t)$
- At each time, the solution is fully determined by amplitude and phase
- The solution manifold dimension is exactly 2

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

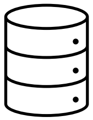
Sample solution (case A)



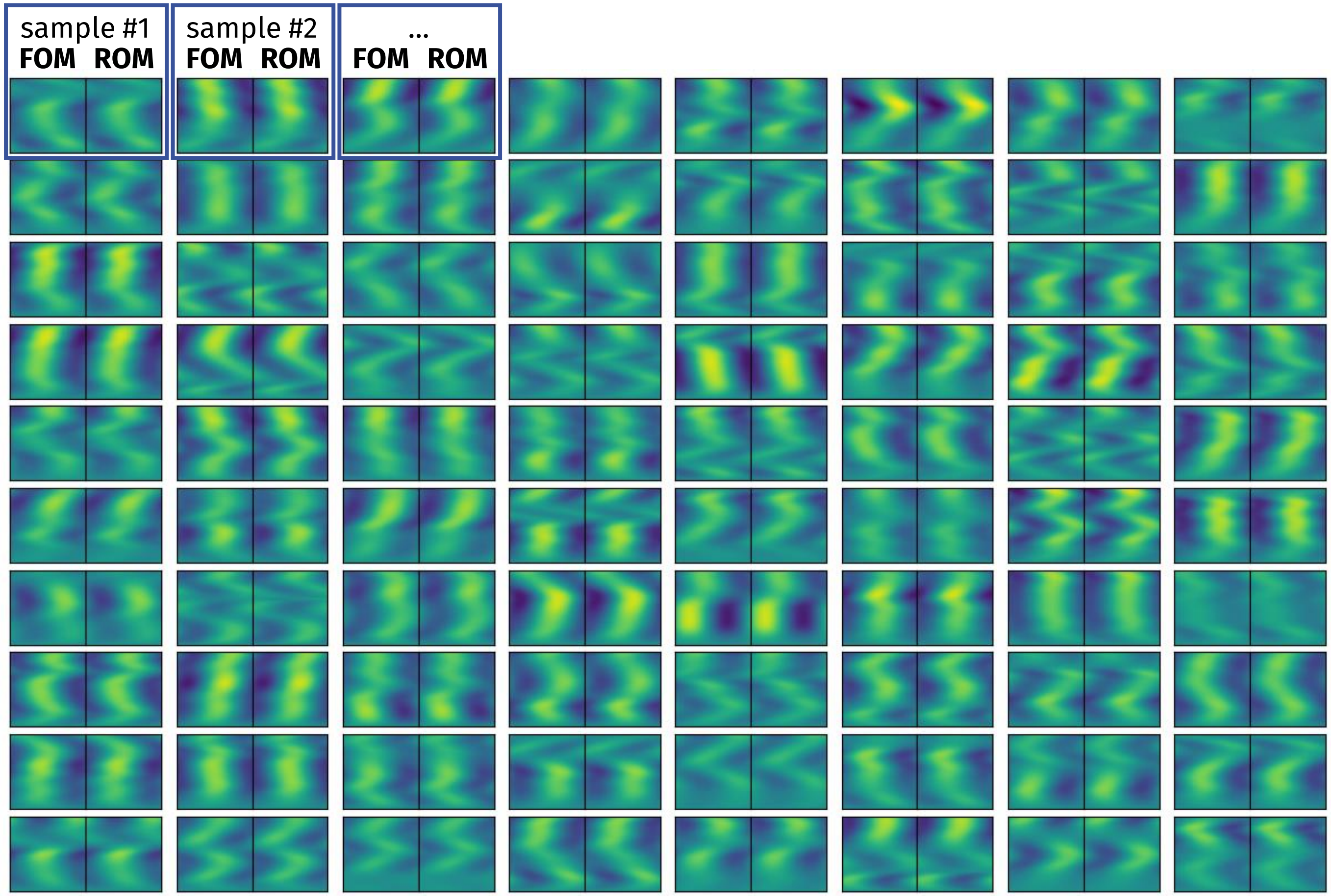
Diffusion-reaction: amplitude + phase

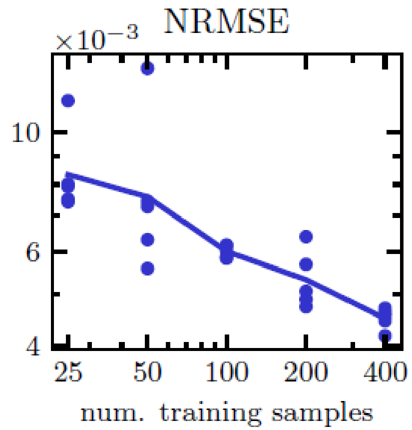


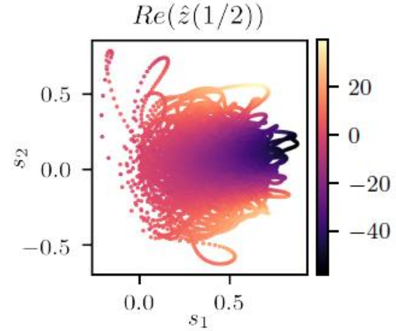
Training/validation set:
- 100 samples
- 100 points in space
- 100 instants in time

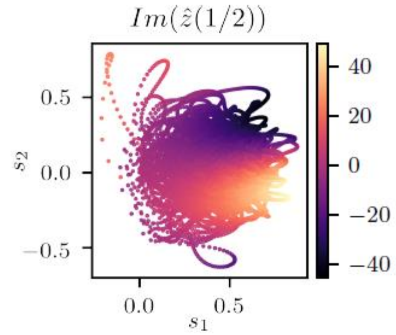


Testing set:
- 1000 samples
- 100 points in space
- 100 instants in time

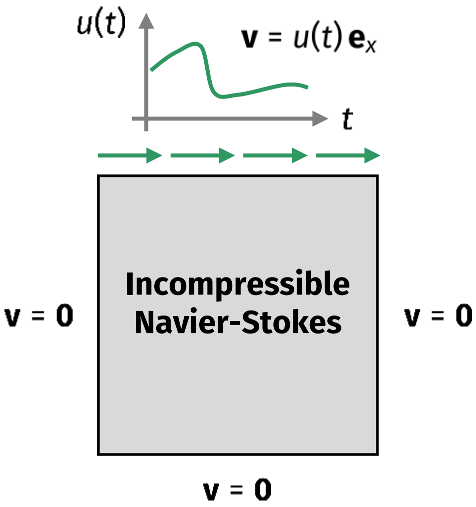








Unsteady Navier-Stokes

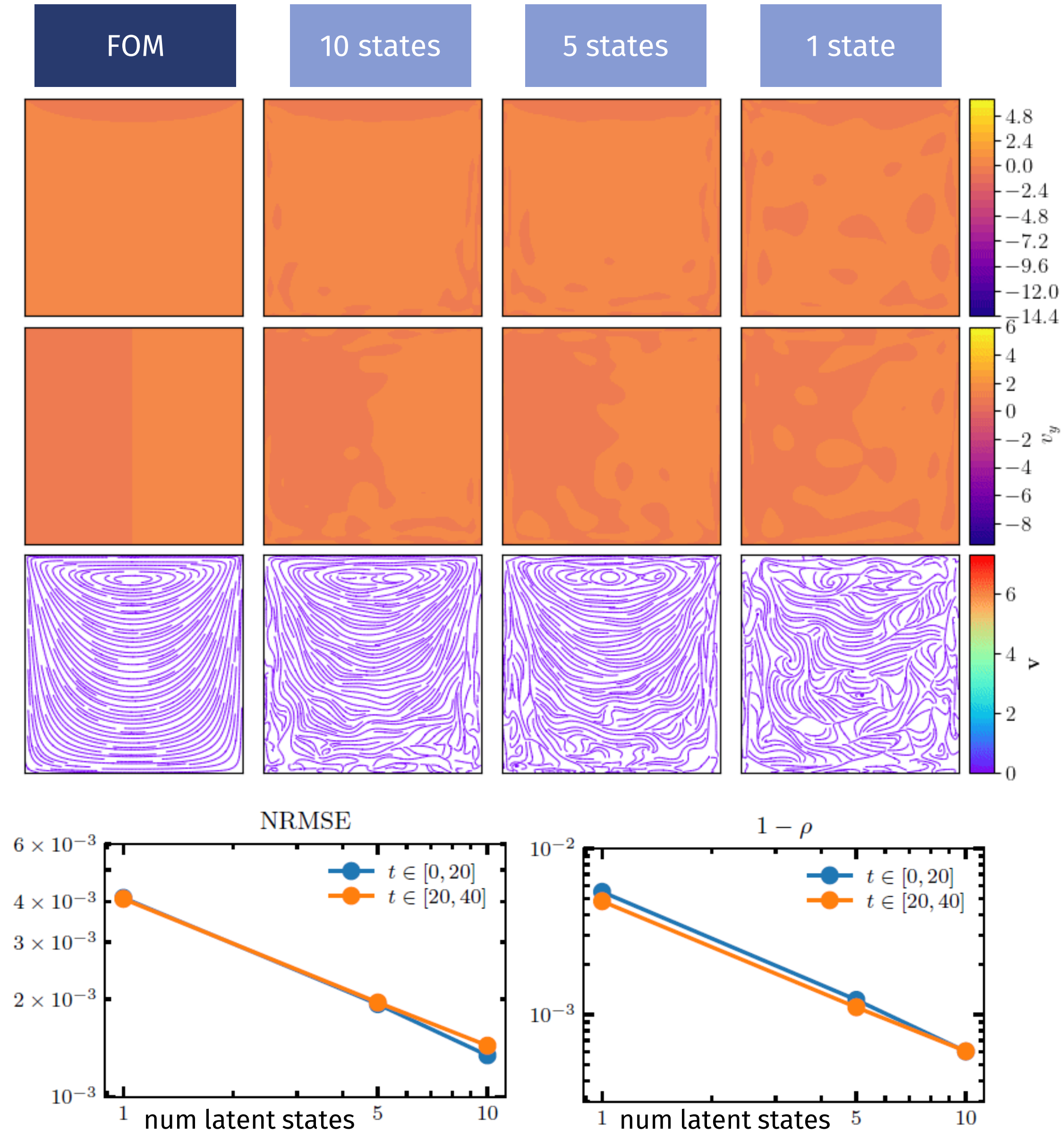


Goal: reconstructing the velocity field $\mathbf{v}(\mathbf{x}, t)$

Training/validation set:
- 80/20 samples
- T=20
- 200 points in $(0, 1)^2$

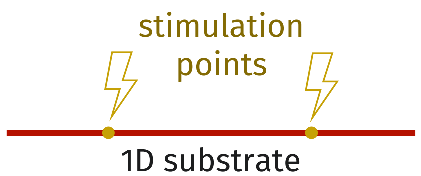
Testing set:
- 200 samples
- T=40 (time extrapolation!)
- 400 points in $(0, 1)^2$

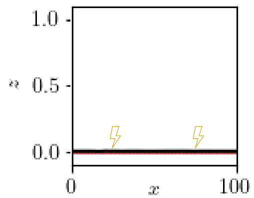
F. Regazzoni, S. Pagani, M. Salvador, L. Dede', A. Quarteroni, arXiv:2305.00094 (2023)



Test case: Aliev Panfilov model

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

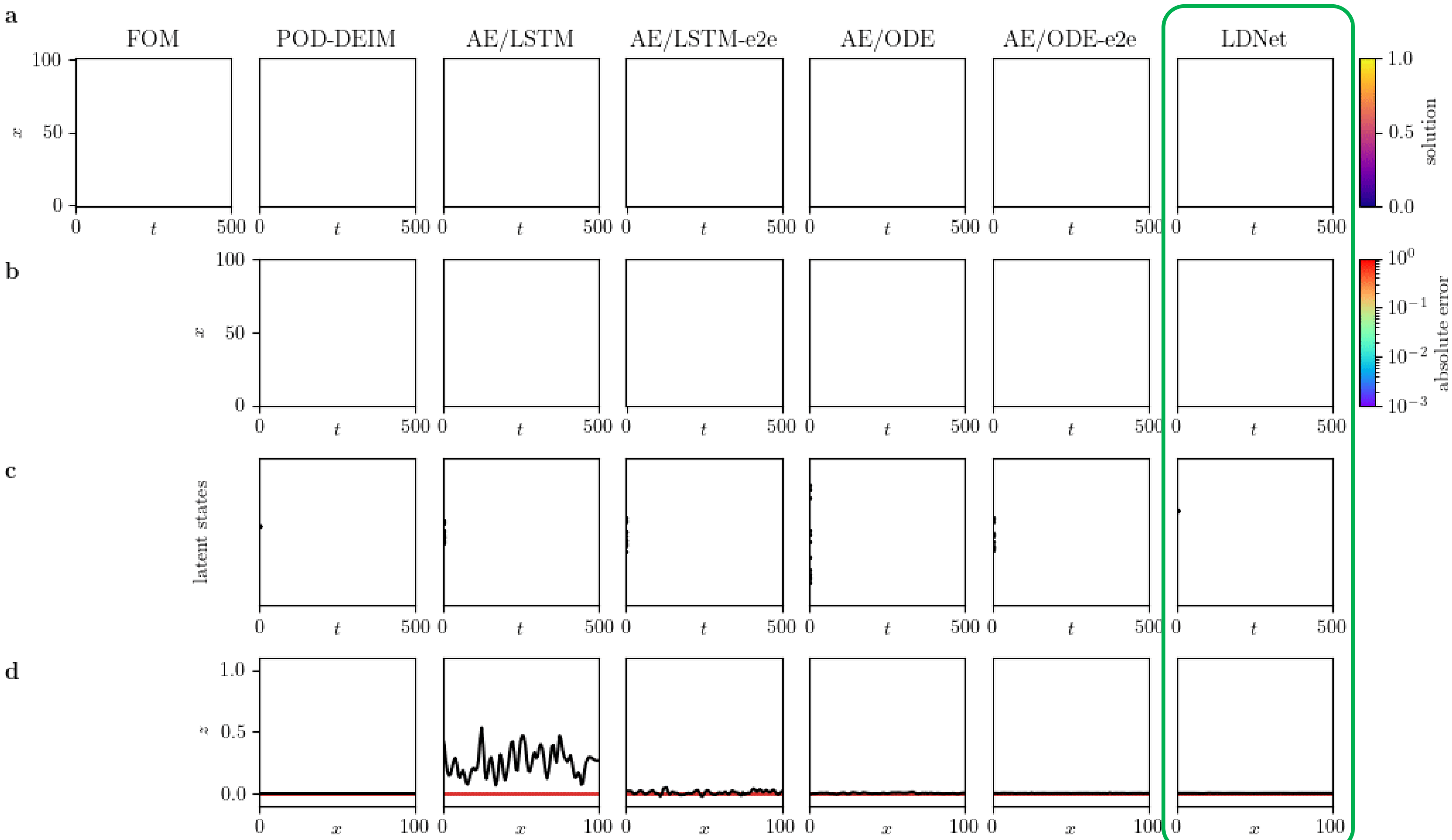




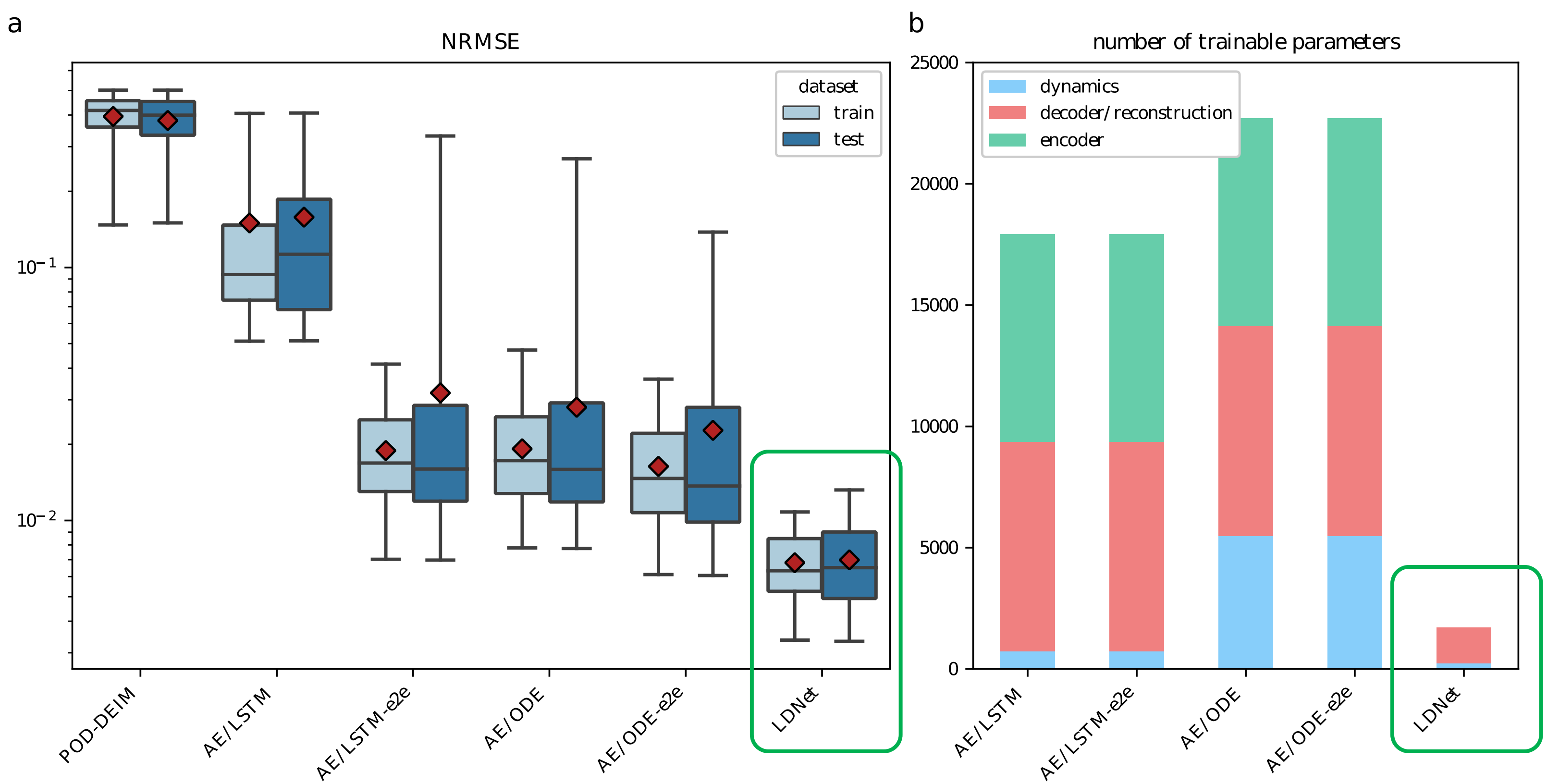
Aliev-Panfilov model

$$\frac{\partial z}{\partial t} - D\frac{\partial^2 z}{\partial x^2} = Kz(1-z)(z-\alpha) - zw + I_{\mathrm{stim}}(x,t) \qquad x \in (0,L),\ t \in (0,T],$$

$$\frac{\partial w}{\partial t} = \left(\gamma + \frac{\mu_1 w}{\mu_2 + z}\right)(-w - Kz(z-b-1)) \qquad x \in (0,L),\ t \in (0,T].$$



Test case: Aliev Panfilov model



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

- ✓ Better accuracy
- ✓ Better generalization (lower overfitting)
- ✓ Fewer trainable parameters

F. Regazzoni, S. Pagani, M. Salvador, L. Dede', A. Quarteroni, arXiv:2305.00094 (2023)

For more information:

- F. Regazzoni, L. Dede', A. Quarteroni "Machine learning for fast and reliable solution of time-dependent differential equations", *Journal of Computational Physics* (2019) 397, 108852.
- F. Regazzoni, L. Dede', A. Quarteroni "Machine learning of active force generation models for the efficient multiscale simulation of the cardiac function", *Computer Methods in Applied Mechanics and Engineering* (2020) 370, 113268.
- F. Regazzoni, D. Chapelle, P. Moireau "Combining data assimilation and machine learning to build data-driven models for unknown long time dynamics—Applications in cardiovascular modeling", International Journal for Numerical Methods in Biomedical Engineering (2021) 37(7), e3471.
- F. Regazzoni, M. Salvador, L. Dede', A. Quarteroni "A Machine Learning method for real-time numerical simulations of cardiac electromechanics " *Computer Methods in Applied Mechanics and Engineering* (2022) 393, 114825.
- F. Regazzoni, S. Pagani, A. Quarteroni. "Universal Solution Manifold Networks (USM-Nets): non-intrusive mesh-free surrogate models for problems in variable domains" *ASME Journal of Biomechanical Engineering* (2022) 144(12): 121004
- F. Regazzoni, S. Pagani, M. Salvador, L. Dede', A. Quarteroni "Latent Dynamics Networks (LDNets): learning the intrinsic dynamics of spatio-temporal processes" arXiv:2305.00094 (2023)

Contact: francesco.regazzoni@polimi.it

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

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

Backup slides

The concept of model

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

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

We call **model** an object mapping inputs $\mathbf{u} \in \mathcal{U}$ into outputs $\mathbf{y} \in \mathcal{Y}$, satisfying the assumptions:

- **Time invariance**
  We can assume, W.L.O.G., all experiments starting from $t_0 = 0$.

- **Existence of an initial state**
  The map is well-defined.

- **Causality principle**
  Consistency with the arrow of time.
  $$\forall \mathbf{u}_1, \mathbf{u}_2 \in \mathcal{U} \quad \forall t^* \in [0,T] \qquad \text{if} \quad \mathbf{u}_1|_{[0,t^*]} = \mathbf{u}_2|_{[0,t^*]} \quad \text{then} \quad (\varphi \mathbf{u}_1)|_{[0,t^*]} = (\varphi \mathbf{u}_2)|_{[0,t^*]} \tag{1}$$

- **No input-output direct dependence**
  The output for $t = 0$ is the same for each experiment.
  $$\exists \mathbf{y}_0 \in Y \quad \text{s.t.} \quad \forall \mathbf{u} \in \mathcal{U} \quad (\varphi \mathbf{u})(0) = \mathbf{y}_0 \tag{2}$$

Definition

Set of all **models**: $\Phi = \{\varphi \colon \mathcal{U} \to \mathcal{Y} \quad \text{s.t. (1) and (2) hold}\}$

$$\|\varphi\|_\Phi = \sup_{\mathbf{u}\in\mathcal{U}} \|\varphi \mathbf{u}\|_{\mathcal{Y}} = \sup_{\mathbf{u}\in\mathcal{U}} \sup_{t\in[0,T]} \|(\varphi \mathbf{u})(t)\|_Y$$

The concept of model

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

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

Definition

We call **model** an object mapping inputs $\mathbf{u} \in \mathcal{U}$ into outputs $\mathbf{y} \in \mathcal{Y}$, satisfying the assumptions:

(1) *Causality principle*: $\forall\, \mathbf{u}_1, \mathbf{u}_2 \in \mathcal{U} \quad \forall\, t^* \in [0,T] \quad \mathbf{u}_1|_{[0,t^*]} = \mathbf{u}_2|_{[0,t^*]} \Rightarrow (\varphi\mathbf{u}_1)|_{[0,t^*]} = (\varphi\mathbf{u}_2)|_{[0,t^*]}$

(2) *Existence of an initial state*: $\exists\, \mathbf{y}_0 \in Y \quad \text{s.t.} \quad \forall\, \mathbf{u} \in \mathcal{U} \quad (\varphi\mathbf{u})(0) = \mathbf{y}_0$

We define the set of all **models**: $\Phi = \left\{\varphi \colon \mathcal{U} \to \mathcal{Y} \quad \text{s.t. (1) and (2) hold}\right\}$

$$\|\varphi\|_\Phi = \sup_{\mathbf{u}\in\mathcal{U}} \|\varphi\mathbf{u}\|_\mathcal{Y} = \sup_{\mathbf{u}\in\mathcal{U}} \sup_{t\in[0,T]} \|(\varphi\mathbf{u})(t)\|_Y$$

Given:

- a collection of input-output pairs: $\left\{(\widehat{\mathbf{u}}_j, \widehat{\mathbf{y}}_j)\right\}_{j=1,\dots,N_s} \subset \mathcal{U} \times \mathcal{Y}$
- a set of candidate (low-dimensional) models: $\widehat{\Phi} \subseteq \Phi$

Best-approximation problem

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

Questions:

- How to select $\widehat{\Phi} \subseteq \Phi$?
- How to solve the best-approximation problem?

Solving the best-approximation problem

We consider, W.L.O.G., the input-outside-the-state approach.

We parametrize **f** and **g** by a finite number of parameters: $\mathbf{f}(\mathbf{x}, \mathbf{u}; \mu)$, $\mathbf{g}(\mathbf{x}; \nu)$

First-Order Optimality Conditions (Lagrange multipliers method)

$$\begin{cases} \dot{\mathbf{x}}_j = \mathbf{f}(\mathbf{x}_j, \hat{\mathbf{u}}; \mu) \\ \mathbf{x}_j(0) = \mathbf{0} \end{cases}$$

Primal (forward) system

$$\begin{cases} -\dot{\mathbf{z}}_j = \nabla_{\mathbf{x}}^T \mathbf{g} \, \left(\hat{\mathbf{y}}_j - \mathbf{g}(\mathbf{x}_j; \nu)\right) + \nabla_{\mathbf{x}}^T \mathbf{f} \, \mathbf{z}_j \\ \mathbf{z}_j(T) = \mathbf{0} \end{cases}$$

Dual (backward) system

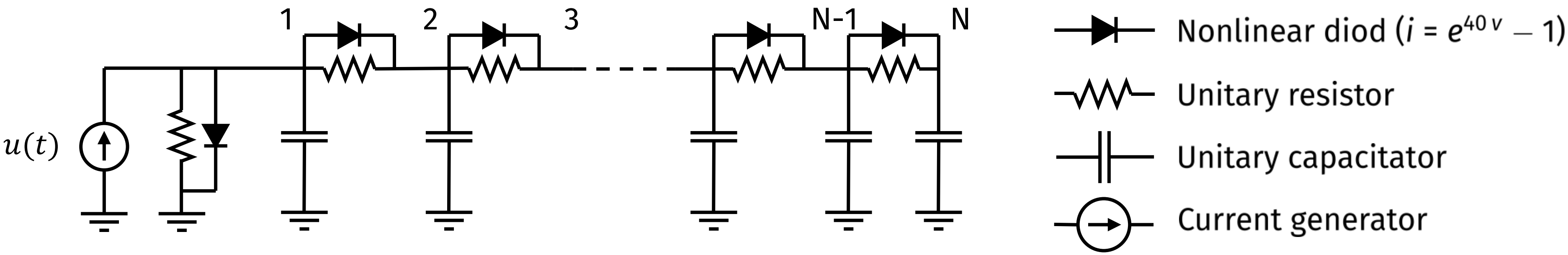
$$\begin{cases} \sum_{j=1}^k \int_0^T \nabla_\mu^T \mathbf{f} \, \mathbf{z}_j dt = \mathbf{0} \\ \sum_{j=1}^k \int_0^T \nabla_\nu^T \mathbf{g} \, \left(\hat{\mathbf{y}}_j - \mathbf{g}(\mathbf{x}_j; \nu)\right) dt = \mathbf{0} \end{cases}$$

Vanishing gradient condition

→ Numerical resolution (Levenberg-Marquardt)

- Discretization of the objective functional by composite trapezoidal rule.
- Discretization of the state-equation by Forward Euler scheme.

Test case 1: Nonlinear transmission line circuit (ODE)

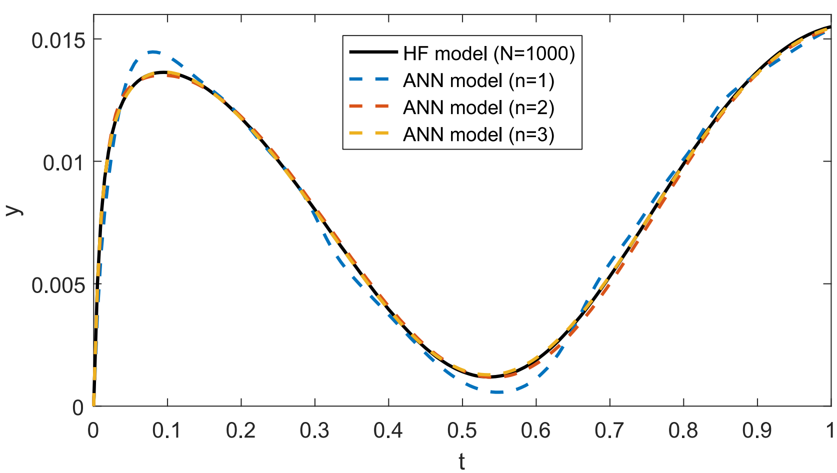


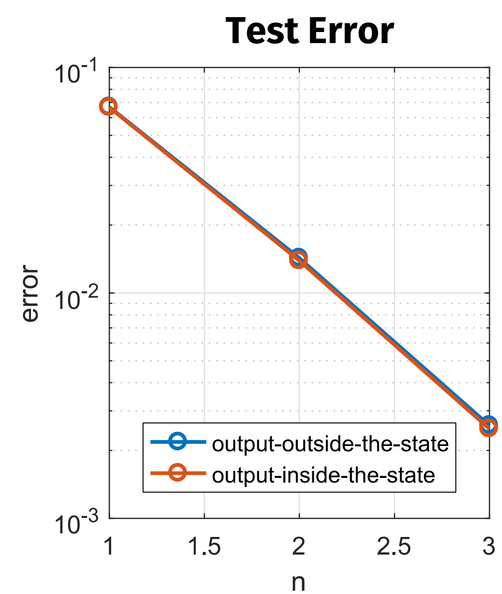
High-fidelity model (N = 1000)

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

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

A sample element of the test set

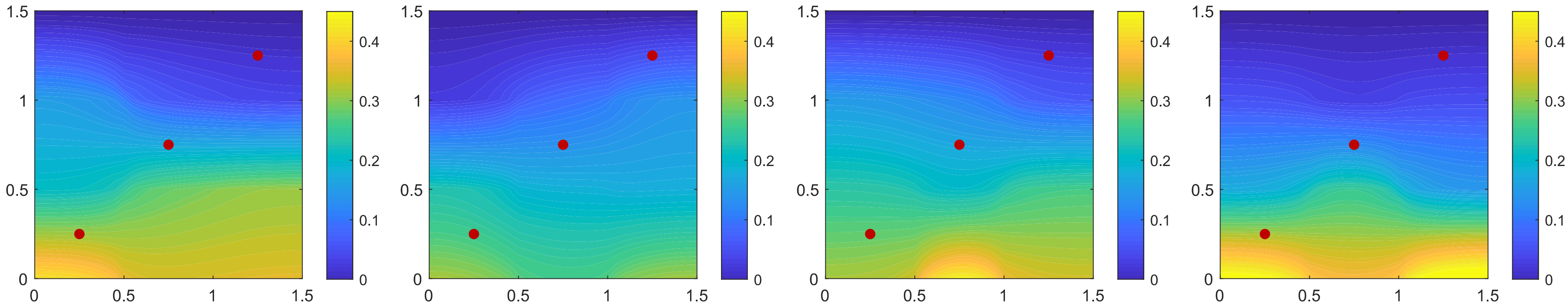




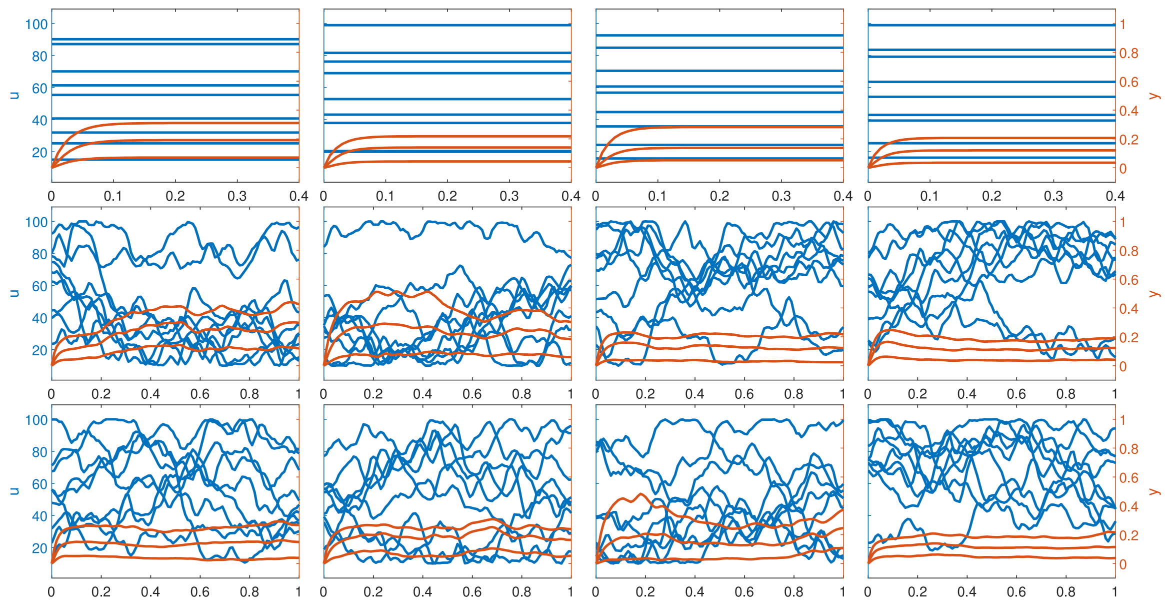
Empirically: exponential convergence w.r.t. number of states

Test case 2: Heat equation (Parabolic PDE)

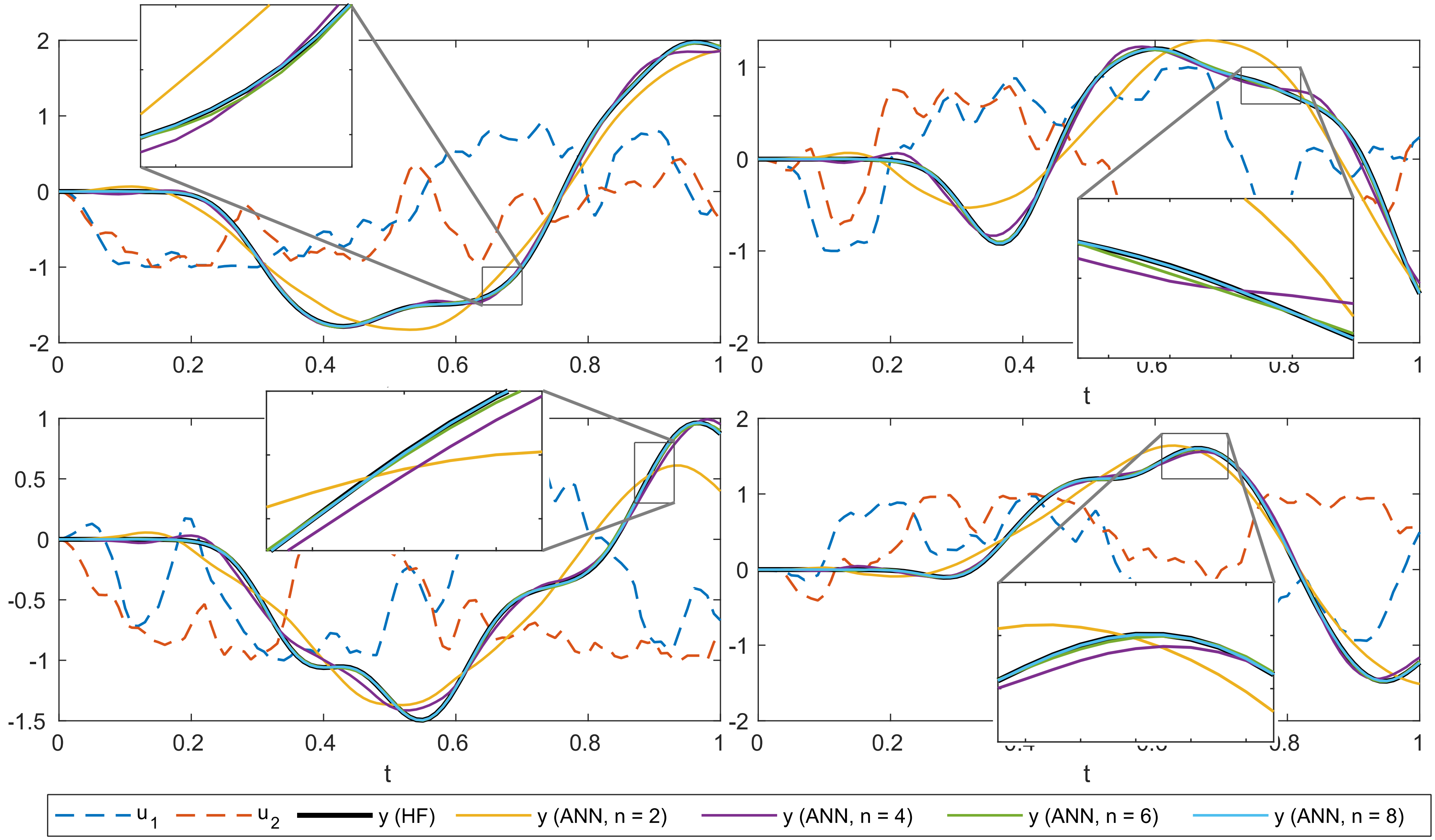
Snapshots of the solution (P2 Finite Element approximation with 3731 dof)



Training Set (subset)



Test case 3: Wave equation (Hyperbolic PDE)



Cardiac electromechanics

Electrophysiology

$$\begin{cases} \frac{\partial v}{\partial t} + I^{\text{ion}}(v,r) = \operatorname{div}(J\mathbf{F}^{-1}\mathbf{D}\mathbf{F}^{-T}\nabla v) - I^{\text{app}} & \text{in } \Omega_0 \times (0,T) \\ \frac{\partial r}{\partial t} = g(v,r) & \text{in } \Omega_0 \times (0,T) \\ (J\mathbf{F}^{-1}\mathbf{D}\mathbf{F}^{-T}\nabla v)\cdot \mathbf{n}_0 = 0 & \text{on } \partial\Omega_0 \times (0,T) \end{cases}$$

Mechanics

$$\begin{cases} \rho\frac{\partial^2 \mathbf{d}}{\partial t} - \operatorname{div}\left(\frac{\partial}{\partial \mathbf{F}}(\mathcal{W} + \mathcal{W}^{\text{vol}}) + \mathbf{P}_a\right) = 0 & \text{in } \Omega_0 \times (0,T) \\ \left(\frac{\partial}{\partial \mathbf{F}}(\mathcal{W} + \mathcal{W}^{\text{vol}}) + \mathbf{P}_a\right)\mathbf{n}_0 = \mathbf{0} & \text{on } \Gamma_N \times (0,T) \\ + \text{ boundary conditions} \end{cases}$$

$$\mathcal{W} = \frac{C}{2}\left(\exp\left(\textstyle\sum_{a,b\in\{f,s,n\}} b_{ab}(\mathbf{E} : \mathbf{a}\otimes\mathbf{b})^2\right) - 1\right)$$

d

c

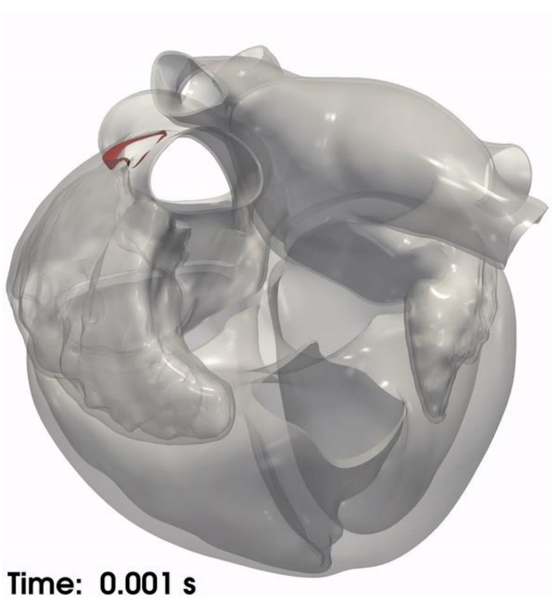
Microscale force generation

$$\frac{\partial \mathbf{p}}{\partial t} = \mathbf{\Phi}(\mathbf{p}, c, SL) \quad \text{in } \Omega_0 \times (0,T)$$

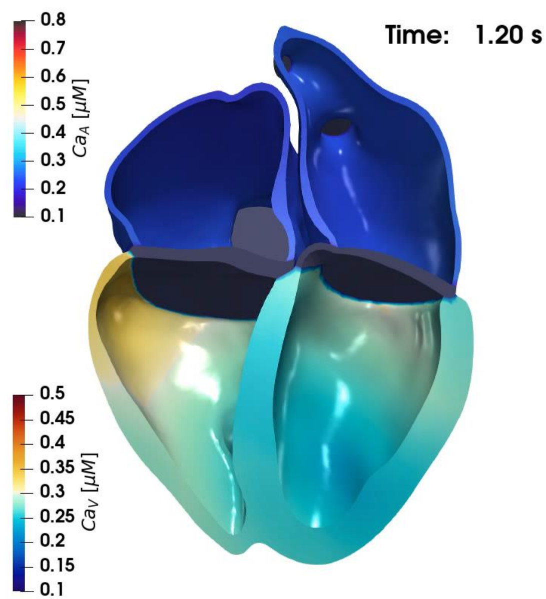
$$SL = SL_0\sqrt{\mathcal{I}_{4f}}$$

$$\mathbf{P}_a = F_{\text{max}}\, P(\mathbf{p})\,\frac{\mathbf{F}\mathbf{f}_0 \otimes \mathbf{f}_0}{\sqrt{\mathcal{I}_{4,f}}}$$

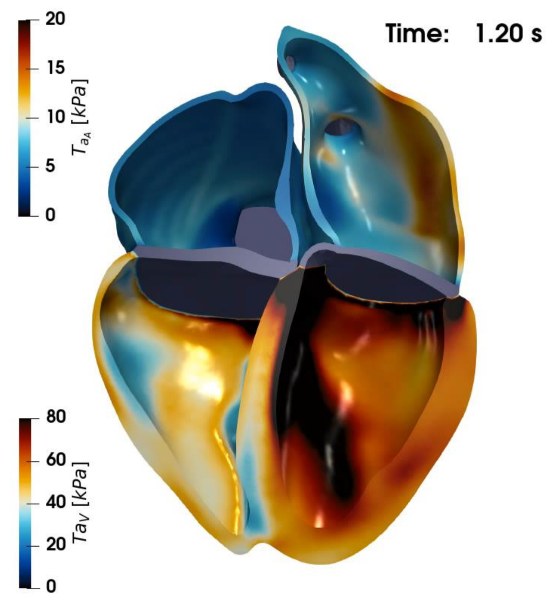
wave-front propagation



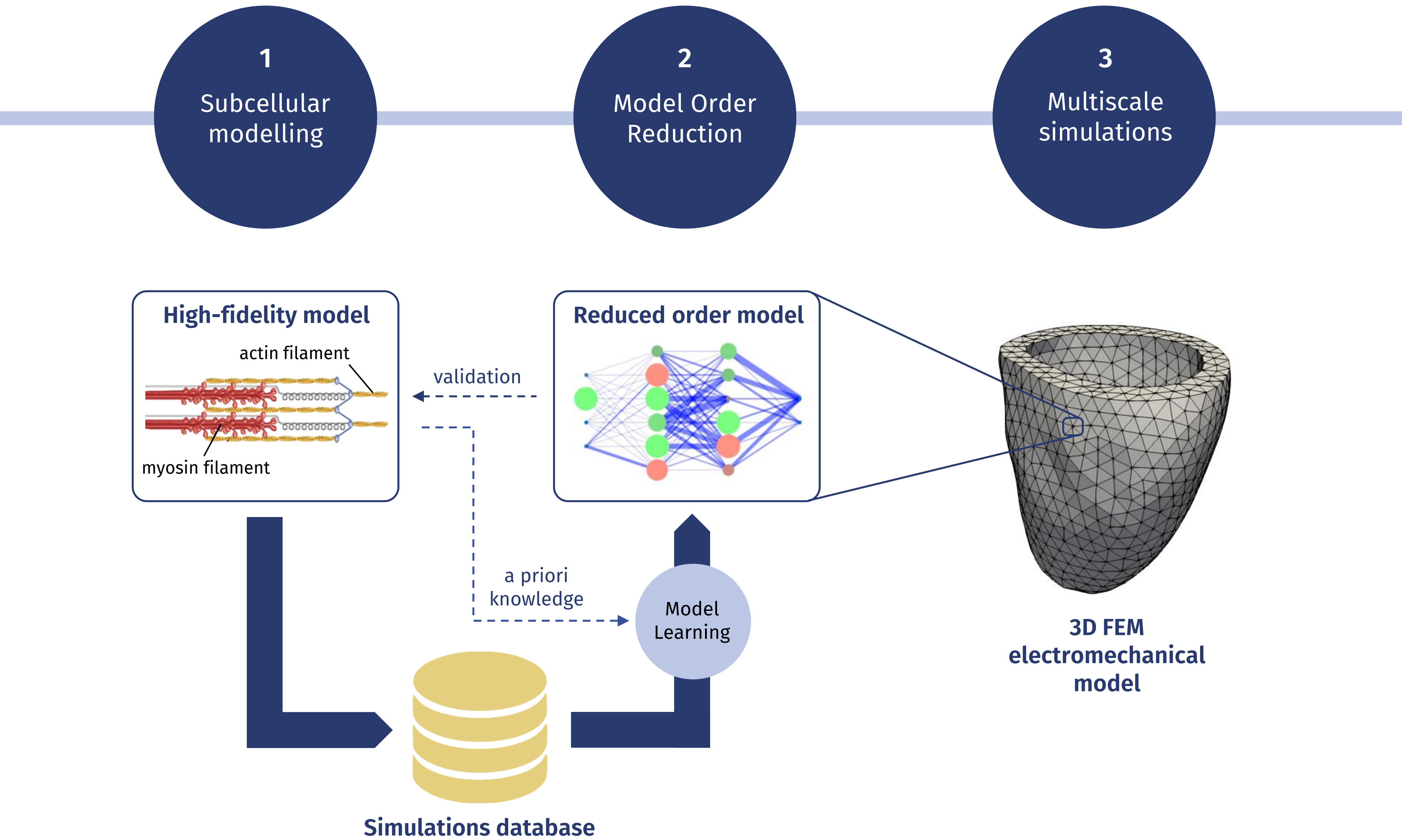
calcium concentration



active stress

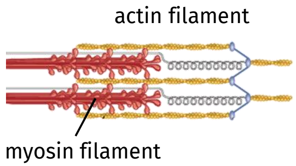


Model-Learning: application to multiscale problems



ANN-based cardiac force generation model

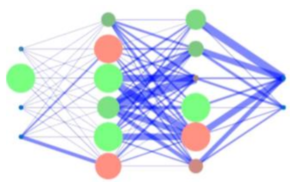
High-fidelity model



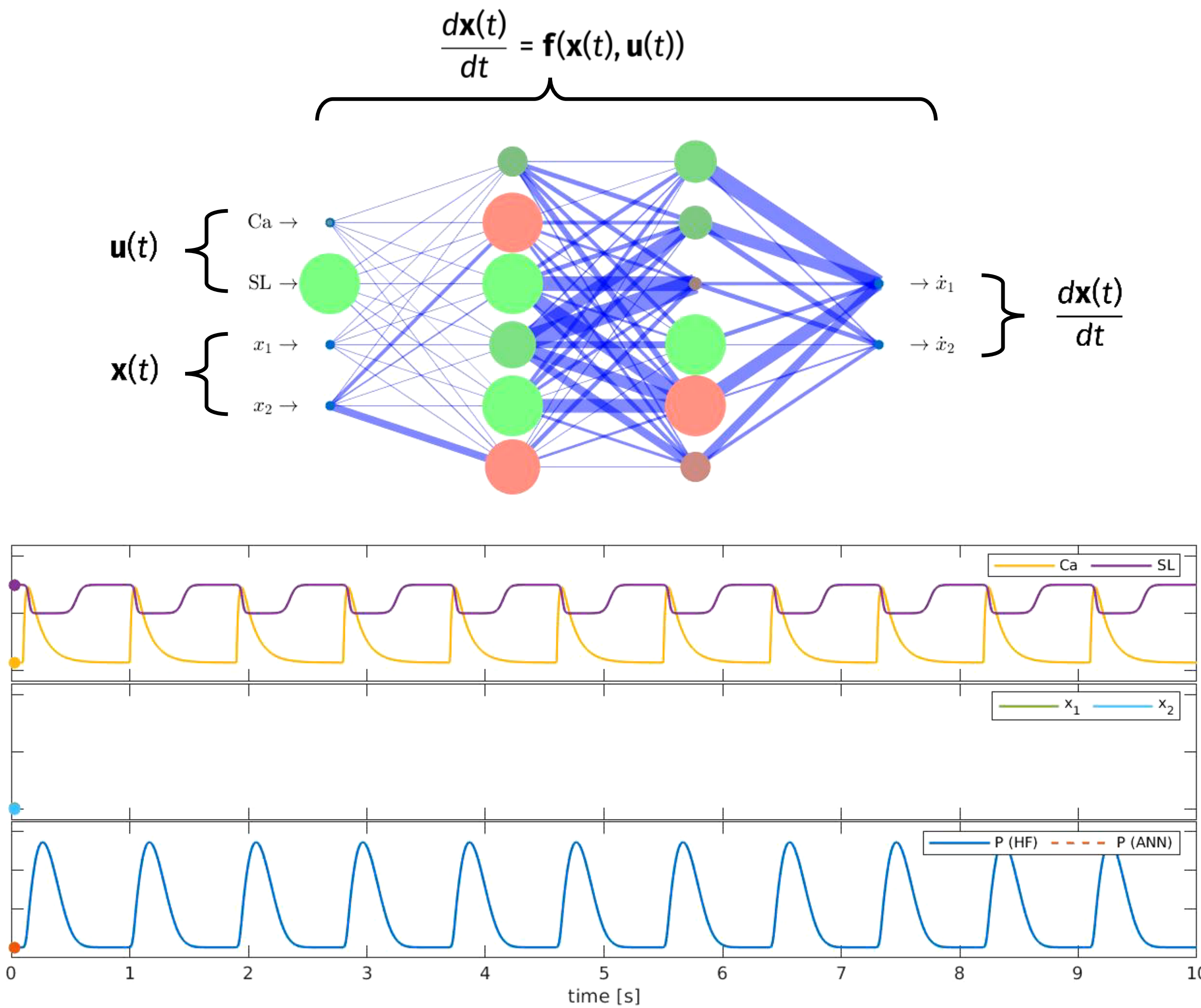
2174 state variables

150 training samples

Reduced order model



2 state variables



Results:

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

F. Regazzoni, L. Dedé, A. Quarteroni, *Computer Methods in Applied Mechanics and Engineering*, 2020

Semi-physical (grey-box) MOR

Black-box cost functional:

$$E_{\mathrm{d}}^2 = a_{\mathrm{d}}^{-1} \sum_{j=1}^{N_s} \int_0^{T_j} |\widehat{y}_j(t) - y_j(t)|^2 dt$$

A priori knowledge from the HF model:

$$E_{\mathrm{g}}^2 = a_{\mathrm{g}}^{-1} \sum_{j \in J_r} \sum_{i=2}^{n} \frac{\left(\mathbf{x}_j(T_j) \cdot \mathbf{e}_i\right)^2}{\frac{1}{T_j} \int_0^{T_j} \left(\mathbf{x}_j(t) \cdot \mathbf{e}_i\right)^2 \, \mathrm{d}t}$$

for some t^*, $\mathbf{X}(t^*) = \mathbf{X}_0 \quad \Rightarrow \quad \mathbf{x}(t^*) = \mathbf{x}_0$

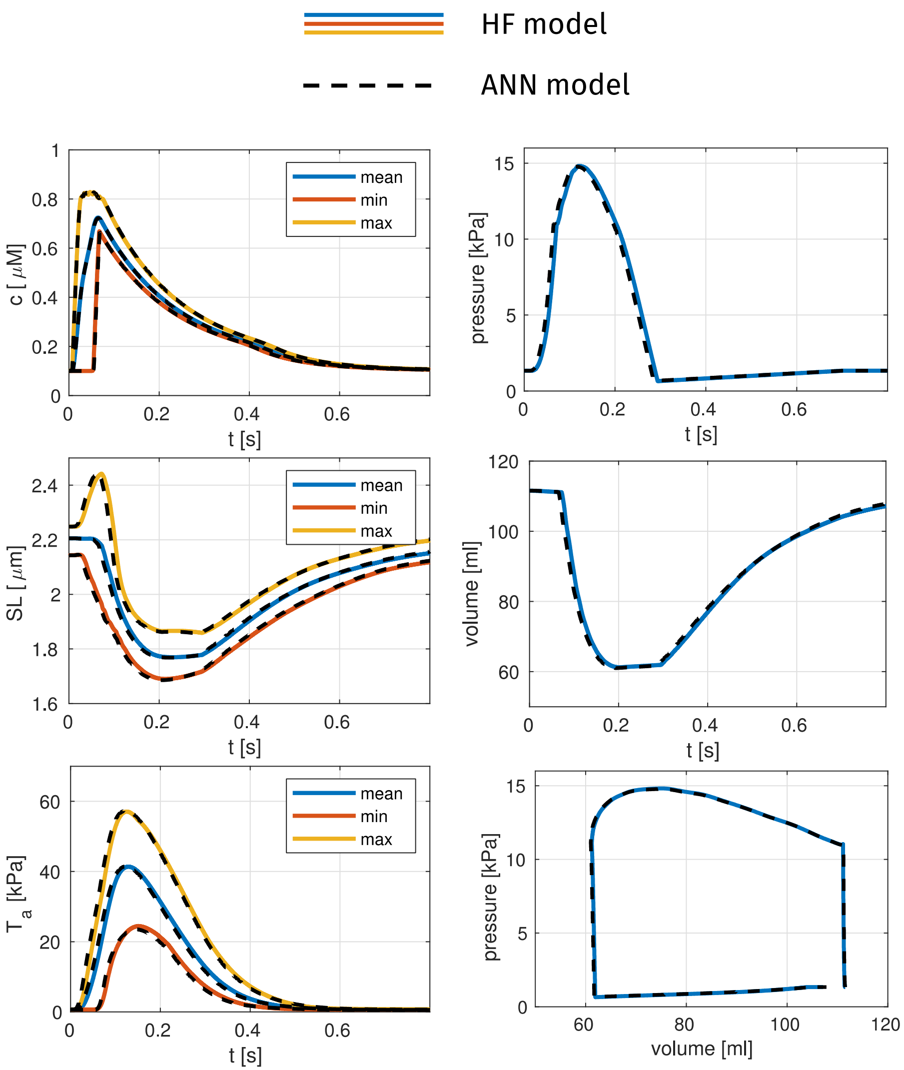
(for $j \in J_r$ we have $\mathbf{X}(T_j) = \mathbf{X}_0$)

$$E_{\mathrm{e}}^2 = a_{\mathrm{e}}^{-1} |\mathbf{f}(\mathbf{x}_0, \mathbf{u}_0)|^2$$

$(\mathbf{X}_0, \mathbf{u}_0)$ equilibrium for $\mathbf{F} \quad \Rightarrow \quad (\mathbf{x}_0, \mathbf{u}_0)$ equilibrium for $\mathbf{f}$

$$\begin{cases} \min\limits_{\mu \in \mathbb{R}^w} & \frac{1}{2} w_{\mathrm{d}}^2 E_{\mathrm{d}}^2 + \frac{1}{2} w_{\mathrm{g}}^2 E_{\mathrm{g}}^2 + \frac{1}{2} w_{\mathrm{e}}^2 E_{\mathrm{e}}^2 \\ \text{s.t.} & \dot{\mathbf{x}}_j(t) = \mathbf{f}(\mathbf{x}_j(t), \hat{\mathbf{u}}_j(t); \mu), \quad t \in (0, T_j], \quad j = 1, \dots, N_s \\ & y_j(t) = \mathbf{x}_j(t) \cdot \mathbf{e}_1, \quad t \in (0, T_j], \quad j = 1, \dots, N_s \\ & \mathbf{x}_j(0) = \mathbf{x}_0, \quad j = 1, \dots, N_s. \end{cases}$$

HF-Electromechanics vs ANN-Electromechanics



Accuracy

Indicator	HF-EM	ANN-EM	Relative error
Stroke volume (mL)	58.45	58.42	$5.64 \cdot 10^{-4}$
Ejection fraction (%)	43.03	43.01	$5.65 \cdot 10^{-4}$
Max pressure (mmHg)	112.5	112.3	$2.18 \cdot 10^{-3}$
Work (mJ)	739.2	737.2	$1.71 \cdot 10^{-3}$

Computational time (20 cores)

	Ionic	Potential	Force gen.	Mechanics	Total
HF-EM	3.13 %	0.47 %	83.07 %	13.33 %	20h 18’
ANN-EM	41.21 %	4.80 %	2.54 %	51.45 %	2h’ 03’

400 x speedup (force generation model)

10 x speedup (overall)

Memory usage

from 2198 (HF-EM) to 24 (ANN-EM) variables per nodal point

100 x memory saving

Dealing with non-uniqueness

Question: is the solution unique? **In general, no!**

Suppose that the triplet $(\mathbf{f}, \mathbf{g}, \mathbf{x}_0)$ is a solution. Let $\mathbf{h}\colon \mathbb{R}^n \to \mathbb{R}^n$ be invertible and sufficiently regular.

$$\begin{aligned} \widetilde{\mathbf{f}}(\widetilde{\mathbf{x}}, \mathbf{u}) &= (\nabla \mathbf{h} \circ \mathbf{h}^{-1})(\widetilde{\mathbf{x}})\ \mathbf{f}(\mathbf{h}^{-1}(\widetilde{\mathbf{x}}), \mathbf{u}) \\ \widetilde{\mathbf{g}}(\widetilde{\mathbf{x}}) &= \mathbf{g}(\mathbf{h}^{-1}(\widetilde{\mathbf{x}})) \\ \widetilde{\mathbf{x}}_0 &= \mathbf{h}(\mathbf{x}_0). \end{aligned} \quad \longrightarrow \quad \varphi_{\mathbf{f},\mathbf{g},\mathbf{x}_0} = \varphi_{\widetilde{\mathbf{f}},\widetilde{\mathbf{g}},\widetilde{\mathbf{x}}_0}$$

Idea: reduce $\widehat{\mathcal{F}}$, $\widehat{\mathcal{G}}$ and/or $\widehat{\mathcal{X}}$, to rule out (or reduce) ambiguity of representation (e.g. $\widetilde{\mathbf{x}} = \mathbf{h}_1(\mathbf{x}) = \mathbf{x} - \mathbf{x}_0$).

(a) Input-outside-the-state approach

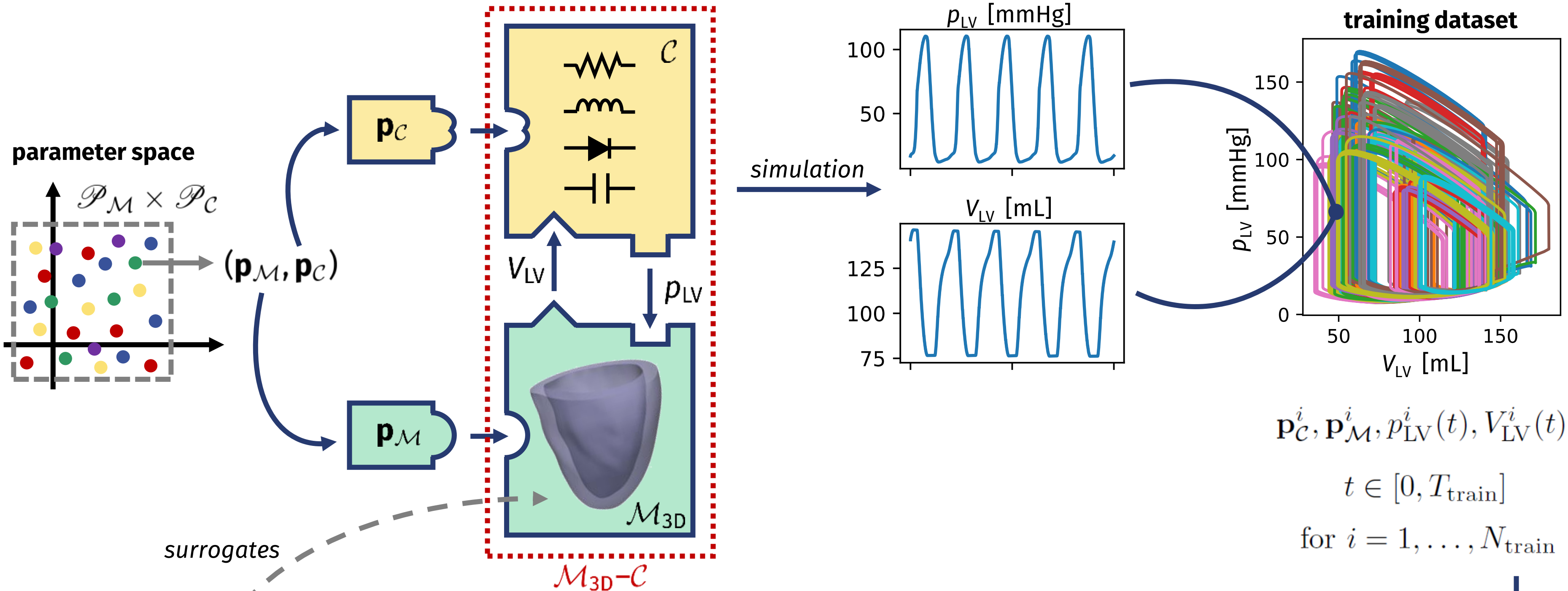
$$\begin{cases} \min\limits_{\mathbf{f}\in\widehat{\mathcal{F}}, \mathbf{g}\in\widehat{\mathcal{G}}} & \frac{1}{2}\sum_{j=1}^{N_s}\int_0^T |\widehat{\mathbf{y}}_j(t) - \mathbf{g}(\mathbf{x}_j(t))|^2 dt \\ \text{s.t.} & \dot{\mathbf{x}}_j(t) = \mathbf{f}(\mathbf{x}_j(t), \widehat{\mathbf{u}}_j(t)), \quad t \in (0, T], \quad j = 1, \dots, N_s \\ & \mathbf{x}_j(0) = \mathbf{0}, \quad j = 1, \dots, N_s, \end{cases}$$

(b) Input-inside-the-state approach

$$\begin{cases} \min\limits_{\mathbf{f}\in\widehat{\mathcal{F}}} & \frac{1}{2}\sum_{j=1}^{N_s}\int_0^T |\widehat{\mathbf{y}}_j(t) - \pi^{N_y}(\mathbf{x}_j(t))|^2 dt \\ \text{s.t.} & \dot{\mathbf{x}}_j(t) = \mathbf{f}(\mathbf{x}_j(t), \widehat{\mathbf{u}}_j(t)), \quad t \in (0, T], \quad j = 1, \dots, N_s \\ & \mathbf{x}_j(0) = (\mathbf{y}_0^T, \mathbf{0}^T)^T, \quad j = 1, \dots, N_s, \end{cases}$$

where $\pi^{N_y}(\mathbf{x}) = (x_1, x_2, \dots, x_{N_y})^T$

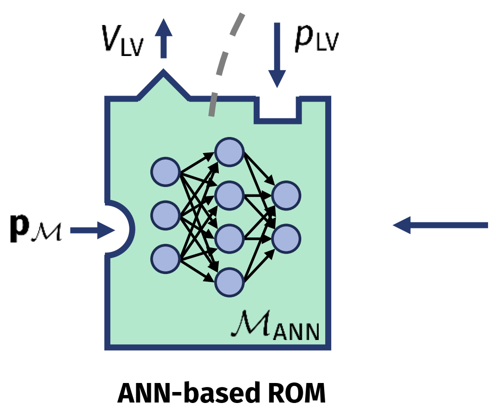
A neural network based surrogate model of the LV function



$$\mathbf{p}_\mathcal{C}^i, \mathbf{p}_\mathcal{M}^i, p_{LV}^i(t), V_{LV}^i(t)$$

$$t \in [0, T_{\text{train}}]$$

$$\text{for } i = 1, \dots, N_{\text{train}}$$

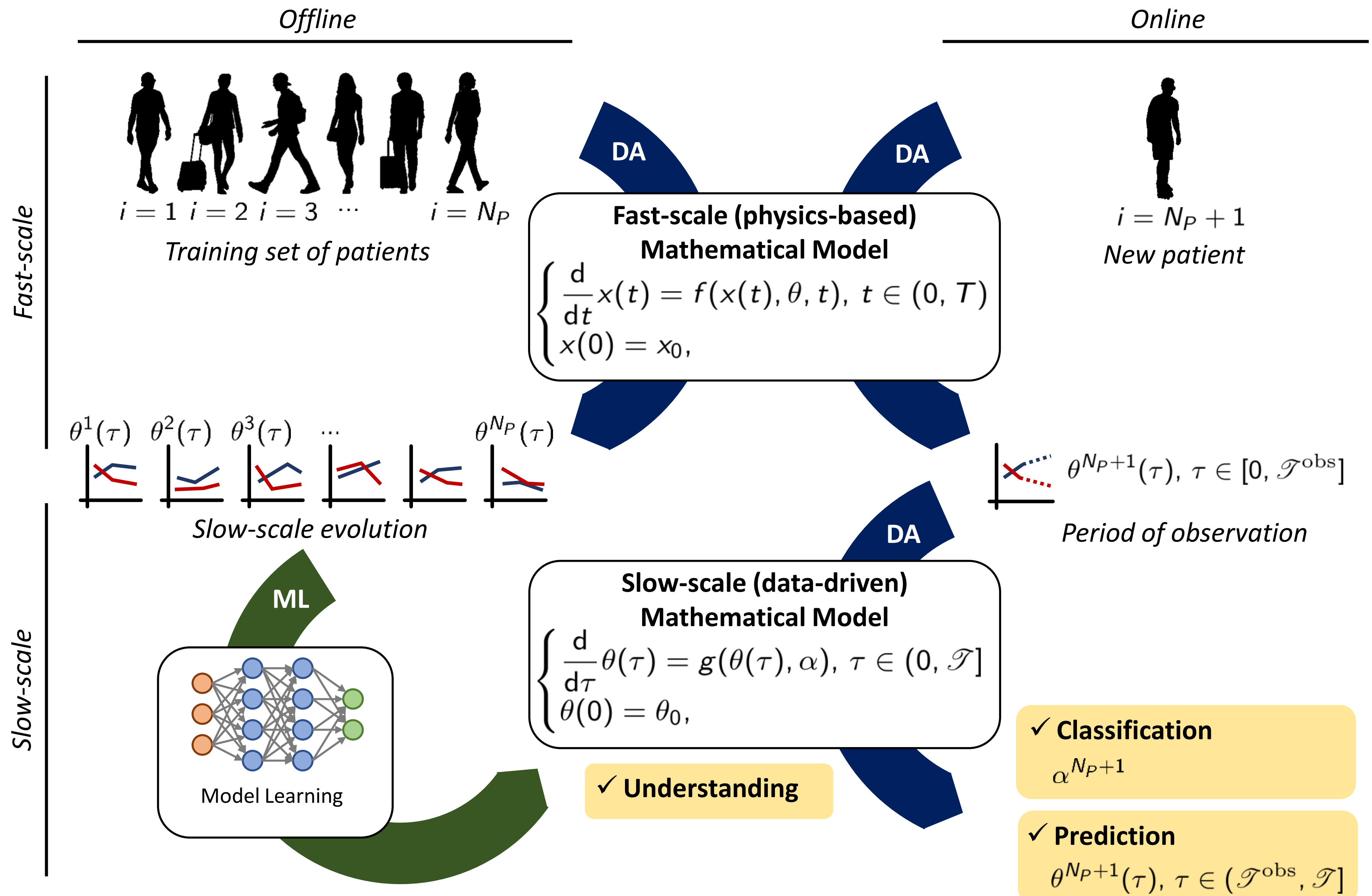


Model Learning

$$\begin{cases} \widehat{\mathbf{w}} = \underset{\mathbf{w} \in \mathbb{R}^{N_w}}{\operatorname{argmin}} \left[\sum_{i=1}^{N_{\text{train}}} \int_0^{T_{\text{train}}} |V_{LV}^i(t) - V_{LV}^{ANN}(\mathbf{z}^i(t))|^2 \, dt + \beta |\mathbf{w}|^2 \right] \\ \text{such that, for each } i = 1, \dots, N_{\text{train}}: \\ \qquad \frac{d\mathbf{z}^i(t)}{dt} = \mathcal{NN}\left(\mathbf{z}^i(t), p_{LV}^i(t), \cos(\frac{2\pi t}{T_{HB}}), \sin(\frac{2\pi t}{T_{HB}}); \mathbf{p}_\mathcal{M}^i; \mathbf{w} \right) \quad \text{for } t \in (0, T_{\text{train}}], \\ \qquad \mathbf{z}^i(0) = \mathbf{z}_0, \end{cases}$$

F. . Rregazzoni, L. Dede', A. Quarteroni "Machine learning for fast and reliable solution of time-dependent differential equations“, *Journal of Computational Physics* (2019) 397, 108852

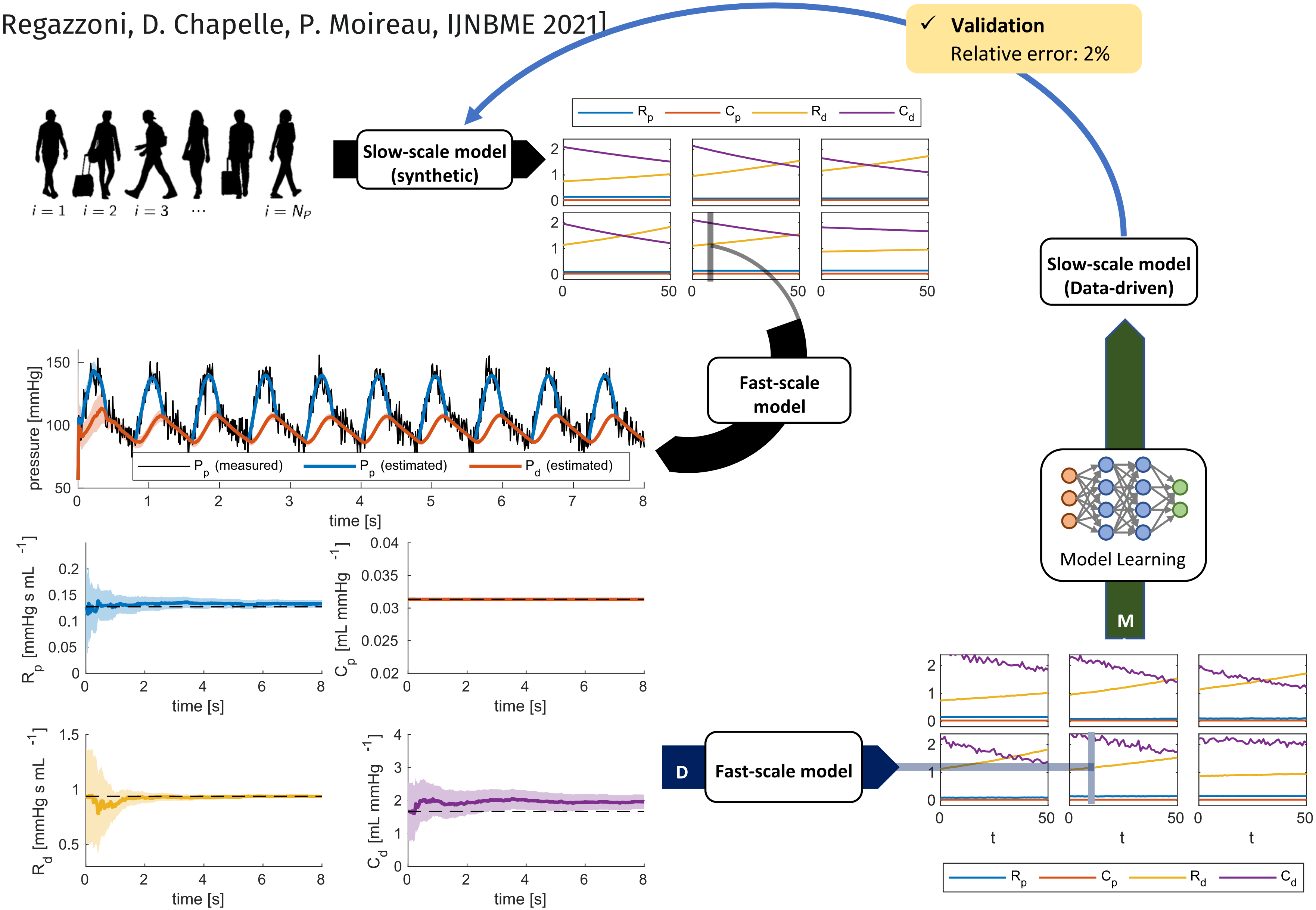
Data-driven model of slow-scale processes



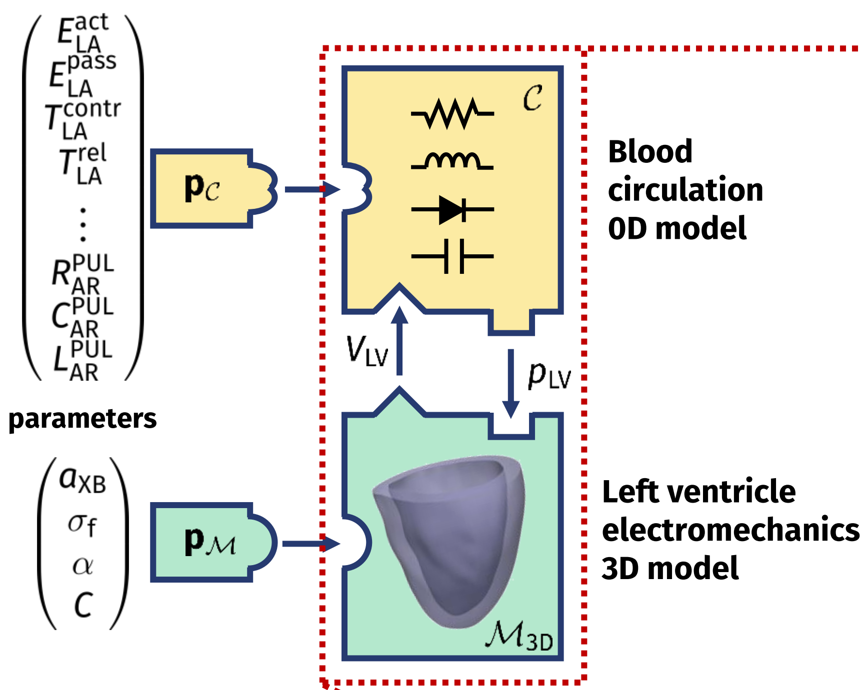
[F.R., D. Chapelle, P. Moireau, IJNBME 2021]

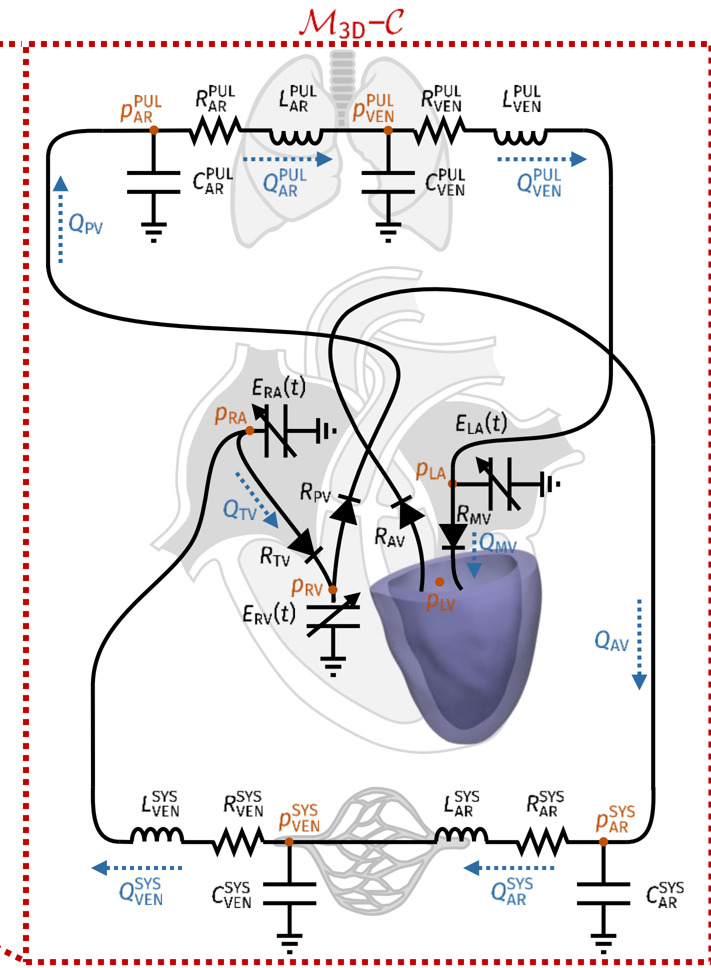
Application to hypertension

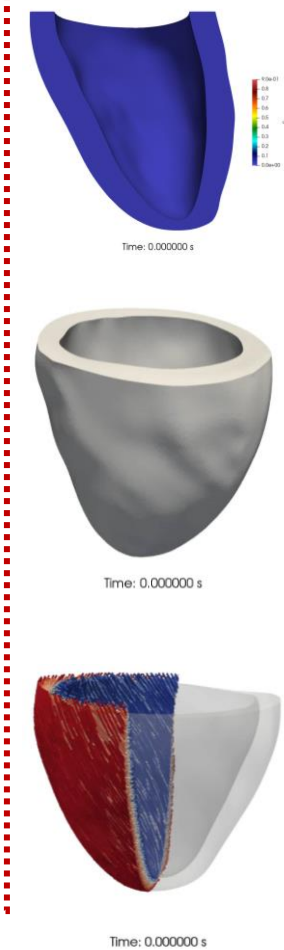
[F.Regazzoni, D. Chapelle, P. Moireau, IJNBME 2021]



The high-fidelity cardiocirculatory model

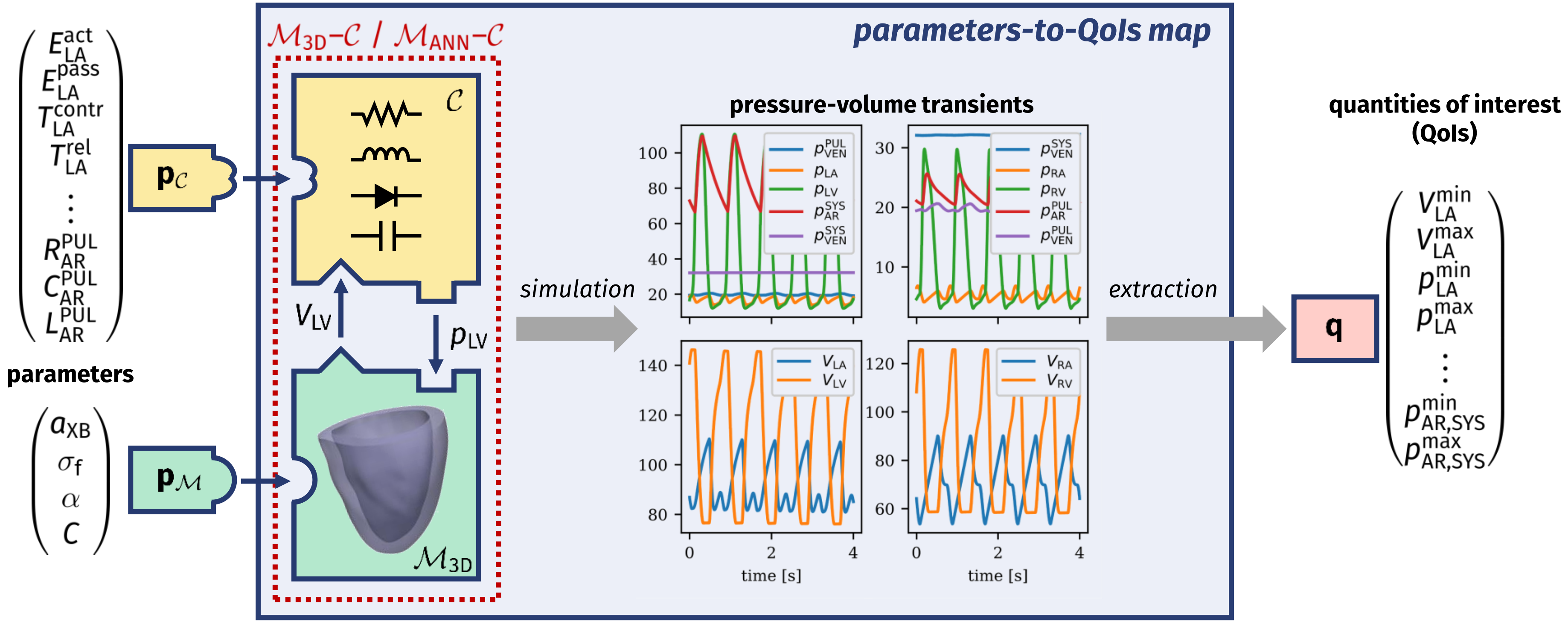






$$
\begin{cases}
\dfrac{\partial \mathbf{y}(t)}{\partial t} = \mathcal{L}(\mathbf{y}(t), p_{\mathrm{LV}}(t), t; \mathbf{p}_\mathcal{M}) & \text{for } t \in (0, T], \\
\dfrac{d\mathbf{c}(t)}{dt} = \mathbf{f}(\mathbf{c}(t), p_{\mathrm{LV}}(t), t; \mathbf{p}_\mathcal{C}) & \text{for } t \in (0, T], \\
V_{\mathrm{LV}}^{\mathrm{0D}}(\mathbf{c}(t)) = V_{\mathrm{LV}}^{\mathrm{3D}}(\mathbf{y}(t)) & \text{for } t \in (0, T], \\
\mathbf{y}(0) = \mathbf{y}_0, \\
\mathbf{c}(0) = \mathbf{c}_0,
\end{cases}
$$

Parameters-to-QoIs map: full-order model

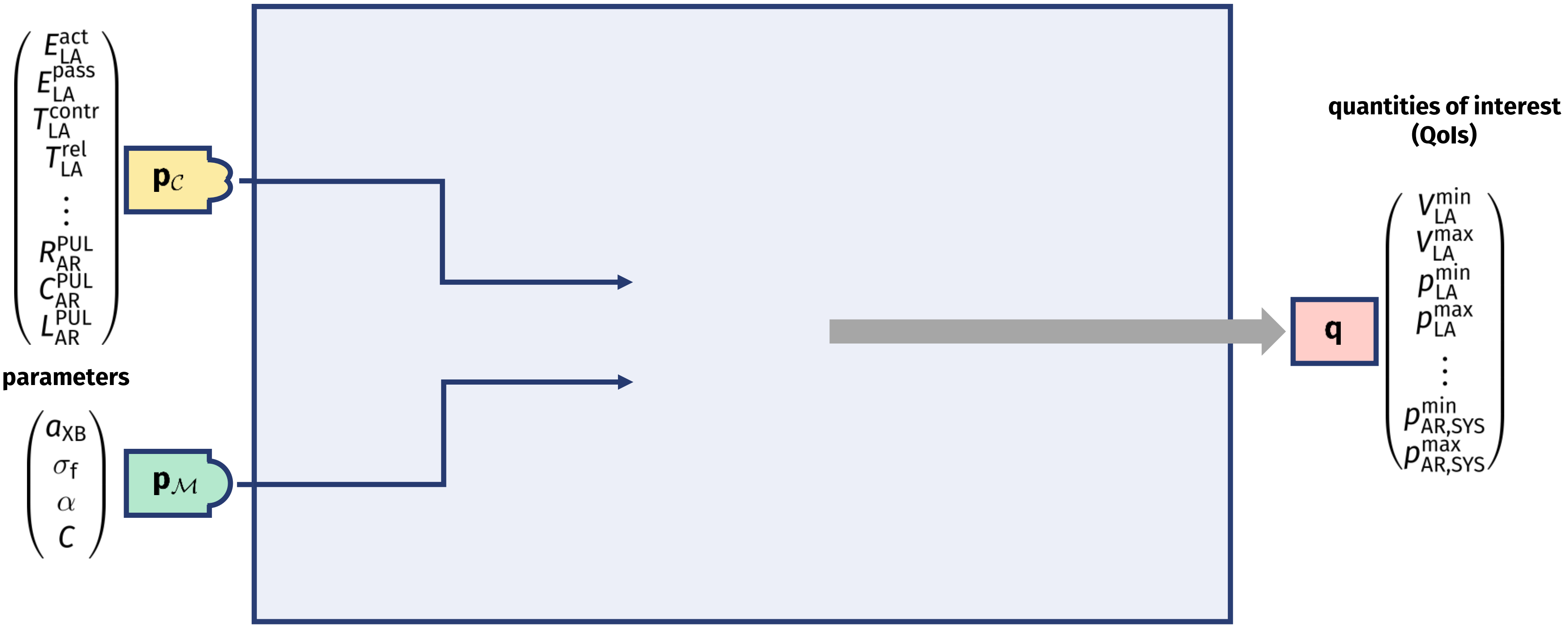


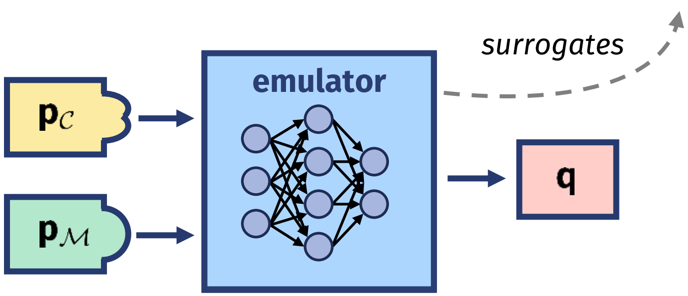
Computationally demanding
(our numerical tests: 4 hours on a 32-cores cluster computer for one heartbeat)

Unaffordable computational costs associated with:

- Sensitivity analysis
- Uncertainty quantification
- Parameter estimation under uncertainty

Machine Learning based emulators



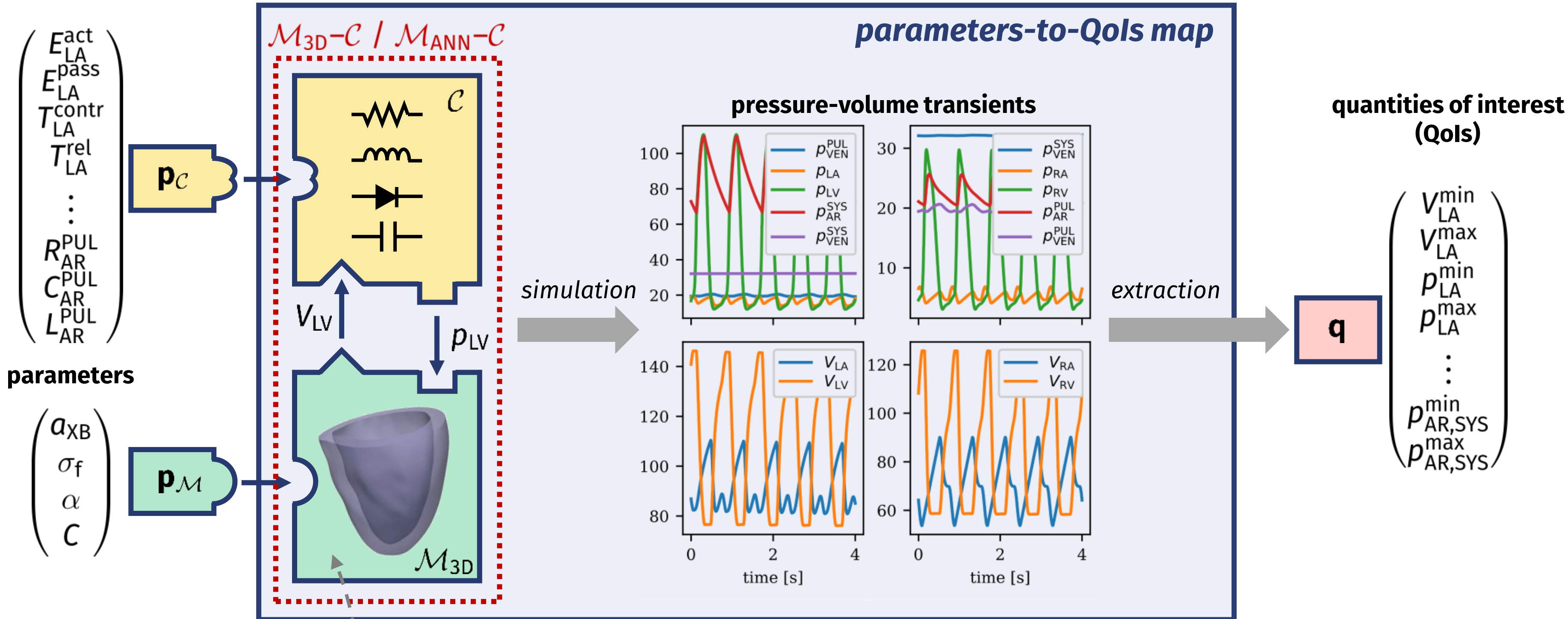


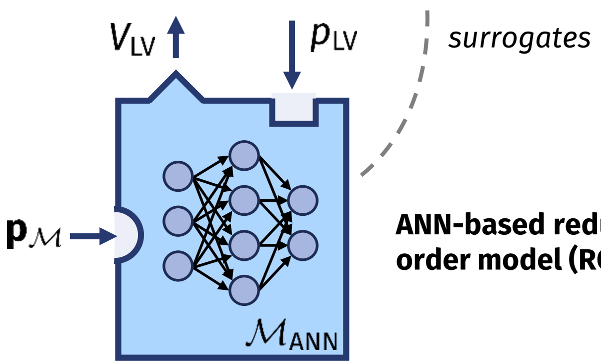
- ✓ Gaussian Process emulators (GPEs)
- ✓ Artificial Neural Networks (ANNs)
- ✓ Decision tree algorithms: K-Nearest Neighbor (KNN), eXtreme Gradient Boosting (XGBoost)

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

- P. Di Achille, A. Harouni, S. Khamzin, O. Solovyova, et al., *Frontiers in Physiology* 9 (2018)
- Y. Dabiri, A. Van der Velden, K. L. Sack, J. S. Choy, et al., *Frontiers in Physics* 7 (2019)
- S. Longobardi, A. Lewalle, S. Coveney, et al., *Phil. Transactions of the Royal Society* (2020)
- L. Cai, L. Ren, Y. Wang, W. Xie, et al., *Royal Society Open Science* 8 (1) (2021)

Model-Learning based emulator

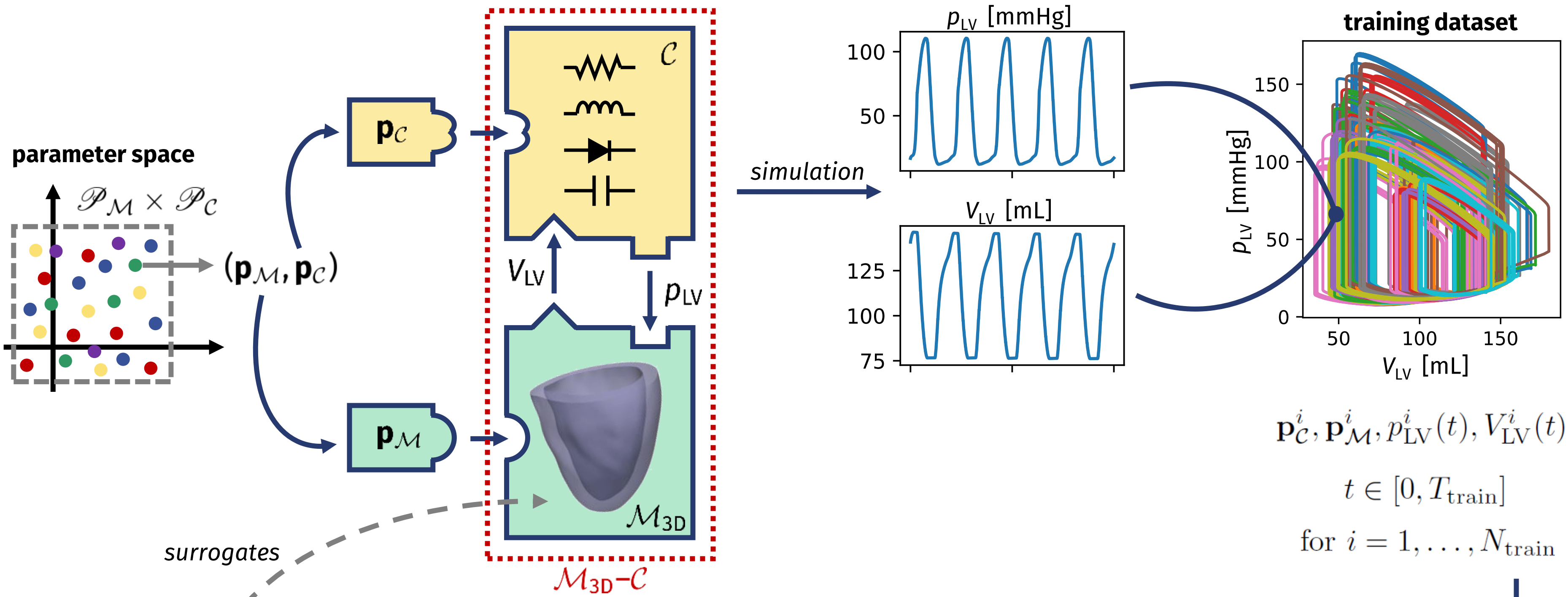


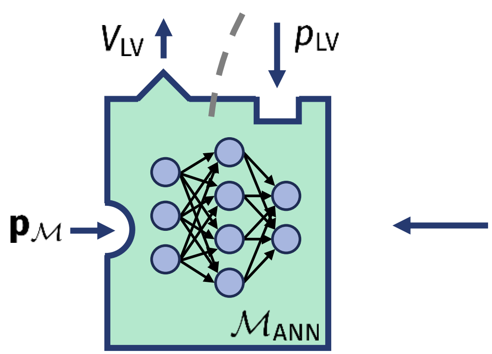


ANN-based reduced order model (ROM)

- ✓ We surrogate only the computationally demanding 3D components
- ✓ Lightweight 0D components are left in high-fidelity form
- ✓ Evaluation consists in numerically solving a (small) system of ordinary differential equations
- ✓ Enables real-time simulations

- ☺ 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





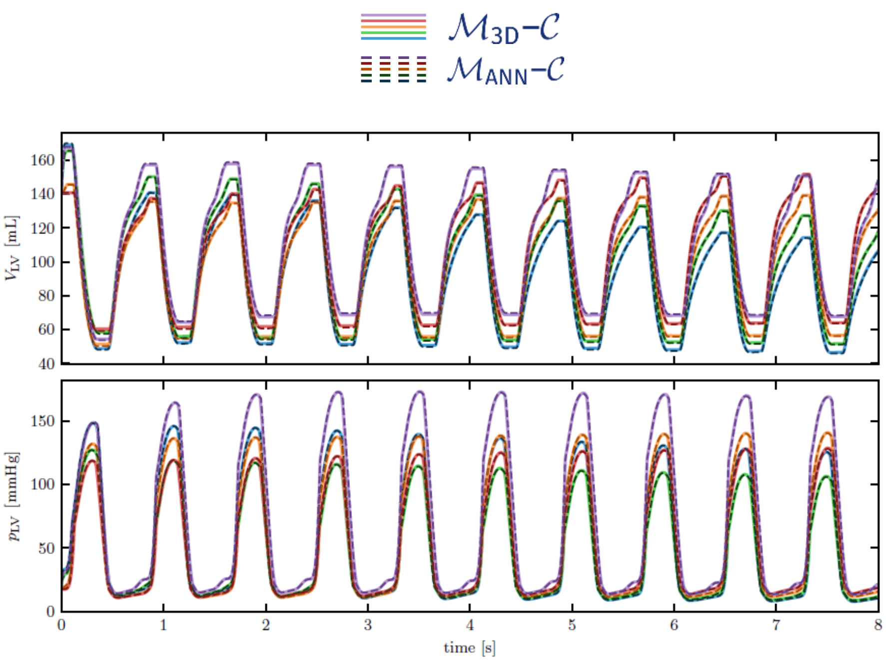
ANN-based ROM

Model Learning

$$
\begin{cases}
\widehat{\mathbf{w}} = \underset{\mathbf{w}\in\mathbb{R}^{N_w}}{\operatorname{argmin}} \left[\sum_{i=1}^{N_{\text{train}}} \int_0^{T_{\text{train}}} |V_{LV}^i(t) - V_{LV}^{ANN}(\mathbf{z}^i(t))|^2 \, dt + \beta |\mathbf{w}|^2 \right] \\
\text{such that, for each } i = 1, \ldots, N_{\text{train}}: \\
\quad \dfrac{d\mathbf{z}^i(t)}{dt} = \mathcal{NN}\left(\mathbf{z}^i(t), p_{LV}^i(t), \cos(\tfrac{2\pi t}{T_{HB}}), \sin(\tfrac{2\pi t}{T_{HB}}); \mathbf{p}_{\mathcal{M}}^i; \mathbf{w}\right) \quad \text{for } t \in (0, T_{\text{train}}], \\
\quad \mathbf{z}^i(0) = \mathbf{z}_0,
\end{cases}
$$

F. . Rregazzoni, L. Dede', A. Quarteroni "Machine learning for fast and reliable solution of time-dependent differential equations", *Journal of Computational Physics* (2019) 397, 108852

Results

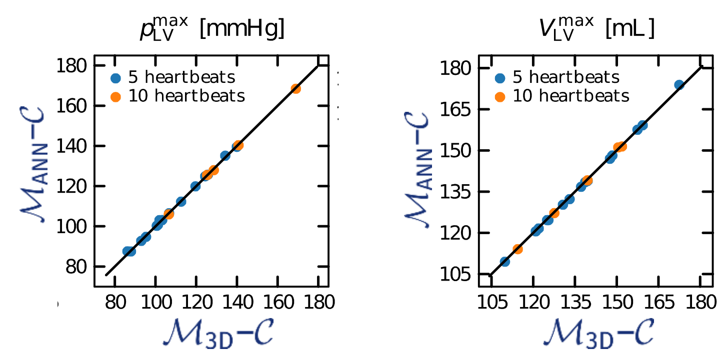


testing accuracy

	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 cores supercomputer	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

460'000x speedup

F. Regazzoni, M. Salvador, L. Dedé, A. Quarteroni, *Computer Methods in Applied Mechanics and Engineering*, 2022

Trained ROMs

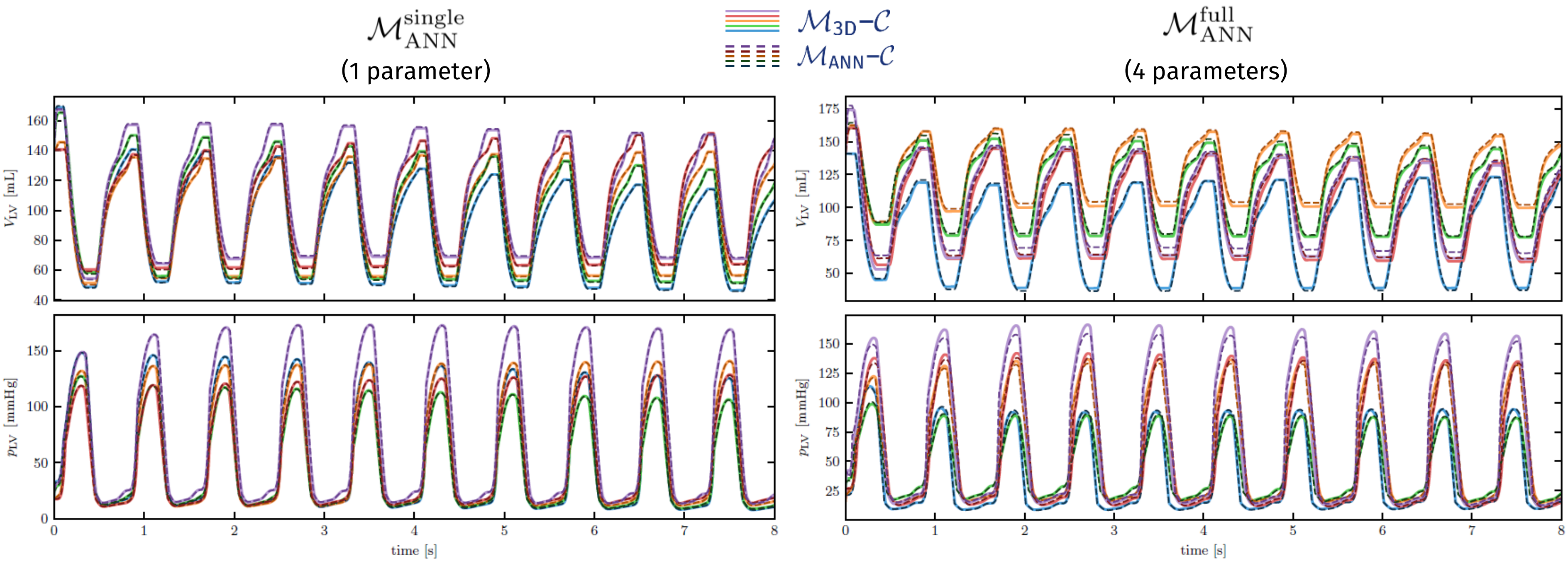
Parameter	Baseline	Unit	Description
a_{XB}	160.0	MPa	Cardiomyocytes contractility
σ_{f}	76.43	$\mathrm{mm\,s^{-1}}$	Electrical conductivity along fibers
α	60.0	degrees	Fibers angle rotation
C	0.88	kPa	Passive stiffness

Trained model	Parameters $\mathbf{p}_{\mathcal{M}}$	Training set size N_{train}	Hyperparameters N_z	N_{layers}	N_{neurons}	β
$\mathcal{M}_{\mathrm{ANN}}^{\mathrm{single}}$	$[a_{\mathrm{XB}}]$	30	2	1	8	0
$\mathcal{M}_{\mathrm{ANN}}^{\mathrm{full}}$	$[a_{\mathrm{XB}}, \sigma_{\mathrm{f}}, \alpha, C]$	40	1	1	12	0.01

Testing PV-loop

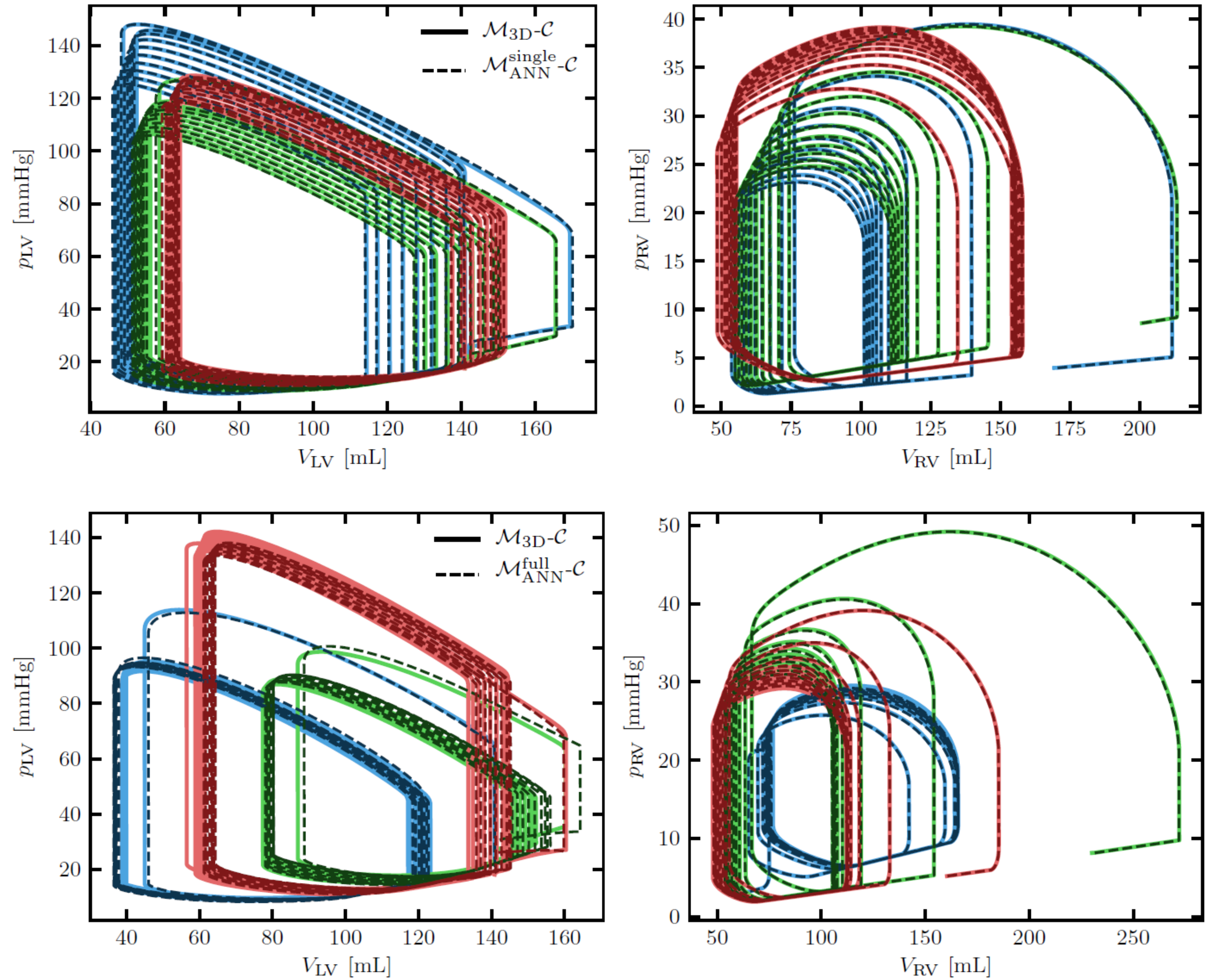
$\mathcal{M}_{\mathrm{ANN}}^{\mathrm{single}}$ (1 parameter)

$\mathcal{M}_{\mathrm{ANN}}^{\mathrm{full}}$ (4 parameters)



$\mathcal{M}_{\mathrm{ANN}}^{\mathrm{single}}$
(1 parameter)

$\mathcal{M}_{\mathrm{ANN}}^{\mathrm{full}}$
(4 parameters)



Trained ROM: QoIs

		$p_{LV}(t)$	$V_{LV}(t)$	p_{LV}^{min}	p_{LV}^{max}	V_{LV}^{min}	V_{LV}^{max}
		5 heartbeats					
$\mathcal{M}_{ANN}^{single}$-$\mathcal{C}$ vs $\mathcal{M}_{3D}$-$\mathcal{C}$	relative error	0.0336	0.0090	0.0097	0.0046	0.0139	0.0035
	R^2			99.691	99.864	99.896	99.948
$\mathcal{M}_{ANN}^{full}$-$\mathcal{C}$ vs $\mathcal{M}_{3D}$-$\mathcal{C}$	relative error	0.0620	0.0285	0.0517	0.0272	0.0471	0.0127
	R^2			94.370	95.302	95.942	97.061
		10 heartbeats					
$\mathcal{M}_{ANN}^{single}$-$\mathcal{C}$ vs $\mathcal{M}_{3D}$-$\mathcal{C}$	relative error	0.0293	0.0071	0.0113	0.0037	0.0096	0.0031
	R^2			99.924	99.980	99.851	99.944
$\mathcal{M}_{ANN}^{full}$-$\mathcal{C}$ vs $\mathcal{M}_{3D}$-$\mathcal{C}$	relative error	0.0631	0.0265	0.0442	0.0147	0.0382	0.0122
	R^2			92.227	99.957	99.229	99.063

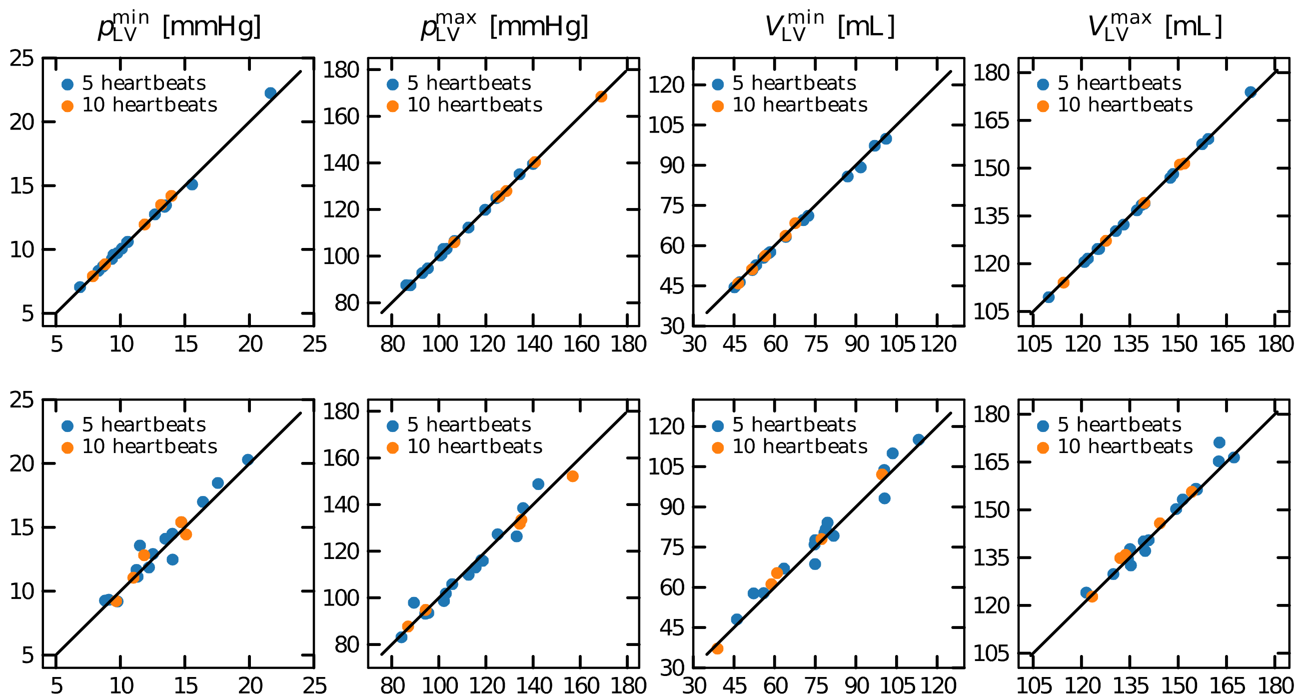
Test dataset #1
Same time horizon as training dataset

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

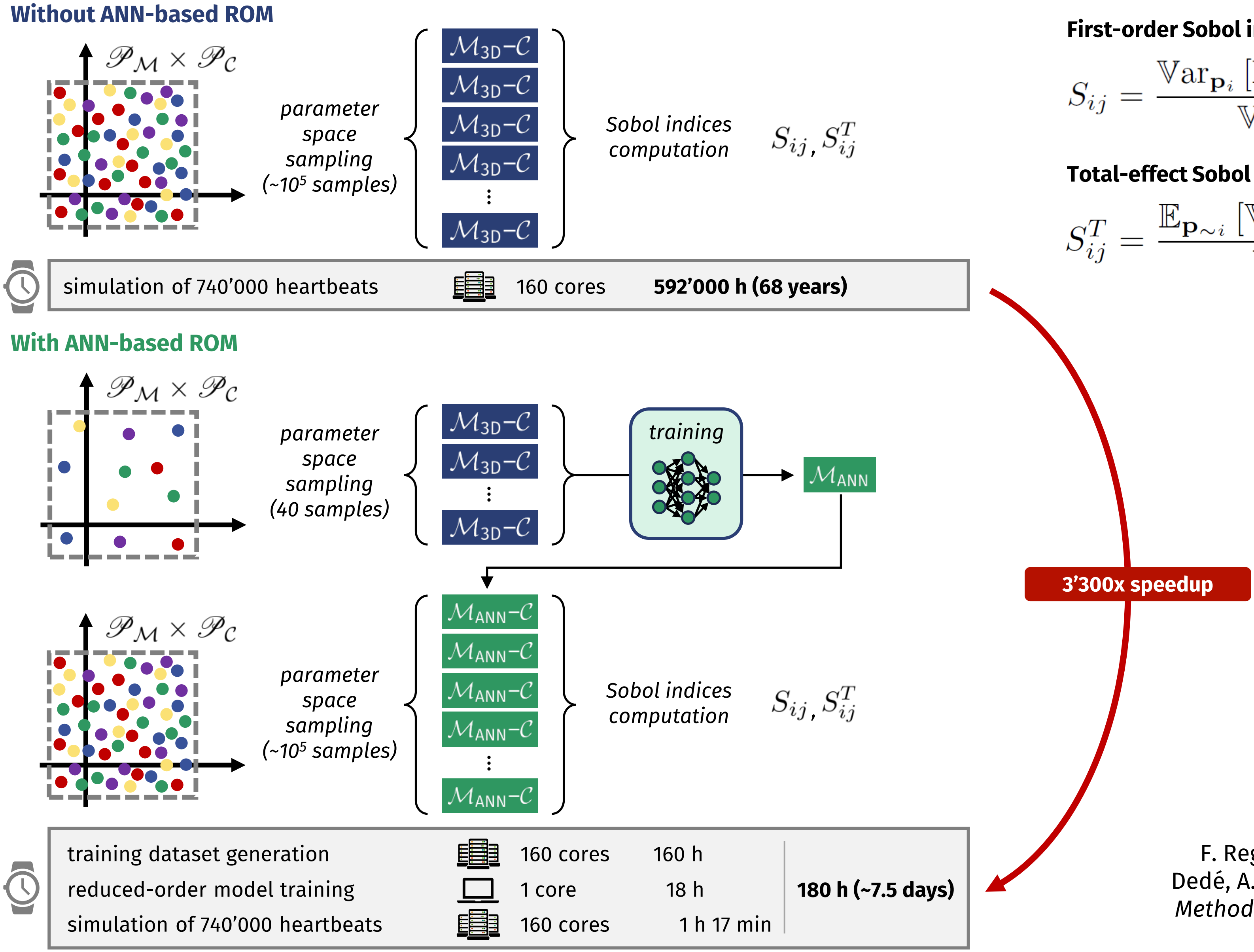
➡ time extrapolation!

$\mathcal{M}_{ANN}^{single}$-$\mathcal{C}$ vs $\mathcal{M}_{3D}$-$\mathcal{C}$
(1 parameter)

$\mathcal{M}_{ANN}^{full}$-$\mathcal{C}$ vs $\mathcal{M}_{3D}$-$\mathcal{C}$
(4 parameters)



Variance-based Sensitivity Analysis



First-order Sobol indices:

$$S_{ij} = \frac{\mathbb{V}\mathrm{ar}_{\mathbf{p}_i}\left[\mathbb{E}_{\mathbf{p}_{\sim i}}\left[\mathbf{q}_j | \mathbf{p}_i\right]\right]}{\mathbb{V}\mathrm{ar}\left[\mathbf{q}_j\right]}$$

Total-effect Sobol indices:

$$S_{ij}^T = \frac{\mathbb{E}_{\mathbf{p}_{\sim i}}\left[\mathbb{V}\mathrm{ar}_{\mathbf{p}_i}\left[\mathbf{q}_j | \mathbf{p}_{\sim i}\right]\right]}{\mathbb{V}\mathrm{ar}\left[\mathbf{q}_j\right]}$$

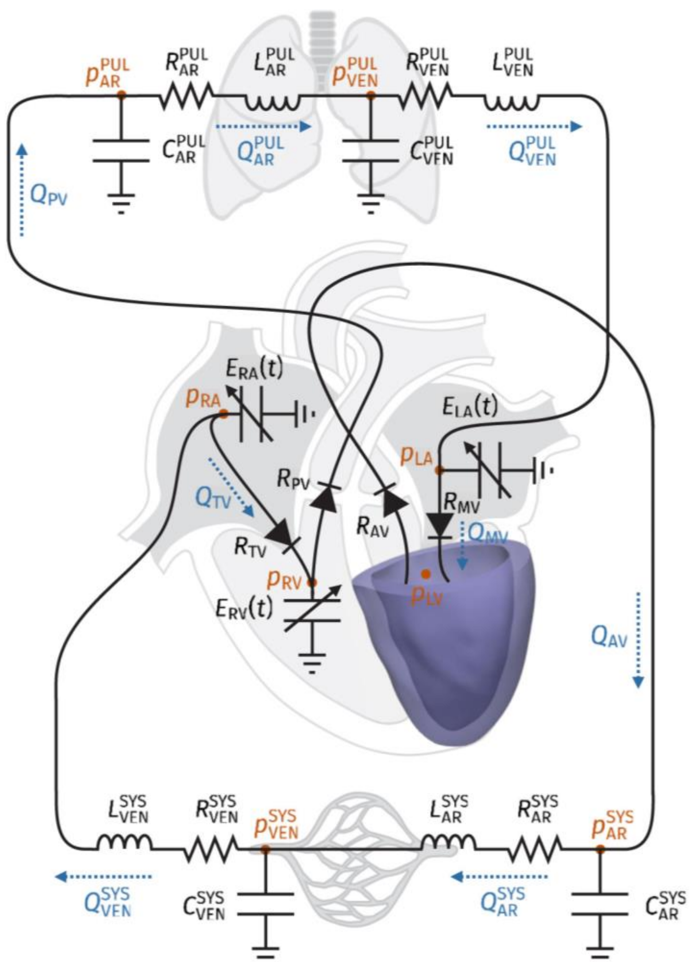
F. Regazzoni, M. Salvador, L. Dedé, A. Quarteroni, *Computer Methods in Applied Mechanics and Engineering*, 2022

Variance-based Sensitivity Analysis (results)

	V_{LA}^{min}	V_{LA}^{max}	p_{LA}^{min}	p_{LA}^{max}	V_{LV}^{min}	V_{LV}^{max}	p_{LV}^{min}	p_{LV}^{max}	SV_{LV}	V_{RA}^{min}	V_{RA}^{max}	p_{RA}^{min}	p_{RA}^{max}	V_{RV}^{min}	V_{RV}^{max}	p_{RV}^{min}	p_{RV}^{max}	SV_{RV}	$p_{AR,SYS}^{min}$	$p_{AR,SYS}^{max}$
E_{LA}^{act}	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
E_{LA}^{pass}	0.23	0.26	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00
T_{LA}^{contr}	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
T_{LA}^{rel}	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
t_L^{av}	0.03	0.00	0.00	0.13	0.01	0.04	0.00	0.02	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.02
$V_{0,LA}$	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
C	0.03	0.02	0.04	0.03	0.01	0.05	0.05	0.05	0.06	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.01	0.01	0.02	0.05
α	0.03	0.02	0.07	0.04	0.14	0.03	0.08	0.10	0.13	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.02	0.01	0.05	0.10
σ_f	0.00	0.00	0.00	0.00	0.02	0.00	0.01	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01
a_{XB}	0.10	0.07	0.15	0.09	0.51	0.16	0.18	0.22	0.26	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.04	0.04	0.10	0.22
E_{RA}^{act}	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.03	0.05	0.01	0.00	0.00	0.01	0.00	0.00	0.00	0.00
E_{RA}^{pass}	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.17	0.21	0.05	0.01	0.00	0.00	0.04	0.00	0.00	0.00	0.00
T_{RA}^{contr}	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
T_{RA}^{rel}	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
t_R^{av}	0.00	0.00	0.01	0.00	0.00	0.01	0.01	0.00	0.00	0.42	0.18	0.35	0.54	0.00	0.01	0.03	0.01	0.02	0.00	0.00
$V_{0,RA}$	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
E_{RV}^{act}	0.01	0.01	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.02	0.01	0.03	0.02	0.37	0.15	0.13	0.01	0.00	0.00	0.00
E_{RV}^{pass}	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.01	0.15	0.12	0.22	0.16	0.02	0.03	0.42	0.01	0.02	0.00	0.00
T_{RV}^{contr}	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.00	0.00
T_{RV}^{rel}	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.00	0.06	0.00	0.00	0.00	0.00
$V_{0,RV}$	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.06	0.03	0.00	0.00	0.00	0.00	0.00
R_{AR}^{SYS}	0.02	0.01	0.03	0.01	0.10	0.02	0.04	0.23	0.08	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.53	0.24
C_{AR}^{SYS}	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.05	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.13	0.05
R_{VEN}^{SYS}	0.07	0.09	0.10	0.09	0.01	0.09	0.09	0.02	0.09	0.10	0.30	0.18	0.15	0.10	0.51	0.16	0.24	0.82	0.00	0.02
C_{VEN}^{SYS}	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
L_{AR}^{SYS}	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
L_{VEN}^{SYS}	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
R_{AR}^{PUL}	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
C_{AR}^{PUL}	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00
R_{VEN}^{PUL}	0.00	0.00	0.01	0.00	0.00	0.01	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
C_{VEN}^{PUL}	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
L_{AR}^{PUL}	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
L_{VEN}^{PUL}	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
R_{min}	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.01	0.01	0.01	0.01	0.00	0.00	0.01	0.03	0.00	0.00	0.00
R_{max}	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
V_{heart}^{tot}	0.39	0.45	0.53	0.52	0.15	0.53	0.49	0.27	0.31	0.04	0.07	0.06	0.05	0.32	0.24	0.11	0.55	0.03	0.14	0.26

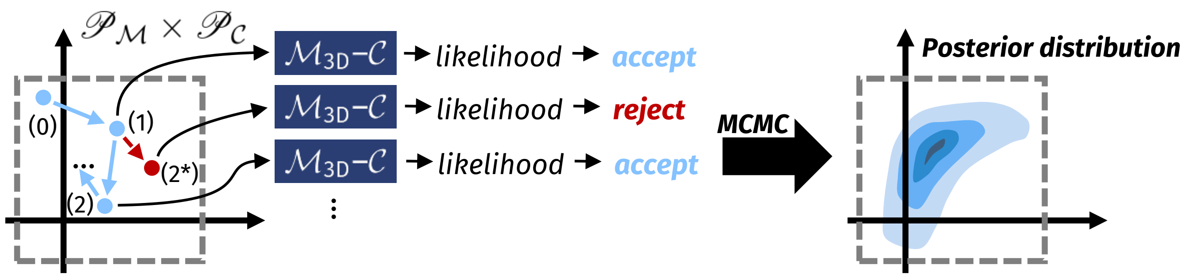
First-order Sobol indices:

$$S_{ij} = \frac{\mathbb{V}\mathrm{ar}_{\mathbf{p}_i}\left[\mathbb{E}_{\mathbf{p}_{\sim i}}\left[\mathbf{q}_j|\mathbf{p}_i\right]\right]}{\mathbb{V}\mathrm{ar}\left[\mathbf{q}_j\right]}$$



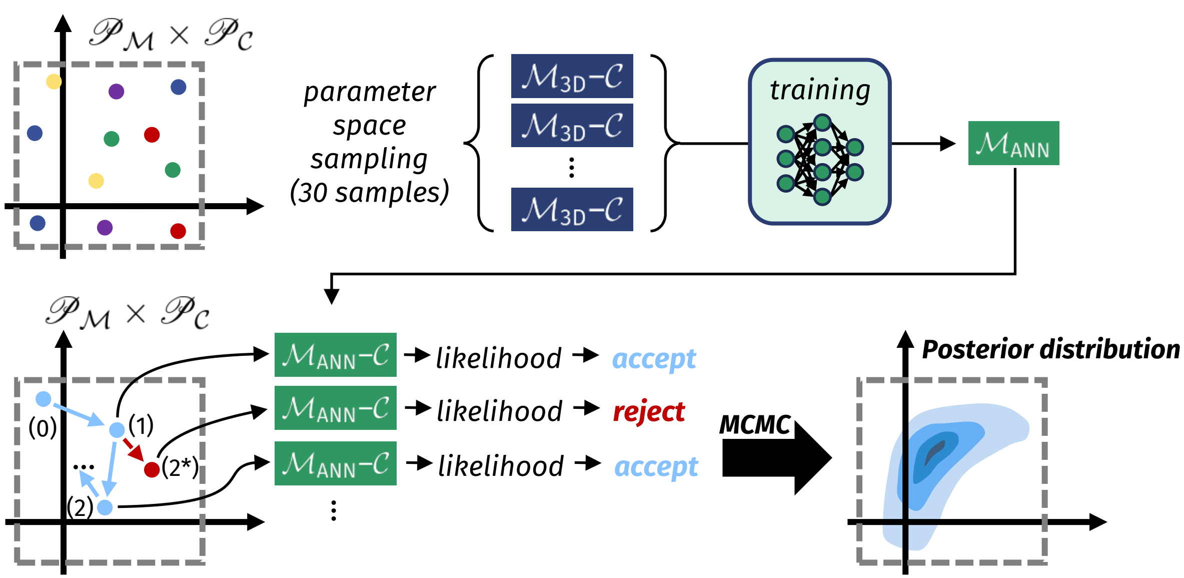
Bayesian Parameter estimation

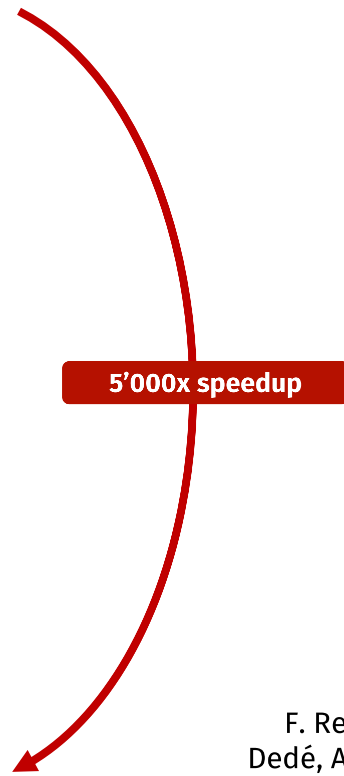
Without ANN-based ROM



simulation of 960'000 heartbeats	160 cores		**768'000 h (87 years)**

With ANN-based ROM





training dataset generation	160 cores	120 h	**162 h (~6.25 days)**
reduced-order model training	1 core	18 h	
simulation of 960'000 heartbeats	20 cores	13 h 20 min	

F. Regazzoni, M. Salvador, L. Dedé, A. Quarteroni, *Computer Methods in Applied Mechanics and Engineering*, 2022

Bayesian Parameter estimation (results)

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

$\mathbf{q}_{\text{obs}} = \mathcal{F}(\mathbf{p}) + \boldsymbol{\epsilon}$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\cdot|\mathbf{0}, \boldsymbol{\Sigma})$, $\boldsymbol{\Sigma}$ = noise covariance (measurement error + model error)

$\pi_{\text{prior}}(\mathbf{p})$ Prior distribution (a priori knowledge on the parameters)

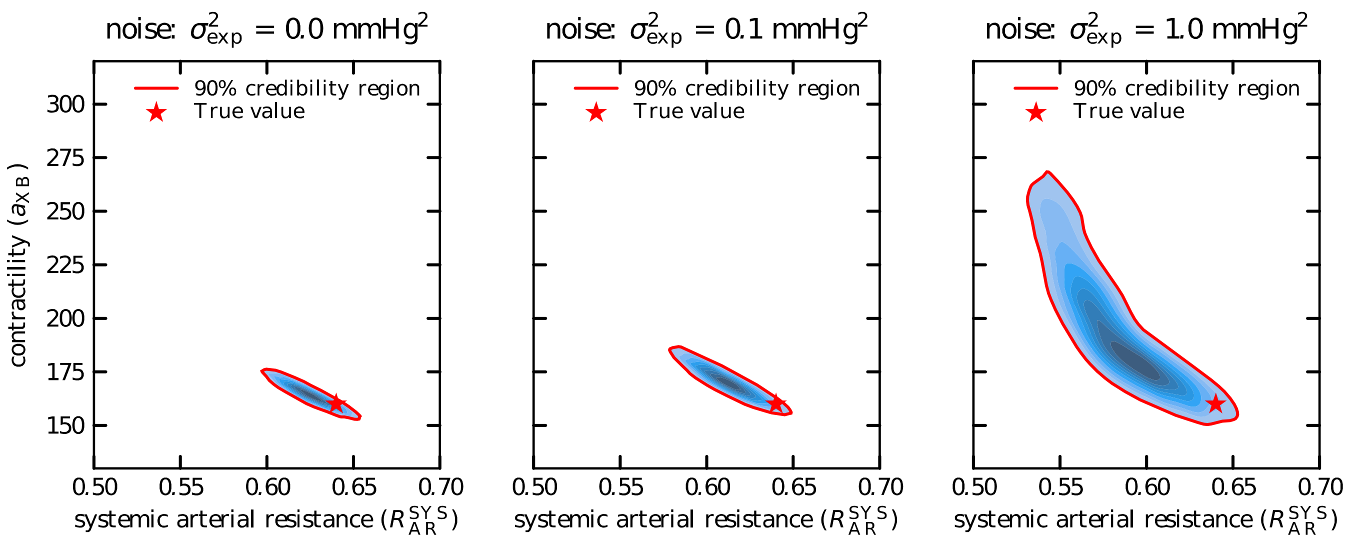
Posterior distribution:

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

Test case

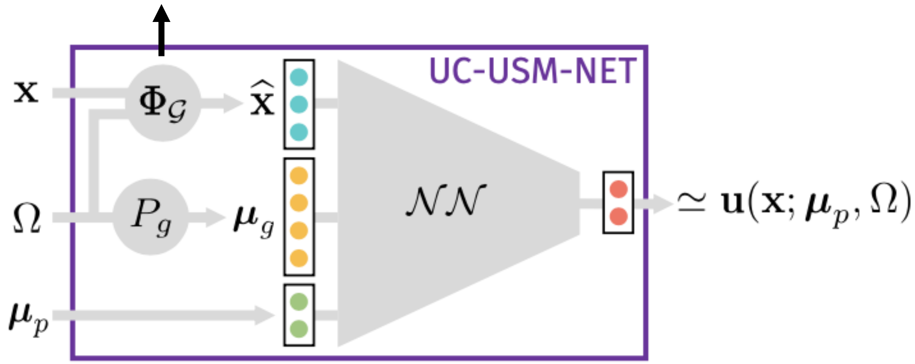
Observed QoIs: maximim and minimum arterial pressures

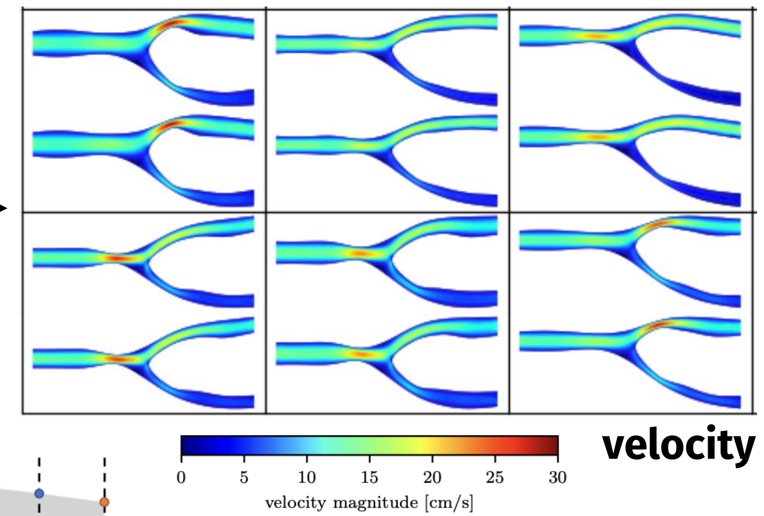
Unknown parameters: myocardial contractility, systemic arterial resistance

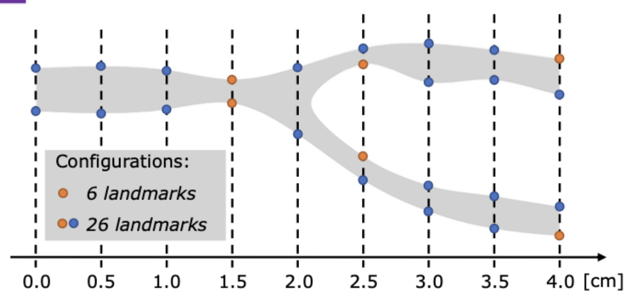


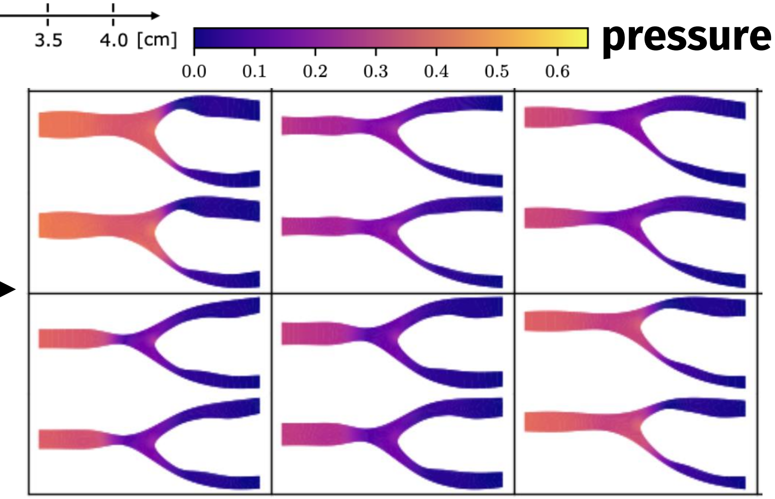
Universal Solution Manifold Networks (USM-Nets)

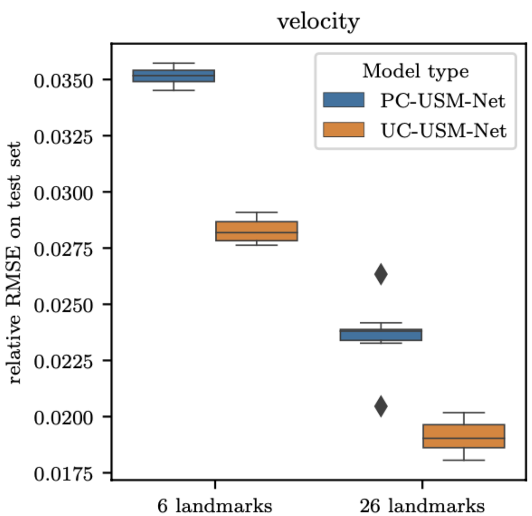
Universal coordinates system











F. Regazzoni, S. Pagani, A. Quarteroni. "*ASME Journal of Biomechanical Engineering*, 2022