



MOX-Report No. 68/2021

**A machine learning method for real-time numerical
simulations of cardiac electromechanics**

Regazzoni, F.; Salvador, M.; Dede', L.; Quarteroni, A.

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

mox-dmat@polimi.it

<http://mox.polimi.it>

A machine learning method for real-time numerical simulations of cardiac electromechanics

Francesco Regazzoni^{1,*}, Matteo Salvador¹ Luca Dedè¹, Alfio Quarteroni^{1,2}

¹ MOX-Dipartimento di Matematica, Politecnico di Milano, Milan, Italy

² École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland (*Professor Emeritus*)

* *Corresponding author* (francesco.regazzoni@polimi.it)

Abstract

We propose a machine learning-based method to build a system of differential equations that approximates the dynamics of 3D electromechanical models for the human heart, accounting for the dependence on a set of parameters. Specifically, our method permits to create a reduced-order model (ROM), written as a system of Ordinary Differential Equations (ODEs) wherein the forcing term, given by the right-hand side, consists of an Artificial Neural Network (ANN), that possibly depends on a set of parameters associated with the electromechanical model to be surrogated. This method is non-intrusive, as it only requires a collection of pressure and volume transients obtained from the full-order model (FOM) of cardiac electromechanics. Once trained, the ANN-based ROM can be coupled with hemodynamic models for the blood circulation external to the heart, in the same manner as the original electromechanical model, but at a dramatically lower computational cost. Indeed, our method allows for real-time numerical simulations of the cardiac function. Our results show that the ANN-based ROM is accurate with respect to the FOM (relative error between 10^{-3} and 10^{-2} for biomarkers of clinical interest), while requiring very small training datasets (30-40 samples). We demonstrate the effectiveness of the proposed method on two relevant contexts in cardiac modeling. First, we employ the ANN-based ROM to perform a global sensitivity analysis on both the electromechanical and hemodynamic models. Second, we perform a Bayesian estimation of two parameters starting from noisy measurements of two scalar outputs. In both these cases, replacing the FOM of cardiac electromechanics with the ANN-based ROM makes it possible to perform in a few hours of computational time all the numerical simulations that would be otherwise unaffordable, because of their overwhelming computational cost, if carried out with the FOM. As a matter of fact, our ANN-based ROM is able to speedup the numerical simulations by more than three orders of magnitude.

Keywords: Cardiac Electromechanics, Machine Learning, Reduced Order Modeling, Global Sensitivity Analysis, Bayesian Parameter Estimation

1 Introduction

Numerical simulations of cardiac electromechanics are gaining momentum in the context of cardiovascular research and computational medicine [12, 15, 44, 55, 59, 62, 64]. Cardiac in silico models are based on an anatomically detailed and biophysically accurate representation of the human heart, consisting of multiscale mathematical models of several physical processes, from the cellular to the organ scale, described by systems of differential equations. However, the clinical exploitation of

cardiac numerical simulations is seriously hampered by their overwhelming computational cost. As a matter of fact, the simulation of a single heartbeat for an anatomically accurate patient-specific model may require several hours of computational time even on a supercomputer platform.

A promising approach to address this issue is to replace the computationally expensive cardiac electromechanical model, say the full-order model (FOM), with a reduced version of it, called reduced-order model (ROM), to be called any time new parameters come in. In the so-called *offline phase*, the ROM is built from a database of numerical simulations that are previously obtained by solving the FOM itself. Then, in the *online phase*, the ROM is used as a surrogate of the FOM, at a dramatically reduced computational cost, for any new instance of the parameters. Clearly, this procedure pays off if the computational gain of the online phase outweighs the cost of the offline phase, or whenever the online phase requires real-time execution, which is often the case in the clinical practice, where timeliness is a key factor.

Recently, this framework has been applied in the context of cardiac modeling, primarily by using machine learning algorithms, including Gaussian Process emulators (GPEs), Artificial Neural Networks (ANNs), decision tree algorithms such as eXtreme Gradient Boosting (XGBoost) and K-Nearest Neighbor (KNN) [7, 10, 11, 13, 31]. These emulators are trained to fit the map that links the model parameters with a set of scalar outputs of interest, known as quantities of interest (QoIs), which represent clinically meaningful biomarkers. This map is learned from a collection of numerical simulations, obtained by sampling the parameter space, which can then be used as FOM surrogate.

Replacing a FOM with a ROM is particularly advantageous in the so-called many-query settings, i.e. when the forward model has to be solved many times for different parameter values. This is the case, for example, of parameter calibration [6, 22] and sensitivity analysis [28, 43, 56, 67], fundamental tools for a reliable use of computational models in the clinical practice. In fact, only a few of the many parameters feeding cardiac electromechanical models can be directly measured, while most of them must be estimated through indirect (possibly non-invasive) measurements, by solving suitable (typically ill-posed) inverse problems [2]. It is therefore important to quantify how much each parameter affects the outputs of the model and, vice versa, how uncertainty on measured data reverberates on parameter uncertainty. These requirements are at the basis of the *verification and validation under uncertainty* (VVUQ) paradigm [25, 36] and are often necessary to comply with standards issued by health agencies and policy makers, such as the ASME V&V40 standard [1] recognized by the FDA [14].

In this paper, we propose a machine learning method to build ROMs of cardiac electromechanical models. Our approach relies on the ANN-based method that we proposed in [47], which can learn a time-dependent differential equation from a collection of input-output pairs. With respect to existing approaches, we only surrogate the time-dependent pressure-volume relationship of a cardiac chamber, while we do not reduce the model describing external circulation. The latter is indeed either a low dimensional 0D windkessel or closed-loop circulation model comprised of few state variables (up to two dozens), which does not require further reduction. In other terms, we derive an ANN-based ROM for the computationally demanding 3D components, while we leave in FOM version the lightweight ones. Unlike emulators, for which the online phase consists in evaluations of the map linking model parameters to QoIs, with our approach the online phase consists instead in numerical simulations, in which the ANN-based ROM of the electromechanical model is coupled with the circulation model, at a very low computational cost. As a matter of fact, these numerical simulations can be performed in real-time on a standard laptop.

We consider two different test cases to prove the efficacy of our mathematical approach. We perform variance-based global sensitivity analysis on both electromechanical and hemodynamic parameters and Bayesian estimation by means of the Markov Chain Monte Carlo (MCMC) method to infer two parameters from noisy measurements of two scalar outputs.

This paper is organized as follows. In Sec. 2 we present the proposed method and the models used to produce the numerical results, which are presented and commented in Sec. 3. Then, in Sec. 4, we critically discuss the obtained results and the pros and cons of the proposed method, with respect to other methods available in the literature. Finally, we draw our conclusions and final remarks in Sec. 5.

This manuscript is accompanied by <https://github.com/FrancescoRegazzoni/cardioEM-learning>, a public repository containing the codes and the datasets necessary to reproduce the presented results.

2 Methods

We introduce our method in an abstract formulation for cardiac electromechanics. As a matter of fact, our approach, thanks to its non-intrusive nature, can be applied to different electromechanical models that are available in literature.

2.1 The full-order model (FOM)

Let us consider a generic model of cardiac electromechanics, that is a set of differential equations describing physical processes involved in the heart function. We introduce the state vector $\mathbf{y}(t)$, collecting the state variables associated with this multiphysics system. These may include the transmembrane potential, gating variables, ionic concentrations, protein states, tissue displacement, or simply phenomenological variables. In this paper, we focus on the single-chamber case of the human heart (e.g. the left ventricle, that we call from now on LV), as the generalization to multiple chambers is straightforward. By introducing a nonlinear differential operator \mathcal{L} that encodes the differential equations and boundary conditions associated with the electromechanical model, the latter reads

$$\begin{cases} \frac{\partial \mathbf{y}(t)}{\partial t} = \mathcal{L}(\mathbf{y}(t), p_{LV}(t), t; \mathbf{p}_{\mathcal{M}}) & \text{for } t \in (0, T], \\ \mathbf{y}(0) = \mathbf{y}_0, \end{cases} \quad (1)$$

where $p_{LV}(t)$ denotes the LV endocardial pressure (here seen as an input), $\mathbf{p}_{\mathcal{M}}$ are the model parameters (possibly including, e.g., electrical conductivities, cell membrane conductances, protein binding affinities, contractility, passive tissue properties) and \mathbf{y}_0 is the initial state. We denote by $\mathcal{P}_{\mathcal{M}} \subseteq \mathbb{R}^{N_{\mathcal{M}}}$ the space of parameters such that $\mathbf{p}_{\mathcal{M}} \in \mathcal{P}_{\mathcal{M}}$, being $N_{\mathcal{M}}$ the number of parameters. We notice that the right-hand side of (1) depends on t , as heartbeats are paced by externally applied stimuli that we assume to have a period of duration T_{HB} , which is fixed a priori.

The 3D cardiac electromechanical model (1), henceforth denoted by \mathcal{M}_{3D} , must be coupled with a closure relationship assigning the pressure $p_{LV}(t)$. One possible option is to couple the \mathcal{M}_{3D} model with a 0D model for the external circulation (see e.g. [3, 21, 50]), thus obtaining a system in the form of

$$\begin{cases} \frac{\partial \mathbf{y}(t)}{\partial t} = \mathcal{L}(\mathbf{y}(t), p_{LV}(t), t; \mathbf{p}_{\mathcal{M}}) & \text{for } t \in (0, T], \\ \frac{dc(t)}{dt} = \mathbf{f}(\mathbf{c}(t), p_{LV}(t), t; \mathbf{p}_{\mathcal{C}}) & \text{for } t \in (0, T], \\ V_{LV}^{0D}(\mathbf{c}(t)) = V_{LV}^{3D}(\mathbf{y}(t)) & \text{for } t \in (0, T], \\ \mathbf{y}(0) = \mathbf{y}_0, \\ \mathbf{c}(0) = \mathbf{c}_0, \end{cases} \quad (2)$$

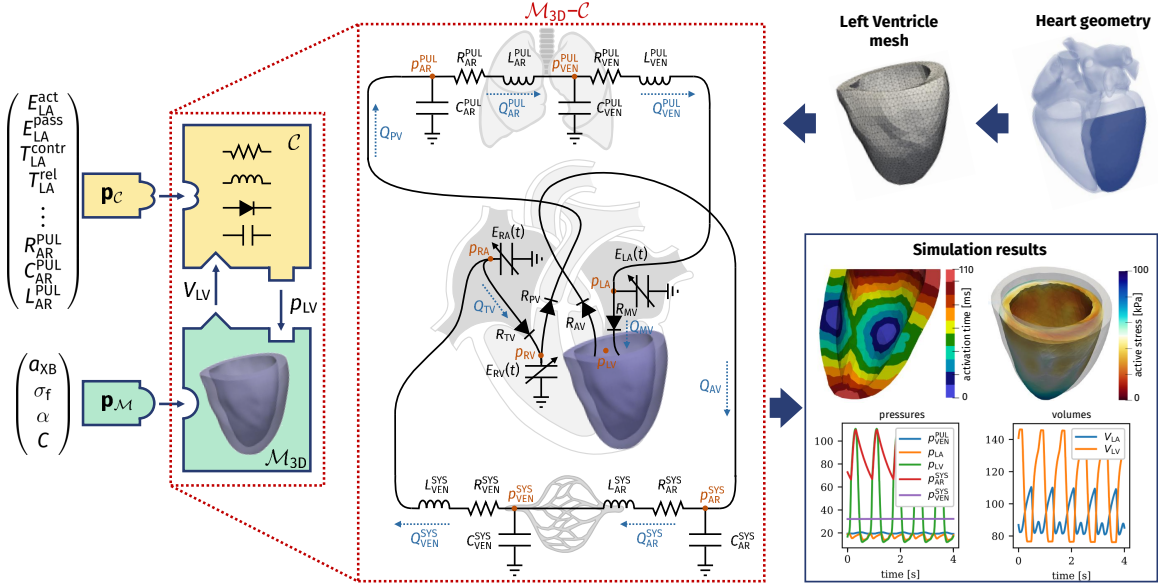


Figure 1: The $\mathcal{M}_{3D}\text{-}\mathcal{C}$ model. The parameters \mathbf{p}_C and \mathbf{p}_M (left) are associated with the \mathcal{C} and \mathcal{M}_{3D} respectively. The two mathematical models are coupled via the variables p_{LV} and V_{LV} . Their union constitutes the model $\mathcal{M}_{3D}\text{-}\mathcal{C}$. The model $\mathcal{M}_{3D}\text{-}\mathcal{C}$ that we consider to produce the numerical results of this paper is shown in the center of the figure. For more details on the $\mathcal{M}_{3D}\text{-}\mathcal{C}$ model and for the definition of the parameters \mathbf{p}_C and \mathbf{p}_M , see Sec. 2.6. The 3D Finite Element model requires a computational mesh, obtained from the patient's heart geometry (top right). The output of the numerical simulations (bottom right) consists in spatially distributed fields (such as activation time, active tension and tissue displacement) and in 0D transients (such as pressures and volumes).

where $\mathbf{c}(t)$ are the state variables of the circulation model (pressures, volumes and fluxes in the circulatory network) and $\mathbf{p}_C \in \mathcal{P}_C \subseteq \mathbb{R}^{N_C}$ is a vector of N_C parameters (e.g. vascular resistances and conductances). The mechanical and hemodynamic models are coupled through the geometric consistency relationship $V_{LV}^{3D}(\mathbf{y}(t)) = V_{LV}^{0D}(\mathbf{c}(t))$, where the left-hand and right-hand sides represent the LV volume predicted by the \mathcal{M}_{3D} and by the \mathcal{C} models, respectively. The LV pressure p_{LV} is determined as a Lagrange multiplier that enforces the consistency relationship. An alternative approach is to adopt different closure relationships in the different phases of the heartbeat [19, 30] with suitable preload and afterload models, such as windkessel models [65]. These relationships link the changes in LV pressure p_{LV} with its volume, obtained as $V_{LV} = V_{LV}^{3D}(\mathbf{y})$. In both cases, should the \mathcal{M}_{3D} be coupled to a closed-loop circulation model or to an afterload-preload relationship, we denote by $\mathcal{M}_{3D}\text{-}\mathcal{C}$ the resulting coupled model. In Fig. 1 we show an $\mathcal{M}_{3D}\text{-}\mathcal{C}$ model in which \mathcal{C} consists of a the lumped-parameter closed-loop model of [50], which is employed to produce the numerical results of Sec. 3.

We remark that the closure relationships never directly involve the state \mathbf{y} of the model \mathcal{M}_{3D} , but only the LV volume $V_{LV} = V_{LV}^{3D}(\mathbf{y})$. This is a key observation since it suggests that a ROM that is able to surrogate the relationship between p_{LV} and V_{LV} , albeit agnostic of the state \mathbf{y} , can replace the \mathcal{M}_{3D} model in its coupling to the \mathcal{C} model. In what follows, we present our strategy to build an ANN-based ROM of the \mathcal{M}_{3D} model, denoted by \mathcal{M}_{ANN} , which can be coupled with the \mathcal{C} model resulting in the $\mathcal{M}_{ANN}\text{-}\mathcal{C}$ coupled model, that will surrogate the $\mathcal{M}_{3D}\text{-}\mathcal{C}$ model.

2.2 The reduced-order model (ROM)

To setup a ROM surrogating the \mathcal{M}_{3D} model of Eq. (1), we employ the machine learning method that we proposed in [47]. This method is designed to learn a differential equation from time-dependent input-output pairs, by training an Ordinary Differential Equation (ODE) model, whose right-hand side is represented by an ANN. In this work, we define our ANN-based ROM \mathcal{M}_{ANN} as

$$\begin{cases} \frac{d\mathbf{z}(t)}{dt} = \mathcal{NN}\left(\mathbf{z}(t), p_{LV}(t), \cos(\frac{2\pi t}{T_{HB}}), \sin(\frac{2\pi t}{T_{HB}}), \mathbf{p}_{\mathcal{M}}; \widehat{\mathbf{w}}\right) & \text{for } t \in (0, T], \\ \mathbf{z}(0) = \mathbf{z}_0, \end{cases} \quad (3)$$

where $\mathbf{z}(t) \in \mathbb{R}^{N_z}$ is the reduced state and $\mathcal{NN}: \mathbb{R}^{N_z + N_{\mathcal{M}} + 3} \rightarrow \mathbb{R}^{N_z}$ is a fully connected ANN. The ANN input consists indeed of N_z state variables, $N_{\mathcal{M}}$ scalar parameters, the pressure p_{LV} , and the two periodic inputs $\cos(2\pi t/T_{HB})$ and $\sin(2\pi t/T_{HB})$ (whose role will be clarified later), for a total of $N_z + N_{\mathcal{M}} + 3$ input neurons. The vector $\widehat{\mathbf{w}} \in \mathbb{R}^{N_w}$ encodes the weights and biases of the ANN, that need to be suitably trained. We remark that, among the arguments of the ANN, we have not introduced the time variable t , but rather $\cos(2\pi t/T_{HB})$ and $\sin(2\pi t/T_{HB})$, that are the coordinates of a point cyclically moving along a circumference with period T_{HB} . In this way, the ROM encodes by construction the periodicity associated with the heartbeat pacing. This expedient allows for the use of the ROM also for time spans longer than those shown during the training phase. Moreover, by introducing the parametric dependence (i.e. on $\mathbf{p}_{\mathcal{M}}$) within the ANN, the latter is not specific to a particular parameter setting.

In this work, according to [47], we adopt an output-inside-the-state approach, that is we train the model so that the LV volume coincides with the first state variable. More precisely, the LV volume predicted by the \mathcal{M}_{ANN} model is by definition $V_{LV}^{ANN}(\mathbf{z}(t)) := \mathbf{z}(t) \cdot \mathbf{e}_1$, where \mathbf{e}_1 is the first element of the canonical basis of \mathbb{R}^{N_z} . Therefore, the first ROM state variable has a clear physical interpretation (i.e. it coincides with the LV volume), while the other ROM states are latent variables with no immediate physical interpretation, providing however a compact representation of the full-order state $\mathbf{y}(t)$. Coherently with this choice, we define the initial state as $\mathbf{z}_0 = (V_{LV}^{3D}(\mathbf{y}_0), \mathbf{0})^T$. The remaining initial states are set, without loss of generality, to zero (this choice does not reduce the space of candidate models, as proved in [47]).

Here we need to determine the optimal value of the weights $\widehat{\mathbf{w}}$, such that the \mathcal{M}_{ANN} model reproduces the outputs of the \mathcal{M}_{3D} model as accurately as possible. With this goal, we generate a training set, by sampling the parameter space $\mathcal{P}_{\mathcal{M}} \times \mathcal{P}_{\mathcal{C}}$ with N_{train} sample points. For each sample, we perform a simulation with the \mathcal{M}_{3D-C} model until time T_{train} , and we record the LV pressure and volume transients. The training set is thus given by

$$\mathbf{p}_{\mathcal{C}}^i, \mathbf{p}_{\mathcal{M}}^i, p_{LV}^i(t), V_{LV}^i(t) \quad t \in [0, T_{\text{train}}], \quad \text{for } i = 1, \dots, N_{\text{train}}.$$

We remark that, due to the non-intrusive nature of our method, there is no need to retain the FOM states $\mathbf{y}(t)$. Finally, we train the ANN weights $\widehat{\mathbf{w}}$ by minimizing the discrepancy between the training data and the model outputs, that is by considering the following constrained optimization problem

$$\begin{cases} \widehat{\mathbf{w}} = \underset{\mathbf{w} \in \mathbb{R}^{N_w}}{\operatorname{argmin}} \left[\sum_{i=1}^{N_{\text{train}}} \int_0^{T_{\text{train}}} |V_{LV}^i(t) - V_{LV}^{ANN}(\mathbf{z}^i(t))|^2 dt + \beta |\mathbf{w}|^2 \right] \\ \text{such that, for each } i = 1, \dots, N_{\text{train}}: \\ \frac{d\mathbf{z}^i(t)}{dt} = \mathcal{NN}\left(\mathbf{z}^i(t), p_{LV}^i(t), \cos(\frac{2\pi t}{T_{HB}}), \sin(\frac{2\pi t}{T_{HB}}); \mathbf{p}_{\mathcal{M}}^i; \mathbf{w}\right) & \text{for } t \in (0, T_{\text{train}}], \\ \mathbf{z}^i(0) = \mathbf{z}_0, \end{cases} \quad (4)$$

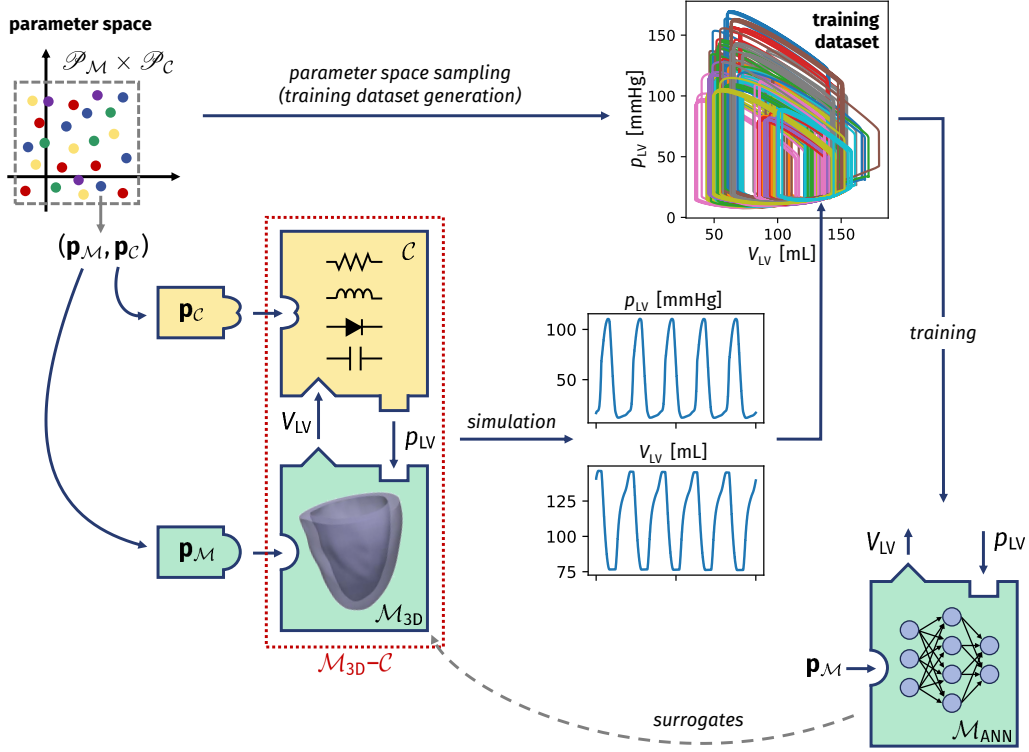


Figure 2: Training pipeline of the \mathcal{M}_{ANN} model. First, we sample the parameter space $\mathcal{P}_{\mathcal{M}} \times \mathcal{P}_{\mathcal{C}}$ (top left figure) and, for each parameter instance $(\mathbf{p}_{\mathcal{M}}, \mathbf{p}_{\mathcal{C}})$, we simulate some heartbeats through the $\mathcal{M}_{3\text{D}}-\mathcal{C}$ model (center figure). Finally, from the training set obtained by collecting the resulting pressure and volume transients (top right figure), we train the ANN-based model \mathcal{M}_{ANN} (bottom right figure), according to Eq. (4).

where $\beta > 0$ is a regularization hyperparameter. We remark that the optimization problem (4) is not a standard machine learning problem of data fitting. Indeed, the ANN appears at the right-hand side of a differential equation that acts as a constraint under which the loss function is minimized. To train this model, we use the algorithm proposed in [47], which envisages approximating the differential equation by the Forward Euler method, the loss function by the trapezoidal method, and then computing the gradients by solving the adjoint equations. The parameters are then optimized by means of the Levenberg-Marquardt method [35]. The training pipeline is summarized in Fig. 2.

Once the ANN has been trained (that is, the optimal weights $\hat{\mathbf{w}}$ have been determined), the \mathcal{M}_{ANN} model can be used as a surrogate of the $\mathcal{M}_{3\text{D}}$ model, also for different combinations of the parameters than those contained in the training set. Moreover, it can be coupled with the closure relationships \mathcal{C} , thus obtaining the $\mathcal{M}_{\text{ANN}}-\mathcal{C}$ model. For example, by considering the case of a

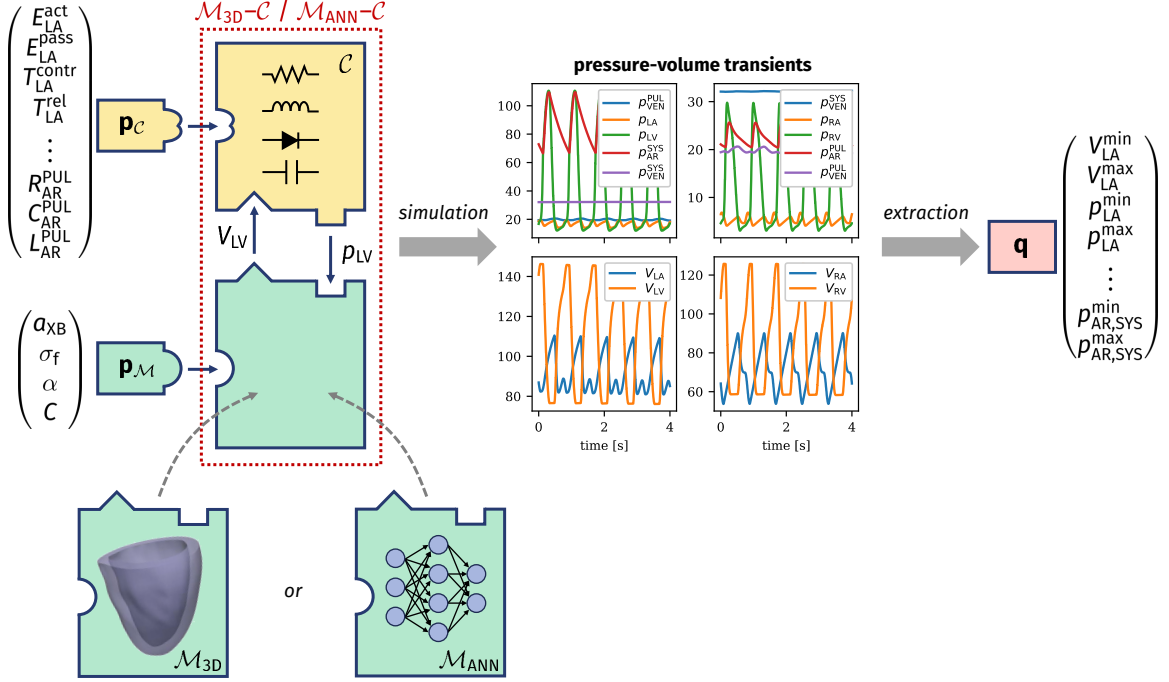


Figure 3: Parameters-to-QoIs computation using either the \mathcal{M}_{3D} or the \mathcal{M}_{ANN} model. Given a parameter instance $(\mathbf{p}_M, \mathbf{p}_C)$, either the \mathcal{M}_{3D} or the \mathcal{M}_{ANN} model can be coupled with the \mathcal{C} model to obtain pressure and volume transients, from which a set of QoIs are extracted. See Tabs. 1, 2 and 3 for the definition of \mathbf{p}_M , \mathbf{p}_C and \mathbf{q} , respectively.

closed-loop circulation model, as in (2), its reduced counterpart \mathcal{M}_{ANN-C} reads

$$\left\{ \begin{array}{ll} \frac{d\mathbf{z}(t)}{dt} = \mathcal{NN}\left(\mathbf{z}(t), p_{LV}(t), \cos\left(\frac{2\pi t}{T_{HB}}\right), \sin\left(\frac{2\pi t}{T_{HB}}\right); \mathbf{p}_M; \hat{\mathbf{w}}\right) & \text{for } t \in (0, T], \\ \frac{d\mathbf{c}(t)}{dt} = \mathbf{f}(\mathbf{c}(t), p_{LV}(t), t; \mathbf{p}_C) & \text{for } t \in (0, T], \\ V_{LV}^{0D}(\mathbf{c}(t)) = V_{LV}^{ANN}(\mathbf{z}(t)) & \text{for } t \in (0, T], \\ \mathbf{z}(0) = \mathbf{z}_0, & \\ \mathbf{c}(0) = \mathbf{c}_0. & \end{array} \right. \quad (5)$$

As matter of fact, the \mathcal{M}_{3D} model has the same input-output structure of the model being surrogated, that is \mathcal{M}_{ANN} . Hence, the latter can be employed in replacement of the former in approximating the outputs associated with a given set of parameters $(\mathbf{p}_M, \mathbf{p}_C)$, as shown in Fig. 3.

2.3 Hyperparameters tuning

Like any machine learning algorithm, our method depends on a set of *hyperparameters*, that is variables that are not trained (they are not part of the vector $\hat{\mathbf{w}}$), but are used to control the training process. These include ANN architecture hyperparameters (namely the number of layers N_{layers} and the number of neurons per layer $N_{neurons}$), the regularization factor β and the number

of reduced states N_z . To tune the hyperparameters, we rely on a k -fold cross-validation procedure. Specifically, after splitting the training set into $k = 5$ non-overlapping subsets, we cyclically train the model by excluding one subset that is used as validation set. Finally, we evaluate the average validation errors and we select the hyperparameters setting that attains the lowest validation error and better generalization properties (see Appendix A.4 for more details).

2.4 Global sensitivity analysis

To assess how much each parameter contributes to the determination of a QoI, e.g. a biomarker of clinical interest, we perform a global sensitivity analysis. This is typically done by sampling the parameter space and by computing suitable indicators, such as Borgonovo indices [43], Sobol indices [24, 56], Morris elementary effects [34], ANCOVA indices [67] and Kucherenko indices [28]. In this work, we perform a variance-based sensitivity analysis, which relies on a probabilistic approach. We compute Sobol indices to quantify the sensitivity of a QoI (say \mathbf{q}_j) with respect to a parameter (say \mathbf{p}_i). Specifically, the so-called *first-order Sobol index* (denoted by S_{ij}) indicates the impact on the j -th QoI (i.e. \mathbf{q}_j) of the i -th parameter (i.e. \mathbf{p}_i) when the latter varies alone, according to the definition

$$S_{ij} = \frac{\text{Var}_{\mathbf{p}_i} [\mathbb{E}_{\mathbf{p}_{\sim i}} [\mathbf{q}_j | \mathbf{p}_i]]}{\text{Var} [\mathbf{q}_j]},$$

where $\mathbf{p}_{\sim i}$ indicates the set of all parameters excluding the i -th one. The first-order Sobol index S_{ij} , however, only accounts for the variations of the parameter \mathbf{p}_i alone, averaged over variations in the other parameters, and thus does not account for the *interactions* among parameters. Conversely, it is possible to assess the importance of a parameter in determining a QoI, also accounting for the interactions among parameters, by resorting to the so-called *total-effect Sobol index* S_{ij}^T , defined as

$$S_{ij}^T = \frac{\mathbb{E}_{\mathbf{p}_{\sim i}} [\text{Var}_{\mathbf{p}_i} [\mathbf{q}_j | \mathbf{p}_{\sim i}]]}{\text{Var} [\mathbf{q}_j]} = 1 - \frac{\text{Var}_{\mathbf{p}_{\sim i}} [\mathbb{E}_{\mathbf{p}_i} [\mathbf{q}_j | \mathbf{p}_{\sim i}]]}{\text{Var} [\mathbf{q}_j]}.$$

The latter index quantifies the impact of a given parameter when it varies alone or together with other parameters [56].

To compute an estimate of the Sobol indices, we use the Saltelli method [24, 53], that makes use of Sobol quasi-random sequences to approximate the integrals that need to be computed. In practice, this method requires evaluating the model for a large number of parameters, and then processing the obtained QoIs to provide an estimate of the Sobol indices.

In this paper we perform a variance-based global sensitivity analysis simultaneously with respect to the circulation model parameters \mathbf{p}_C and the electromechanical model parameters \mathbf{p}_E (that is, we set $\mathbf{p} = (\mathbf{p}_M, \mathbf{p}_C)$). To this end, we use model $\mathcal{M}_{\text{ANN-C}}$ as a surrogate for model \mathcal{M}_{3D-C} to perform the evaluations required by the Saltelli method. As we will see in Sec. 3, using $\mathcal{M}_{\text{ANN-C}}$ instead of \mathcal{M}_{3D-C} entails a huge computational gain. In addition, for each parameter setting we simulate a certain number of heartbeats to achieve a limit cycle (i.e. a periodic solution), and we calculate QoIs with respect to the last cycle, which is the most significant one (as it removes the effects of incorrect initializations of the state variables). We remark that the need to simulate a certain number of heartbeats to overpass the transient phase (that typically lasts from 5 to 15 cycles, see Appendix A.2) makes the use of a reduced cardiac electromechanical model even more necessary.

2.5 Parameter estimation under uncertainty

The patient-specific personalization of a cardiac electromechanical model requires, besides the usage of a geometry derived from imaging data, the estimation of the parameters associated with the

mathematical model (or at least the most important ones), starting from clinical measurements. Very often, only a few scalar quantities are available for this purpose; moreover, the resolution of this inverse problem (i.e. estimating \mathbf{p} from \mathbf{q}) should account for the noise that unavoidably affects the measurement of \mathbf{q} and that reflects in uncertainty on \mathbf{p} .

Bayesian methods, such as MCMC [6] and Variational Inference [22], permit to address these issues within a rigorous statistical framework, by providing the *likelihood*, expressed as a probability distribution of the parameters values, given the observed QoIs (denoted by \mathbf{q}_{obs}). Unlike methods that provide point estimates, either based on gradient-descent [35] or genetic [23] optimization algorithms, Bayesian methods are able to provide a probability distribution on the parameter space, encoding the *credibility* of each parameter combination. The credibility computation accounts for the uncertainty on measurements (associated with measurement noise and encoded in the noise covariance matrix Σ), as well as a *prior distribution* on parameters (denoted by $\pi_{\text{prior}}(\mathbf{p})$), that is previous knowledge or belief about the parameters. By denoting the parameters-to-QoIs map by $\mathcal{F}: \mathbf{p} \mapsto \mathbf{q}$, we have $\mathbf{q}_{\text{obs}} = \mathcal{F}(\mathbf{p}) + \epsilon$, where $\epsilon \sim \mathcal{N}(\cdot | \mathbf{0}, \Sigma)$ denotes the measurement error (that we assume for simplicity to be distributed as a Gaussian random variable). Bayes' theorem states that the *posterior distribution* of parameters, that is the degree of belief of their value after having observed \mathbf{q}_{obs} , is given by

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

where the normalization constant Z is defined as

$$Z = \int_{\mathcal{P}} \mathcal{N}(\mathbf{q}_{\text{obs}} | \mathcal{F}(\hat{\mathbf{p}}), \Sigma) d\pi_{\text{prior}}(\hat{\mathbf{p}}).$$

In practice, the computation of π_{post} may be challenging from the computational viewpoint, because of the need to approximate the integral that defines Z . The MCMC method permits to approximate the distribution π_{post} with a relatively small computational effort. Similarly to the Saltelli method that we use for sensitivity analysis, also the MCMC method requires a large number of model evaluations, for different parameter values. Moreover, this method is non intrusive, that is it only requires evaluations of the map $\mathcal{F}: \mathbf{p} \mapsto \mathbf{q}$. Therefore, we can employ for this purpose the $\mathcal{M}_{\text{ANN-C}}$ model as a surrogate for the $\mathcal{M}_{3\text{D-C}}$ model, which drastically reduces the necessary computational time.

Moreover, we remark that the Bayesian framework permits to rigorously account for the approximation error introduced by replacing the high fidelity model $\mathcal{M}_{3\text{D-C}}$ by its surrogate $\mathcal{M}_{\text{ANN-C}}$. Indeed, if we denote by $\tilde{\mathcal{F}}$ the approximated parameters-to-QoIs map represented by the surrogate model $\mathcal{M}_{\text{ANN-C}}$, we have $\mathcal{F}(\mathbf{p}) = \tilde{\mathcal{F}}(\mathbf{p}) + \epsilon_{\text{ROM}}$, where ϵ_{ROM} is the ROM approximation error. It follows $\mathbf{q}_{\text{obs}} = \tilde{\mathcal{F}}(\mathbf{p}) + \epsilon_{\text{ROM}} + \epsilon_{\text{exp}}$, where ϵ_{exp} is the experimental measurement error. Since the two sources of error can be assumed independent, the covariance of the total error $\epsilon = \epsilon_{\text{ROM}} + \epsilon_{\text{exp}}$ satisfies $\Sigma = \Sigma_{\text{ROM}} + \Sigma_{\text{exp}}$, where Σ_{ROM} is the ROM approximation error covariance (which can be estimated by evaluating the ROM on a testing set) and Σ_{exp} is the experimental error covariance (that depends on the measurement protocol at hand). This permits to take into account in the estimation process the error introduced by the surrogate model.

To assess the capability of our ROM to accelerate the estimation of parameters for multiscale cardiac electromechanical models, we perform the following test. First, we perform a simulation with the $\mathcal{M}_{3\text{D-C}}$ model, from which we derive a set of QoIs (\mathbf{q}_{obs}). Then, by employing the $\mathcal{M}_{\text{ANN-C}}$ model as a surrogate of the $\mathcal{M}_{3\text{D-C}}$ model, we obtain a Bayesian estimate of the parameters, that we validate against the values used to generate \mathbf{q}_{obs} .

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

Table 1: Parameters $\mathbf{p}_{\mathcal{M}}$ of the \mathcal{M}_{3D} model considered in this work and associated baseline values.

2.6 The cardiac electromechanical model

As mentioned above, due to its non-intrusive nature, our method is not limited to a specific cardiac electromechanical model, but can be applied to virtually any electromechanical model as long as pressure and volume transients are available for the training procedure. In this section, we briefly introduce the specific model used to produce the numerical results of this paper.

We consider a LV geometry processed from the Zygote 3D heart model [27] endowed with a fiber architecture generated by means of the Bayer-Blake-Plank-Trayanova algorithm [4, 42]. Before starting the simulations, we recover the reference unstressed configuration through the algorithm that we proposed in [50]. To model the action potential propagation, we employ the monodomain equation [8], coupled with the *ten Tusscher-Panfilov* ionic model [60]. We model the microscale generation of active force through the biophysically detailed RDQ20-MF model [46], that is coupled, within an active stress approach, with the elastodynamics equations describing tissue mechanics. On the other hand, the passive behavior of the tissue is modeled through a quasi-incompressible exponential constitutive law [61]. We model the interaction with the pericardium by means of spring-damper boundary conditions at the epicardium, while we adopt energy-consistent boundary conditions [48] to model the interaction with the part of the myocardium beyond the artificial ventricular base. To model blood circulation, that is \mathcal{C} , we rely on the 0D closed-loop model presented in [50], consisting of a compartmental description of the cardiac chambers, systemic and pulmonary, arterial and venous circulatory networks, based on an electrical analogy. The different compartments are modeled as RLC (resistance, inductance, capacitance) circuits, while cardiac valves are described as diodes.

Among the parameters associated with the \mathcal{M}_{3D} model, in this work we focus on the four ones reported in Tab. 1. Similarly, we report in Tab. 2 the parameters associated with the \mathcal{C} model.

We report in Tab. 3 the list of all the QoIs, computed from the solution of the \mathcal{M}_{3D} - \mathcal{C} model, and outputs of interest that are used through this paper. The last column of the table indicates which variables are used for cross-validation during the training phase (see Sec. 2.3), those that are used for sensitivity analysis purposes (see Sec. 2.4) and those that are used to run Bayesian parameter estimations (see Sec. 2.5).

To numerically approximate this multiphysics and multiscale model, we adopt the segregated approach proposed in [51], by which the subproblems are solved sequentially. For space discretization, we rely on bilinear Finite Elements defined on hexahedral meshes, adopting a different spatial resolution for the electrophysiological and the mechanical variables. For time discretization, we employ a staggered scheme, where different time steps are used according to the specific subproblem. Moreover, to avoid the numerical oscillations arising from the mechanical feedback on force generation (that are commonly cured by recurring to a monolithic scheme [30, 40]), we use the stabilized-staggered scheme that we proposed in [52]. This numerical model requires, on a 32 cores computer platform, nearly 4 hours of computational time to simulate a heartbeat. More details on the numerical discretization and the computational platform employed to generate the training data

Parameter	Baseline	Unit	Description
$V_{\text{heart}}^{\text{tot}}$	400.15	mL	Initial blood pool volume of the heart
$E_{\text{LA}}^{\text{act}}, E_{\text{RA}}^{\text{act}}, E_{\text{RV}}^{\text{act}}$	0.07, 0.06, 0.55	mmHg mL ⁻¹	LA/RA/RV active elastance
$E_{\text{LA}}^{\text{pass}}, E_{\text{RA}}^{\text{pass}}, E_{\text{RV}}^{\text{pass}}$	0.18, 0.07, 0.05	mmHg mL ⁻¹	LA/RA/RV passive elastance
$T_{\text{LA}}^{\text{contr}}, T_{\text{RA}}^{\text{contr}}, T_{\text{RV}}^{\text{contr}}$	0.14, 0.14, 0.20	s	LA/RA/RV contraction time
$T_{\text{LA}}^{\text{rel}}, T_{\text{RA}}^{\text{rel}}, T_{\text{RV}}^{\text{rel}}$	0.14, 0.14, 0.32	s	LA/RA/RV relaxation time
$V_{0,\text{LA}}, V_{0,\text{RA}}, V_{0,\text{RV}}$	4.0, 4.0, 16.0	mL	LA/RA/RV reference volume
$t_{\text{L}}^{\text{av}}, t_{\text{R}}^{\text{av}}$	0.16, 0.16	s	Left/Right atrioventricular delay
$R_{\text{min}}, R_{\text{max}}$	0.0075, 75006.2	mmHg s mL ⁻¹	Valve minimum/maximum resistance
$R_{\text{AR}}^{\text{SYS}}, R_{\text{VEN}}^{\text{SYS}}$	0.64, 0.32	mmHg s mL ⁻¹	Systemic arterial/venous resistance
$R_{\text{AR}}^{\text{PUL}}, R_{\text{VEN}}^{\text{PUL}}$	0.032, 0.036	mmHg s mL ⁻¹	Pulmonary arterial/venous resistance
$C_{\text{AR}}^{\text{SYS}}, C_{\text{VEN}}^{\text{SYS}}$	1.2, 60.0	mL mmHg ⁻¹	Systemic arterial/venous capacitance
$C_{\text{AR}}^{\text{PUL}}, C_{\text{VEN}}^{\text{PUL}}$	10.0, 16.0	mL mmHg ⁻¹	Pulmonary arterial/venous capacitance
$L_{\text{AR}}^{\text{SYS}}, L_{\text{VEN}}^{\text{SYS}}$	0.005, 0.0005	mmHg s ² mL ⁻¹	Systemic arterial/venous inductance
$L_{\text{AR}}^{\text{PUL}}, L_{\text{VEN}}^{\text{PUL}}$	0.0005, 0.0005	mmHg s ² mL ⁻¹	Pulmonary arterial/venous inductance

Table 2: Parameters $\mathbf{p}_{\mathcal{C}}$ of the \mathcal{C} model considered in this work and associated baseline values (LA = left atrium; RA = right atrium; RV = right ventricle).

Parameter	Unit	Description	Usage
Left Atrium			
$V_{\text{LA}}^{\text{min}}, V_{\text{LA}}^{\text{max}}$	mL	End-systolic and end-diastolic volume	GSA
$p_{\text{LA}}^{\text{min}}, p_{\text{LA}}^{\text{max}}$	mmHg	Minimum and maximum pressure	GSA
Left ventricle			
$V_{\text{LV}}(t)$	mL	Volume transient	X-validation
$p_{\text{LV}}(t)$	mmHg	Pressure transient	X-validation
$V_{\text{LV}}^{\text{min}}, V_{\text{LV}}^{\text{max}}$	mL	End-systolic and end-diastolic volume	X-validation, GSA
$p_{\text{LV}}^{\text{min}}, p_{\text{LV}}^{\text{max}}$	mmHg	Minimum and maximum pressure	X-validation, GSA
SV_{LV}	mL	Stroke volume ($V_{\text{LV}}^{\text{max}} - V_{\text{LV}}^{\text{min}}$)	X-validation, GSA
Right Atrium			
$V_{\text{RA}}^{\text{min}}, V_{\text{RA}}^{\text{max}}$	mL	End-systolic and end-diastolic volume	GSA
$p_{\text{RA}}^{\text{min}}, p_{\text{RA}}^{\text{max}}$	mmHg	Minimum and maximum pressure	GSA
Right ventricle			
$V_{\text{RV}}^{\text{min}}, V_{\text{RV}}^{\text{max}}$	mL	End-systolic and end-diastolic volume	GSA
$p_{\text{RV}}^{\text{min}}, p_{\text{RV}}^{\text{max}}$	mmHg	Minimum and maximum pressure	GSA
SV_{RV}	mL	Stroke volume ($V_{\text{RV}}^{\text{max}} - V_{\text{RV}}^{\text{min}}$)	GSA
Systemic arterial circulation			
$p_{\text{AR},\text{SYS}}^{\text{min}}, p_{\text{AR},\text{SYS}}^{\text{max}}$	mmHg	Minimum and maximum pressure	GSA, MCMC

Table 3: List of QoIs used through this paper, either for cross-validation (X-validation), global sensitivity analysis (GSA) or MCMC based Bayesian parameter estimation (MCMC).

used in this paper are available in Appendix A.1.

2.7 Software libraries

The cardiac electromechanics simulations considered in this paper are performed by means of the `lifex` library¹, a high-performance C++ platform developed within the iHEART project². To train the ANN-based models, we employ the open source MATLAB library `model-learning`³, that implements the machine learning method proposed in [47] and used in this paper. Sensitivity analysis is carried out through the open source Python library `SALib`⁴ [20]. Finally, for the MCMC based Bayesian parameter estimation we rely on the open source Python library `UQpy`⁵ [37]. To employ the ANN-based models trained with the MATLAB library `model-learning` within the Python environment of `SALib` and `UQpy`, we exploit `pyModelLearning`, a Python wrapper for the `model-learning` library, which is available in its GitHub repository⁶.

Both the MATLAB codes that are used to train the ANN-based ROMs and the datasets that permit to reproduce the numerical results are publicly available in the online repository accompanying this manuscript⁷.

3 Results

3.1 Trained models

To generate the pressure and volume transients needed to train an ANN-based ROM, we sample the parameter space $\mathcal{P}_{\mathcal{M}} \times \mathcal{P}_{\mathcal{C}}$ with a Monte Carlo approach, even if more sophisticated sampling strategies – such as Latin Hypercube Design – can be considered as well. For simplicity, in this stage we only consider a subset of the parameters $\mathbf{p}_{\mathcal{C}}$, selected as the most significant ones, on the basis of a preliminary variance-based global sensitivity analysis, obtained with a version of the closed-loop model in which the LV is also represented by a 0D circuit element (see Appendix A.3).

We consider two scenarios. First, we study the variability with respect to a single parameter, namely the active contractility. Thus, we set $\mathbf{p}_{\mathcal{M}} = [a_{\text{XB}}]$. Under this setting, we generate 30 numerical simulations through the $\mathcal{M}_{3\text{D}}\text{-}\mathcal{C}$ model to train a ROM, henceforth denoted by $\mathcal{M}_{\text{ANN}}^{\text{single}}$. For this ROM, the remaining parameters (i.e. σ_{f} , α and C) are kept constant (more precisely, equal to the baseline values of Tab. 1). Then, we consider the full parametric variability (that is, we set $\mathbf{p}_{\mathcal{M}} = [a_{\text{XB}}, \sigma_{\text{f}}, \alpha, C]$), we generate 40 training samples and we train a second ROM, denoted by $\mathcal{M}_{\text{ANN}}^{\text{full}}$. All the numerical simulations included in the training set are 5 heartbeats long. The optimal sets of hyperparameters, tuned through the algorithm of Sec. 2.3, are reported in Tab. 4. In both the considered cases, training an ANN-based model takes approximately 18 hours on a single core standard laptop.

Once trained, the two ROMs ($\mathcal{M}_{\text{ANN}}^{\text{single}}$ and $\mathcal{M}_{\text{ANN}}^{\text{full}}$) can be coupled with the circulation model \mathcal{C} , thus obtaining the models $\mathcal{M}_{\text{ANN}}^{\text{single}}\text{-}\mathcal{C}$ and $\mathcal{M}_{\text{ANN}}^{\text{full}}\text{-}\mathcal{C}$, according to Eq. (5). These two models represent two surrogates of the $\mathcal{M}_{3\text{D}}\text{-}\mathcal{C}$ model, capable of approximating its output at a dramatically reduced

¹<https://lifex.gitlab.io>

²iHEART - An Integrated Heart Model for the simulation of the cardiac function, European Research Council (ERC) grant agreement No 740132, P.I. Prof. A. Quarteroni

³<https://model-learning.readthedocs.io/>

⁴<https://salib.readthedocs.io/>

⁵<https://uqpyproject.readthedocs.io/>

⁶<https://github.com/FrancescoRegazzoni/model-learning>

⁷<https://github.com/FrancescoRegazzoni/cardioEM-learning>

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

Table 4: Optimal sets of hyperparameters for the two trained models $\mathcal{M}_{\text{ANN}}^{\text{single}}$ and $\mathcal{M}_{\text{ANN}}^{\text{full}}$.

computational cost. As a matter of fact, numerical simulations with either the $\mathcal{M}_{\text{ANN}}^{\text{single}}$ or the $\mathcal{M}_{\text{ANN}}^{\text{full}}$ model take nearly one second of computational time per heartbeat on a single core standard laptop.

To test the accuracy of the $\mathcal{M}_{\text{ANN}}^{\text{single}}\text{-}\mathcal{C}$ and $\mathcal{M}_{\text{ANN}}^{\text{full}}\text{-}\mathcal{C}$ models with respect to the $\mathcal{M}_{3\text{D}}\text{-}\mathcal{C}$ model, we consider a testing dataset, by taking unobserved samples in the parameter space $\mathcal{P}_{\mathcal{M}} \times \mathcal{P}_{\mathcal{C}}$. For both models, we consider 15 testing simulations of the same duration of the ones included in the training set. In addition, to test the reliability of the ROMs over a longer time horizon than the one considered in the training set, we include in the testing set 5 simulations of double length (i.e. 10 heartbeats). Then, we compare the simulations obtained with the two ROMs ($\mathcal{M}_{\text{ANN}}^{\text{single}}$ and $\mathcal{M}_{\text{ANN}}^{\text{full}}$) with the ones obtained with the $\mathcal{M}_{3\text{D}}\text{-}\mathcal{C}$ model for the same parameters $\mathbf{p}_{\mathcal{M}}$ and $\mathbf{p}_{\mathcal{C}}$.

The accuracy of the two ROMs is summarized Tabs. 5 and 6. Specifically, in Tab. 5 we report metrics regarding the ability of the ROMs to correctly predict the function of the LV, that is the chamber surrogated by the ANN-based model. Specifically, we report the relative errors in L^2 norm (i.e. the mean square errors) associated with pressure and volume transients (p_{LV} and V_{LV}) and the relative errors on some biomarkers of clinical interest (maximum and minimum pressures and volumes). For the latter, we also report the coefficient of determination R^2 . We notice that both model $\mathcal{M}_{\text{ANN}}^{\text{single}}\text{-}\mathcal{C}$ and model $\mathcal{M}_{\text{ANN}}^{\text{full}}\text{-}\mathcal{C}$ are able to surrogate model $\mathcal{M}_{3\text{D}}\text{-}\mathcal{C}$ with remarkably good accuracy. Model $\mathcal{M}_{\text{ANN}}^{\text{full}}\text{-}\mathcal{C}$ features slightly larger errors than model $\mathcal{M}_{\text{ANN}}^{\text{single}}\text{-}\mathcal{C}$; this is not surprising since model $\mathcal{M}_{\text{ANN}}^{\text{full}}\text{-}\mathcal{C}$ explores a much larger parametric space than model $\mathcal{M}_{\text{ANN}}^{\text{single}}\text{-}\mathcal{C}$ (four parameters instead of one). Interestingly, the errors obtained over a long time horizon are very similar to those obtained for simulations of the same duration as those used to train the ANNs. This demonstrates the reliability of the ROMs in long-term simulations. Similarly, in Tab. 6 we report the errors and R^2 coefficients associated with the RV output (for simplicity, we here consider only 5 heartbeats long simulations). The accuracy obtained in reproducing the RV function is even better than that for the LV, coherently with the fact that the RV is included in the \mathcal{C} model and thus it is not surrogated by the ANN-based model.

In Figs. 4 and 5 we compare 10 heartbeats long transients obtained with the $\mathcal{M}_{\text{ANN}}^{\text{single}}\text{-}\mathcal{C}$ and $\mathcal{M}_{\text{ANN}}^{\text{full}}\text{-}\mathcal{C}$ models, respectively, to those obtained with the $\mathcal{M}_{3\text{D}}\text{-}\mathcal{C}$ model. All these transients were not used to train the ANN-based models (that is, they belong to the testing set). In Fig. 6 we show the LV biomarkers predicted by the $\mathcal{M}_{\text{ANN}}^{\text{single}}\text{-}\mathcal{C}$ and $\mathcal{M}_{\text{ANN}}^{\text{full}}\text{-}\mathcal{C}$ models versus those predicted by the $\mathcal{M}_{3\text{D}}\text{-}\mathcal{C}$ model.

3.2 Coupling the electromechanical reduced-order model with different circulation models

To generate the data used to train the \mathcal{M}_{ANN} model, we employ the $\mathcal{M}_{3\text{D}}\text{-}\mathcal{C}$ coupled model (that is, we couple the $\mathcal{M}_{3\text{D}}$ model with the circulation model \mathcal{C}). However, the ANN-based ROM \mathcal{M}_{ANN} surrogates the $\mathcal{M}_{3\text{D}}$ model regardless of its coupling with a specific circulation model. In fact, thanks to Eq. (5), the trained \mathcal{M}_{ANN} model can be coupled with circulation models that are different from

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

Table 5: Testing errors and R^2 coefficients on the LV outputs obtained with the two models $\mathcal{M}_{ANN}^{\text{single}}$ and $\mathcal{M}_{ANN}^{\text{full}}$, for 5 heartbeats long (top) and 10 heartbeats long (bottom) simulations.

		5 heartbeats					
		$p_{RV}(t)$	$V_{RV}(t)$	p_{RV}^{\min}	p_{RV}^{\max}	V_{RV}^{\min}	V_{RV}^{\max}
$\mathcal{M}_{ANN}^{\text{single}}\text{-}\mathcal{C}$ vs $\mathcal{M}_{3D}\text{-}\mathcal{C}$	relative error	0.0048	0.0035	0.0015	0.0027	0.0028	0.0004
	$R^2[\%]$			100.000	99.993	99.998	100.000
$\mathcal{M}_{ANN}^{\text{full}}\text{-}\mathcal{C}$ vs $\mathcal{M}_{3D}\text{-}\mathcal{C}$	relative error	0.0069	0.0040	0.0029	0.0079	0.0072	0.0069
	$R^2[\%]$			99.994	99.807	99.819	99.997

Table 6: Testing errors and R^2 coefficients on the RV outputs obtained with the two models $\mathcal{M}_{ANN}^{\text{single}}$ and $\mathcal{M}_{ANN}^{\text{full}}$, for 5 heartbeats long simulations.

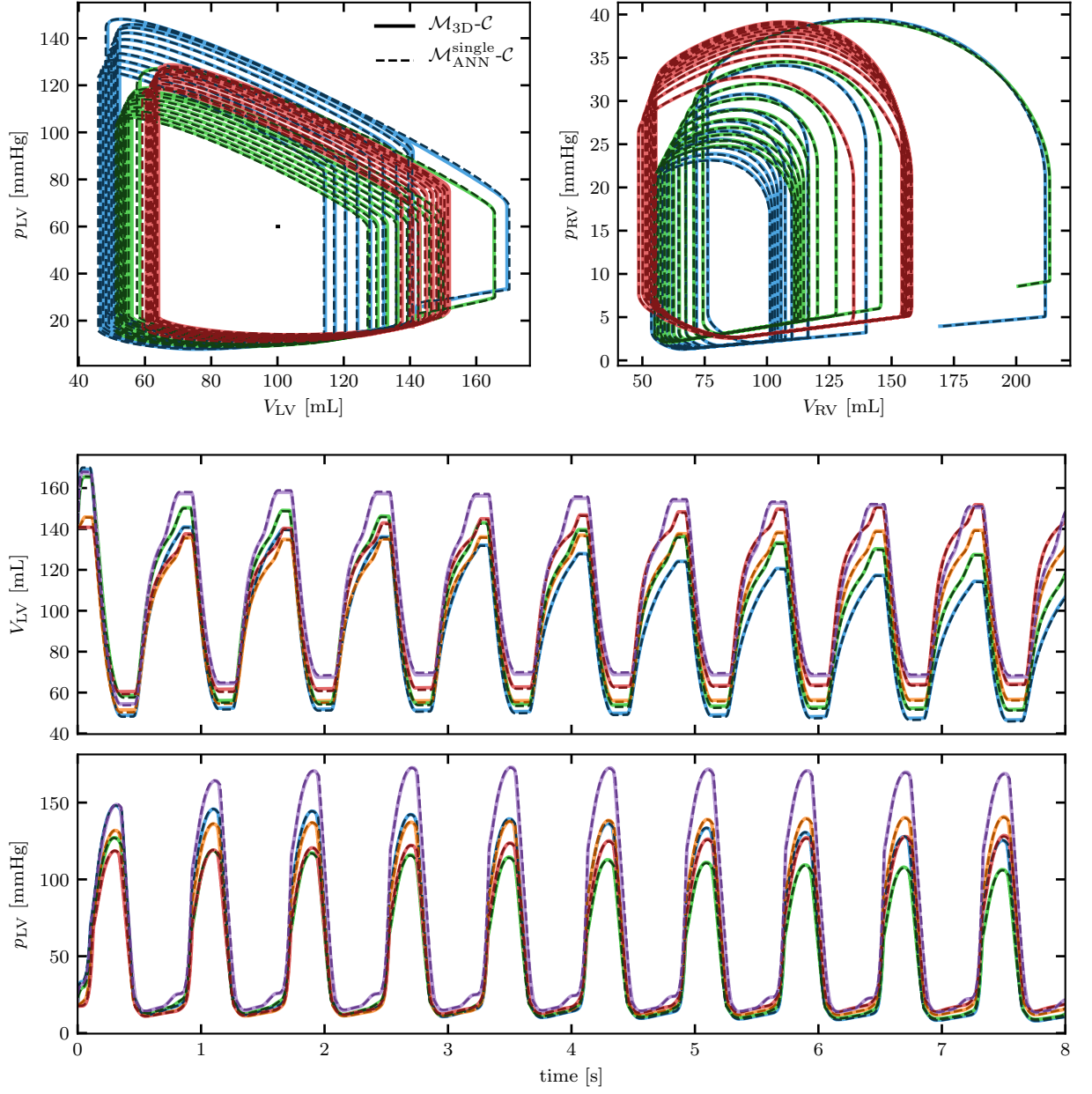


Figure 4: Pressure and volume transients obtained with the $\mathcal{M}_{ANN}^{single-C}$ (dashed lines), compared to those obtained with the \mathcal{M}_{3D-C} model (solid lines). The different colors correspond to different samples of the testing set. For the sake of clarity, only three samples are shown in the first row.

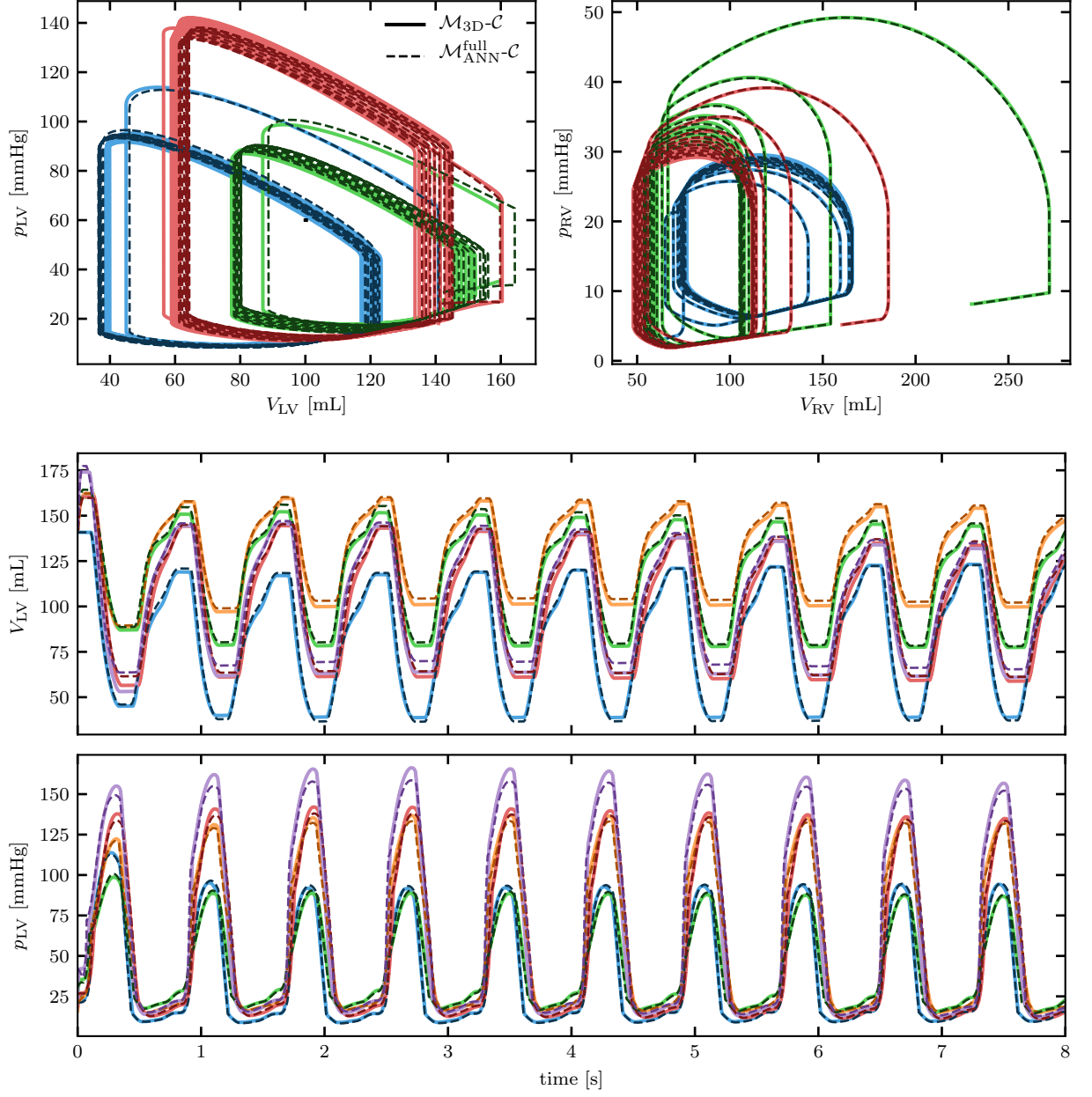


Figure 5: Pressure and volume transients obtained with the $\mathcal{M}_{ANN-C}^{full}$ (dashed lines), compared to those obtained with the \mathcal{M}_{3D-C} model (solid lines). The different colors correspond to different samples of the testing set. For the sake of clarity, only three samples are shown in the first row.

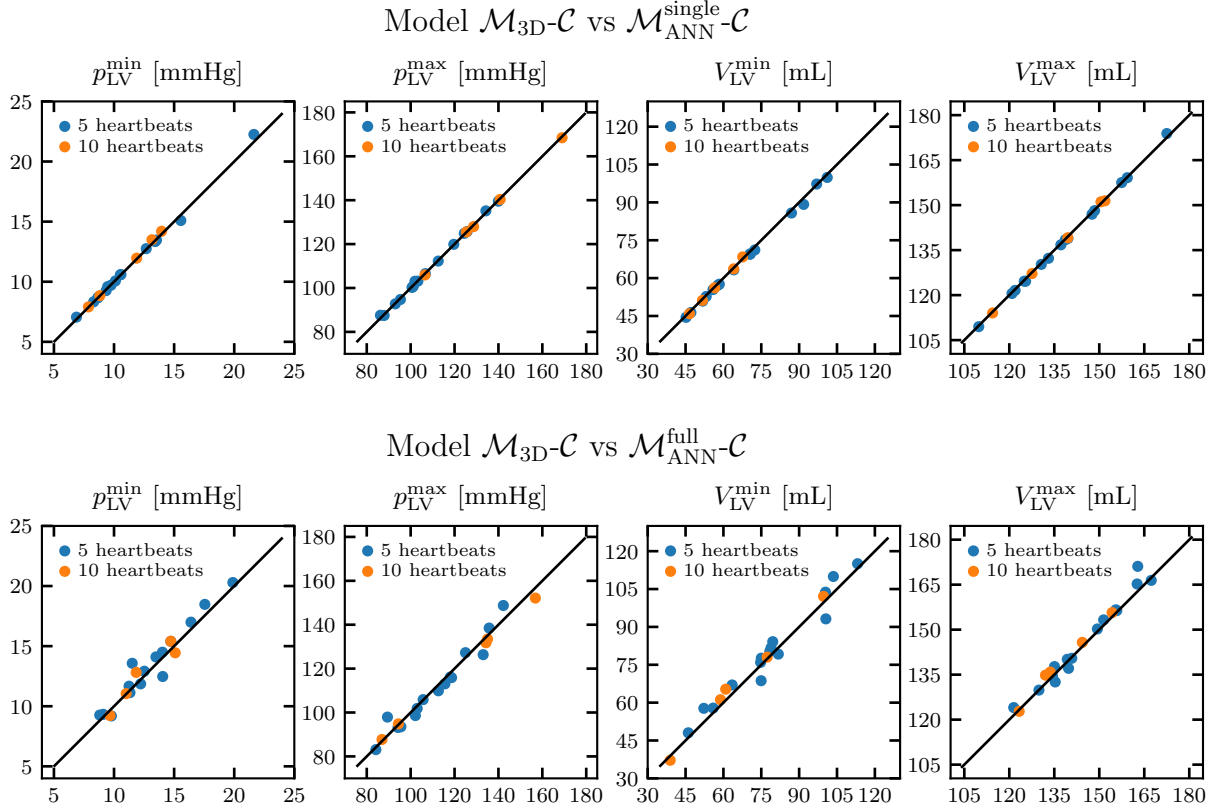


Figure 6: LV biomarkers obtained with the $\mathcal{M}_{ANN-C}^{\text{single}}$ and $\mathcal{M}_{ANN-C}^{\text{full}}$ models versus those obtained with the \mathcal{M}_{3D-C} model in the testing set. The different marker colors are associated with 5 heartbeats and 10 heartbeats long simulations, respectively.

		$p_{LV}(t)$	$V_{LV}(t)$	p_{LV}^{\min}	p_{LV}^{\max}	V_{LV}^{\min}	V_{LV}^{\max}
$\mathcal{M}_{ANN}^{\text{single}}\text{-}\mathcal{C}'$ vs $\mathcal{M}_{3D}\text{-}\mathcal{C}'$	relative error	0.0459	0.0518	0.0065	0.0004	0.0080	0.0009
$\mathcal{M}_{ANN}^{\text{full}}\text{-}\mathcal{C}'$ vs $\mathcal{M}_{3D}\text{-}\mathcal{C}'$	relative error	0.0510	0.0219	0.0116	0.0027	0.0110	0.0063

Table 7: Testing errors on the LV outputs obtained with the two models $\mathcal{M}_{ANN}^{\text{single}}$ and $\mathcal{M}_{ANN}^{\text{full}}$ coupled with the \mathcal{C}' model for 5 heartbeats long simulations.

the one used to generate the training data. We remind that predictions are reliable only if pressure and volume values are inside the ranges explored during the training phase.

We demonstrate the flexibility of our approach by coupling the \mathcal{M}_{ANN} model with a pressure-volume closure relationship that is different from the closed-loop circulation model \mathcal{C} used during the training phase (see Sect. 2.6). Specifically, we consider a circulation model \mathcal{C}' with a windkessel type relationship during the ejection phase and a linear pressure ramp during the filling phase [48]. In spite of the different nature of the two circulation models (the one used during the training and the one used during the testing), the ANN-based ROM trained with the \mathcal{C} model proves to be reliable also when it is coupled with the \mathcal{C}' model. Indeed, as shown in Tab. 7, the results obtained by the $\mathcal{M}_{ANN}\text{-}\mathcal{C}'$ coupled model approximate those of the $\mathcal{M}_{3D}\text{-}\mathcal{C}'$ coupled model with an excellent accuracy. These errors are indeed comparable to the ones obtained by surrogating the $\mathcal{M}_{ANN}\text{-}\mathcal{C}$ model with the $\mathcal{M}_{3D}\text{-}\mathcal{C}$ model (see Tab. 5).

3.3 Global sensitivity analysis

Once we have verified that models $\mathcal{M}_{ANN}^{\text{single}}\text{-}\mathcal{C}$ and $\mathcal{M}_{ANN}^{\text{full}}\text{-}\mathcal{C}$ are able to reproduce the outputs of model $\mathcal{M}_{3D}\text{-}\mathcal{C}$ with high accuracy, we use them to perform a global sensitivity analysis. The aim is to determine which of the parameters of the circulation model (\mathbf{p}_C) and the electromechanical model (\mathbf{p}_M) contribute the most to the determination of a list of outputs of interest, the so-called QoIs (see Tab. 3). For this purpose, we compute the Sobol indices S_{ij} and S_{ij}^T , as described in Sec. 2.4. When replacing model $\mathcal{M}_{3D}\text{-}\mathcal{C}$ with a ROM, we can only estimate sensitivity indices with respect to the parameters \mathbf{p}_M that were considered during the training phase. On the other hand, we can compute sensitivity indices with respect to all parameters \mathbf{p}_C of the circulation model, even those that were not varied during training. It would even be possible – at least in principle – to consider a circulation model different from the one used to generate the training data. In fact, as pointed out in Sec. 2.2, the model $\mathcal{M}_{ANN}^{\text{single}}$ and $\mathcal{M}_{ANN}^{\text{full}}$ surrogate the models \mathcal{M}_{3D} independently of the circulation model to which it is coupled with.

We only report the results obtained by means of the most complete ROM, that is $\mathcal{M}_{ANN}^{\text{full}}\text{-}\mathcal{C}$. The results are shown in Figs. 7 and 8, respectively. First, we notice that the S_{ij} and S_{ij}^T indices have only small differences from each other. This means that the interaction among the different parameters is less significant, in the determination of the QoIs, than the variation of the individual parameters. Furthermore, we note that, as expected, the QoIs associated with a given chamber or compartment are mostly determined by the parameters associated with the same region of the cardiovascular network. However, there are some important exceptions. Indeed, the venous resistance of the systemic circulation R_{VEN}^{SYS} has a strong impact on the right heart (RA and RV), i.e. the part of the network located immediately upstream. In addition, the systemic arterial resistance R_{AR}^{SYS} significantly impacts the maximum LV pressure. In fact, this parameter is known to contribute in the determination of the so-called afterload [58]. We also note that total circulating blood volume $V_{\text{heart}}^{\text{tot}}$ has a major impact on almost all biomarkers. Conversely, the parameters associated with

the pulmonary circulation network have a minimal impact. In addition, parameters describing the resistance of opened and closed valves also have a very little impact. This is an interesting result, since it shows that these parameters, chosen as a very low and very high value respectively (since for reasons of numerical stability they cannot be set equal to zero and infinity), have virtually no impact (at least within the ranges considered here) on the output quantities of biomechanical interest.

Regarding individual compartments, we note that variability can be explained by only a few parameters. Specifically, atria are mainly influenced by passive stiffness and atrioventricular delay, whereas for the RV the most relevant parameters are active and passive stiffness. As expected, the biomarkers associated with the LV – the only chamber included in the \mathcal{M}_{3D} model – are mainly influenced by the parameters $\mathbf{p}_{\mathcal{M}}$. Among these, the most influential one is the active contractility a_{XB} , followed the fibers orientation α and, to a lesser extent, the passive stiffness C . Finally, the electrical conductivity σ_f has a minimal impact on the biomarkers under consideration.

It is important to note that Sobol indices are affected by the amplitude of the ranges in which the parameters are varied. In particular, the wider the range associated with a parameter, the greater the associated Sobol indices will be, as the parameter in question potentially generates greater variability in the QoI. Therefore, the results shown here are valid for the specific ranges we used, which are reported in Appendix A.5.

3.4 Bayesian parameter estimation

In this section we present a further practical use of the cardiac electromechanics ROM presented in this paper. In particular, we show that the ROM can be used to enable Bayesian parameter estimation for the $\mathcal{M}_{3D}\text{-}\mathcal{C}$ model, which, due to the prohibitive computational cost, would not be affordable without the use of a ROM.

First, we consider a prescribed value for the parameter vector $\mathbf{p} = (\mathbf{p}_{\mathcal{M}}, \mathbf{p}_{\mathcal{C}})$ (specifically, we employ the values reported in Tabs. 1 and 2) and we run a simulation using the $\mathcal{M}_{3D}\text{-}\mathcal{C}$ model. Based on the output of this simulation, we consider a couple of QoIs, consisting of minimum and maximum arterial pressure (i.e. we set $\mathbf{q} = (p_{AR,SYS}^{\min}, p_{AR,SYS}^{\max})$). The choice of these two QoIs is motivated by the fact that they are two variables that can be measured non invasively, and in fact they are often monitored in clinical routine. We reconstruct the value of a couple of parameters, namely the active contractility a_{XB} and the systemic arterial resistance R_{AR}^{SYS} , assuming known and keeping fixed the values of the remaining parameters. Indeed, we aim at demonstrating that the ROM we propose is suitable for estimating parameters of both the \mathcal{M}_{3D} model (such as a_{XB}) and of the \mathcal{C} model (such as R_{AR}^{SYS}). Specifically, in this section we rely on the $\mathcal{M}_{ANN}^{\text{single}}\text{-}\mathcal{C}$ model.

To mimic the presence of measurement errors, we artificially add to the exact values of the QoIs $p_{AR,SYS}^{\min}$ and $p_{AR,SYS}^{\max}$ a synthetic noise, with increasing magnitude. Specifically, we add an artificial noise by sampling from independent Gaussian variables with zero mean and with variance σ_{exp}^2 . We consider three cases: $\sigma_{\text{exp}}^2 = 0$ (i.e. no noise), $\sigma_{\text{exp}}^2 = 0.1 \text{ mmHg}^2$ and $\sigma_{\text{exp}}^2 = 1 \text{ mmHg}^2$.

As described in Sec. 2.5, we use the MCMC method to derive a Bayesian estimate of the parameters value from (possibly noisy) measurements of the minimum and maximum arterial pressure. For both unknown parameters we employ a non-informative prior, that is a uniform distribution on the ranges used to train the ROM ($a_{XB} \in [80, 320] \text{ MPa}$ and $R_{AR}^{SYS} \in [0.4, 1.2] \text{ mmHg s mL}^{-1}$). According to Sec. 2.5, we set $\Sigma = \Sigma_{\text{ROM}} + \Sigma_{\text{exp}}$, where the experimental measurement error covariance is given by $\Sigma_{\text{exp}} = \sigma_{\text{exp}}^2 \mathbb{I}_2$ (\mathbb{I}_2 being the 2-by-2 identity matrix) and where the ROM approximation error covariance is estimated from its statistical distribution on the validation set as $\Sigma_{\text{ROM}} = 0.2 \text{ mmHg}^2 \mathbb{I}_2$. More details on the MCMC setup are available in Appendix A.6.

In Fig. 9 we show the posterior distribution π_{post} on the parameters pair (a_{XB}, R_{AR}^{SYS}) obtained for the three noise levels considered. We identify with a red line the 90% credibility region, that

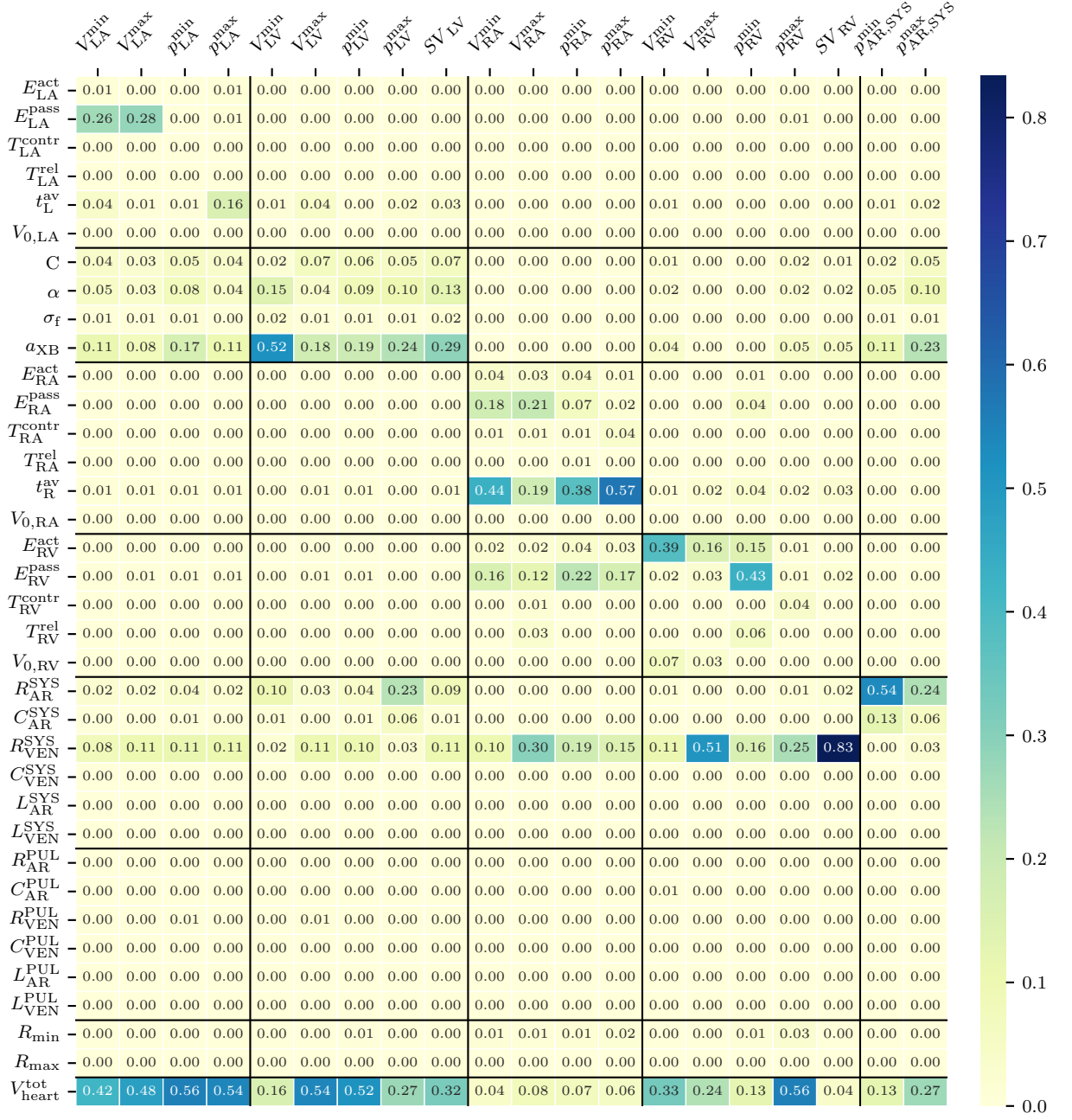


Figure 8: Total-effect Sobol indices S_{ij}^T computed by exploiting the $\mathcal{M}_{\text{ANN}}^{\text{full}}\text{-}\mathcal{C}$ model. For a description of the figure see caption of Fig. 7.

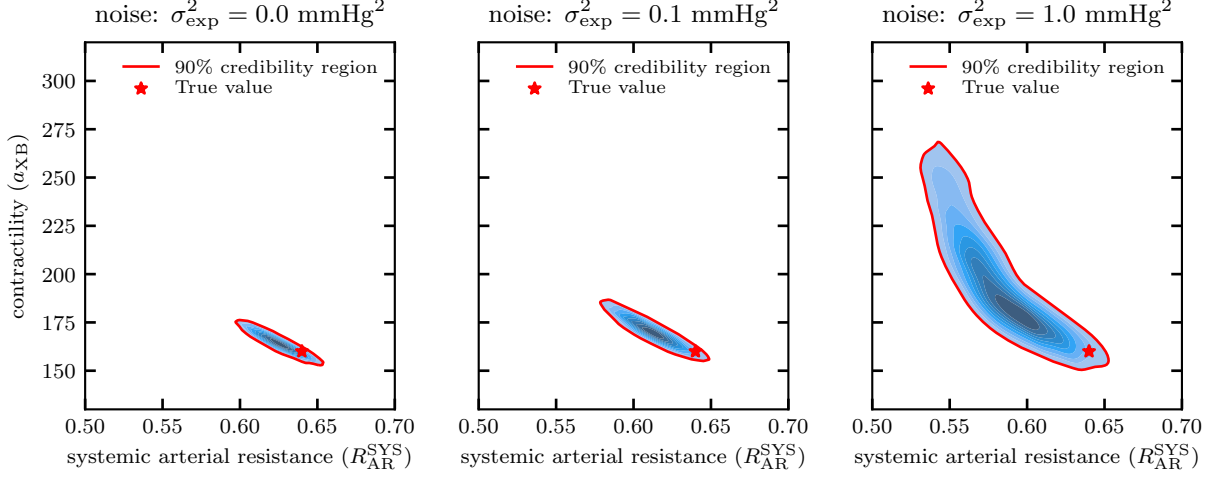





Figure 9: Output of the Bayesian estimation presented in Sec. 3.4. We depict the posterior distribution π_{post} , estimated by means of the MCMC method, for $\sigma_{\text{exp}}^2 = 0$ (left), $\sigma_{\text{exp}}^2 = 0.1 \text{ mmHg}^2$ (middle) and $\sigma_{\text{exp}}^2 = 1 \text{ mmHg}^2$ (right). The red lines show the 90% credibility regions, while the red stars represent the exact value of the unknown parameters a_{XB} and $R_{\text{AR}}^{\text{SYS}}$.

is the region in the parameter space with largest posterior probability such that it covers 90% of π_{post} . We notice that for each value of noise, the credibility region contains the exact value of the parameters (namely $a_{\text{XB}} = 160 \text{ MPa}$ and $R_{\text{AR}}^{\text{SYS}} = 0.64 \text{ mmHg s mL}^{-1}$), represented by a red star. As expected, for larger values of noise, the size of the credibility region increases (that is, the estimate is more uncertain). As a matter of fact, an advantage of Bayesian parameter estimation methods, compared to deterministic methods, is their ability of quantifying the uncertainty associated with the parameters estimate. A further feature of Bayesian methods stands in capturing correlations among the estimated parameters. Indeed, this aspect emerges clearly because of the oblique shape of the credibility regions. This is due to the fact that an increase in a_{XB} or a decrease in $R_{\text{AR}}^{\text{SYS}}$ leads to similar changes in terms of the measured QoIs ($p_{\text{AR},\text{SYS}}^{\min}$ and $p_{\text{AR},\text{SYS}}^{\max}$), thus making their posterior distributions highly correlated.

4 Discussion

In this paper, we develop a machine learning method to build ANN-based ROMs of 3D cardiac electromechanical models. Thanks to the reduced-order differential equations learned through our algorithm, it is possible to approximate the cardiac dynamics in terms of pressure and volume transients with great fidelity and with a huge computational saving. As a matter of fact, once trained, the ANN-based ROM permits to simulate a heartbeat virtually in real time (about one second of numerical simulation for a heartbeat on a standard laptop), when the original electromechanical model requires about four hours per heartbeat on a supercomputer with 32 cores. By taking into account the number of cores, our ANN-based ROM brings the impressive speedup of 460'000x (see Fig. 10). Clearly, a fair evaluation of the computational saving should also consider the time required to generate the training dataset and to train the model. To this aim, we consider two test cases, corresponding to the examples of global sensitivity analysis and Bayesian parameter estimation

<i>model</i>	<i>task</i>	<i>computational platform</i>	<i>computational time</i>
\mathcal{M}_{3D-C}	simulation of a heartbeat	 32 cores supercomputer	4 hours
\mathcal{M}_{ANN-C}	simulation of a heartbeat	 single core standard laptop	1 second
\mathcal{M}_{ANN}	training	 single core standard laptop	18 hours

460'000x speedup

Figure 10: Summary of the computational times associated with FOM and ROM numerical simulations and with training of the ANN-based ROM.

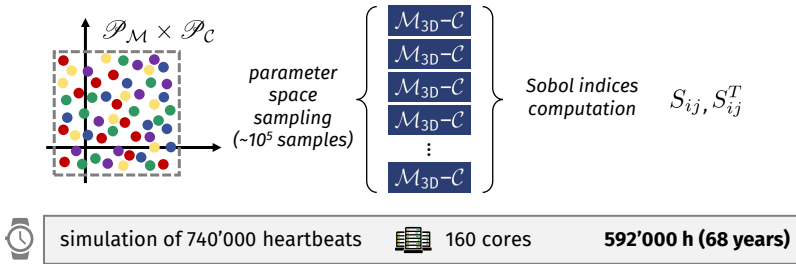
presented in Secs. 2.4 and 2.5, respectively. In both cases, we consider 5 computer nodes with 32 cores each available, and we compare the total computational times using either the \mathcal{M}_{3D-C} or the \mathcal{M}_{ANN-C} model with these computational resources.

The global sensitivity analysis presented in Sec. 2.4 requires the numerical simulation of 74'000 parameter cases. On average, the limit cycle is reached in 10 heartbeats, for a total of 740'000 heartbeats to be simulated. Running this huge amount of numerical simulations with the FOM \mathcal{M}_{3D-C} model would not be feasible, as it would require about 68 years of uninterrupted use of the 5 nodes endowed with 32 cores. On the contrary, by virtue of our machine learning method, we are able to perform a global sensitivity analysis, albeit with a small approximation (see Tabs. 5 and 6) in the numerical results, in 7.5 days (6.7 days to generate the training dataset, 18 hours to train the model and 1 hour and 17 minutes to perform the sensitivity analysis using the \mathcal{M}_{ANN-C} model). Therefore, taking into account the time required to build the ROM, our approach yields a 3'300x speedup (see Fig. 11). Bayesian parameter estimation, on the other hand, requires the numerical simulation of approximately 960'000 heartbeats. In this case, despite using only 20 cores to run the MCMC, we are able to obtain a result in just 6 days and 8 hours (including generation of the training dataset), when with the FOM \mathcal{M}_{3D-C} model it would take more than 87 years. We thus obtain an overall speedup of 5'000x (see Fig. 12). Moreover, we notice that in case we need to execute a Bayesian calibration for different data, it is not necessary to repeat the training of the ANN-based model, but it is sufficient to re-run only the MCMC algorithm, which takes – thanks to our ROM – only 13 hours and 20 minutes. Finally, we remark that the cost required to train the ANN or to perform numerical simulations with the \mathcal{M}_{ANN-C} model does not depend on the specific \mathcal{M}_{3D} model at hand, as it is only based on the generated pressure and volume transients. In particular, it is not expected to raise if the biophysical detail or the number of degrees of freedom of the computational mesh increase.

The ROM proposed in this paper allows, as shown in Sec. 3, to tackle many-query problems, which would otherwise be unaffordable due to their computational cost. Indeed, sensitivity analysis on cardiac models have mainly addressed, at present, single cell models [41, 54, 57]. Sensitivity studies on 3D models of cardiac electromechanics have been mainly limited to simplified geometries [26], or to a few parameters, varying one parameter at a time (that is, *local* sensitivity analysis) [29, 33]. To the best of our knowledge, the only global sensitivity study on a 3D electromechanical model was only recently performed in [30]; however, the huge computational cost associated with the numerical approximation of the forward model dictated the use of an idealized ellipsoidal geometry and of single heartbeat simulations, thus without letting the system reaching a periodic limit cycle. This leads to results with limited accuracy. Furthermore, parameter estimation under uncertainty using 3D models of cardiac electromechanics is hard to attain, due to the computational cost that would be required by Bayesian methods such as MCMC. Indeed, parameters are typically estimated sequentially by using minimization methods [16, 29, 63] or ad hoc developed heuristics [9, 33].

Global sensitivity analysis

Without ANN-based ROM



With ANN-based ROM

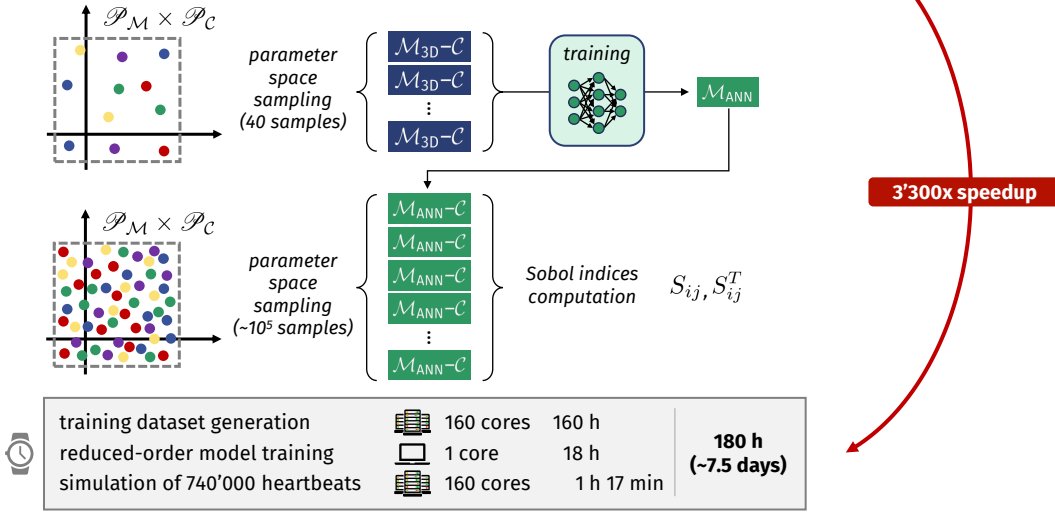
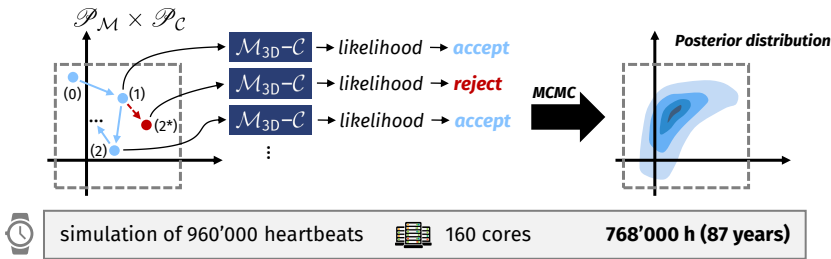


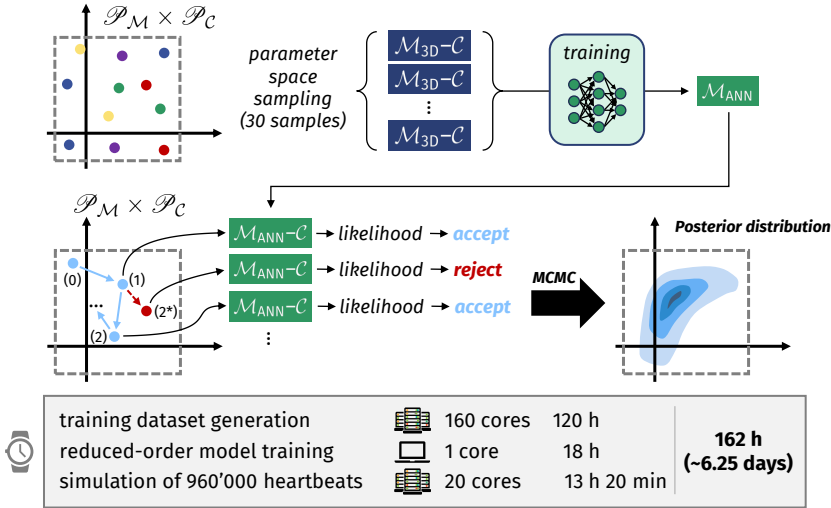
Figure 11: Summary of the computational times associated with global sensitivity analysis.

Bayesian parameter estimation

Without ANN-based ROM



With ANN-based ROM



5'000x speedup

Figure 12: Summary of the computational times associated with Bayesian parameter estimation.

An alternative family of methods, able to provide an estimate of uncertainty, is that of sequential methods of the Kalman filter type (see e.g. [32, 66]), which, however, would require an invasive implementation in the electromechanical model.

Recently, a number of surrogate models of cardiac electromechanics have been built by using machine learning algorithms and they are often called emulators [7, 11, 13, 31]. These emulators are based on a collection of pre-computed numerical simulations obtained by sampling the parameter space, similarly to what has been done in this work. However, the approach behind these emulators is very different from the one we follow. These emulators are in fact maps that fit the parameters-to-QoIs map ($\mathcal{F}: \mathbf{p} \mapsto \mathbf{q}$) in a static manner. On the contrary, with our approach, the ROM makes it possible to perform a real numerical simulation of the cardiac function, since the circulation model \mathcal{C} is kept in its full-order form, while only the computationally demanding part, i.e. the 3D electromechanical model \mathcal{M}_{3D} , is surrogated. This has a number of advantages:

1. The ROM \mathcal{M}_{ANN} is independent of the circulation model \mathcal{C} to which is coupled and it can also be coupled to models different from those used during training, or with different values of its parameters.
2. Unlike the emulators in [7, 11, 13, 31], our ROM allows for time extrapolation, i.e. reliable predictions even over longer time spans than those used during training, as demonstrated by our numerical results. This observation is very important since, while for emulators that fit the parameters-to-QoIs map the simulations contained in the training set must have reached a limit cycle (otherwise the associated QoIs would be meaningless), with our approach it is possible to defer the reaching of the limit cycle to the ROM, during the online phase. This permits to lighten the computational cost associated with the creation of the training set, which typically represents the main component of the total cost (see use cases cited above).
3. The output of the simulations obtained with our ROM is not limited to a list of QoIs. Indeed, it contains the transient of pressures and volumes of the surrogate heart chamber and of the compartments of the circulation model. We notice that the latter is not hidden as a black-box in the parameters-to-QoIs map, as for emulators, but is rather represented explicitly.
4. With our approach, the ROM only needs to learn the variability with respect to $\mathbf{p}_{\mathcal{M}}$ since the circulation model (involving the parameters $\mathbf{p}_{\mathcal{C}}$) remains explicitly represented. Conversely, emulators based on a parameters-to-QoIs map must capture the dependence on both $\mathbf{p}_{\mathcal{M}}$ and $\mathbf{p}_{\mathcal{C}}$ and thus the training set must be large enough to accurately represent their statistical variability. Indeed, we obtain accurate results with only 30–40 samples, a very low number compared to the ones typically required to construct emulators (e.g. 825 samples in [31], 9000 samples in [7]).

A limitation of our ANN-based ROM is that the online phase may be slower than the one of emulators, which do not require to approximate a differential equation but only need to evaluate a function. However, our ANN-based ROM enables real-time simulations and can be readily applied in a number of use cases, such as global sensitivity analysis, parameter estimation and uncertainty quantification. Furthermore, as highlighted above, most of the computational cost is not due to the evaluation of the ROM, but rather to the construction of the training set, which our approach allows to keep very small (30–40 samples in our test cases). Therefore, we conclude that keeping the circulation model in its full-order form and surrogating only the computationally intensive electromechanical model allows for a very favorable trade-off between what is reduced (variability that must be explored during the offline phase) and what is not reduced (variability that must be account for in the online phase).

A different type of emulator is the one we proposed in [49], that permits - similarly to what is done in this paper - to couple a reduced version of the 3D electromechanical model to a circulation model, thus enabling for real-time numerical simulations. However, the emulator of [49] is built for a fixed value of the parameters \mathbf{p}_M , or (through a linear interpolation) for the variability of a single parameter at most. On the other hand, its construction only requires one or two numerical simulations performed through the FOM. Therefore, the emulator of [49] is advantageous when one needs to surrogate the model for a given parametrization (e.g. to quickly converge to a limit cycle, or to perform sensitivity analysis on \mathbf{p}_C only); conversely, when one needs to explore the parametric dependence of an electromechanical model, the approach proposed in this paper turns out to be favorable.

The ROM proposed in this paper features several differences with respect to projection-based ROMs, which have been used in the cardiovascular field as well (see e.g. [5, 38, 39, 45]), or deep learning based approximations of the parameters-to-solution map (see e.g. [17, 18]). The latter families of ROMs indeed provide a richer output than our ROM, as they allow for the approximation of spatially varying outputs of the 3D model, such as transmembrane potential and tissue displacement. However, the online phase of projection-based ROMs is typically more computationally demanding than the one of our ROM [5, 39]. Moreover, unlike our proposed method, projection-based ROMs are intrusive in the equations to be reduced and suitable hyper-reduction techniques should be addressed to handle the nonlinearities of cardiac models. The deep learning based method of [17, 18] features a training phase that is more computationally demanding than the one of the method proposed in this paper, due to the larger size of the outputs. Conversely, in applications in which the knowledge of pressures, volumes and blood fluxes associated with the cardiac chambers are sufficient, our ROM permits to accurately approximate the outputs of the FOM at a very reduced computational cost and in a non-intrusive manner (only pressure and volume recordings are required from the FOM).

We remark that the proposed method is limited to electromechanical models that feature a periodic behavior. The ROM is indeed periodic by construction, due to the presence on the sine and cosine terms, as highlighted in Eq. (3). Therefore, the ROM is not suitable, e.g., to simulate an irregular electrophysiological behavior, such as ventricular arrhythmias, and other electric dysfunctions.

5 Conclusions

We presented a machine learning method to build ANN-based ROMs of cardiac electromechanical models. Indeed, on the basis of pressure and volume transients generated with the FOM, we are capable of learning a system of differential equations that approximates the dynamics of the cardiac chamber to be surrogated. This system of differential equations, linking pressure and volume of a cardiac chamber, is coupled with lumped-parameter models of cardiac hemodynamics, thus allowing for the simulation of the cardiac function at a dramatically reduced computational cost with respect to the original FOM. As a matter of fact, our ROM permits to perform numerical simulations virtually in real-time. Moreover, thanks to its non-intrusive nature, the proposed algorithm can be easily applied to other electromechanical models besides the one considered in this work.

We presented two test cases in which we employ the ANN-based ROM. We carried out a global sensitivity analysis to assess the influence of the parameters of the electromechanical and hemodynamic models on a list of outputs of clinical interest. Then, we performed a Bayesian estimation of a couple of parameters (belonging to the electromechanical and hemodynamic models, respectively), starting from the noisy measurement of a couple of scalar quantities (namely maximum and minimum arterial pressure). In both the cases, performing through the FOM the large number of

numerical simulations needed would not have been feasible, due to their high computational cost (it would in fact have taken tens of years on a supercomputer computational platform). Replacing the FOM with its ANN-based surrogate allowed us to obtain an approximate solution in a few hours of computation. By taking into account that the generation of the numerical simulations contained in the training set required less than 7 days on the same computational platform, our ANN-based ROM allowed us to reduce the total computational time by more than 3'000 times.

Acknowledgements

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 740132, iHEART - An Integrated Heart Model for the simulation of the cardiac function, P.I. Prof. A. Quarteroni).



Appendices

A Numerical settings

In this appendix we report the main numerical approaches that we employ to develop and test the ROM of cardiac electromechanics presented in this paper.

A.1 Electromechanical model

We use the same choices in terms of space and time discretization for all the numerical simulations of this paper. Specifically, we employ a finer mesh for cardiac electrophysiology, which consists of 258'415 DOFs and 240'864 elements (with an average mesh size of $h_{\text{mean}} = 1.7$ mm), and a coarser one for cardiac mechanics, which is made by 35'725 DOFs and 30'108 elements ($h_{\text{mean}} = 3.4$ mm). To advance the electrophysiological variables we use a timestep length of $\Delta t = 100 \mu\text{s}$ and a five times larger timestep to advance the mechanical variables. The numerical simulations were run by using one cluster node endowed with 32 cores (four Intel Xeon E5-4610 v2, 2.3 GHz) which is available at MOX, Dipartimento di Matematica.

A.2 Convergence to the limit cycle

To determine when a simulation reached a limit cycle (i.e. a periodic solution), we employ a criterion based on the increment between successive cycles. Specifically, we consider the limit cycle to be reached when maximum difference between two consecutive cycles in the pressures and volumes of all four chambers is less than 0.8 mmHg and 0.8 mL, respectively. This criterion is typically satisfied in 5 to 15 cycles. In any case, we always perform a minimum of 5 cycles.

A.3 Training algorithm

To generate the training datasets, we employ a Monte Carlo based sampling of the parameter space $\mathcal{P}_{\mathcal{M}} \times \mathcal{P}_{\mathcal{C}}$. As mentioned, we consider a subset of the parameters $\mathbf{p}_{\mathcal{C}}$, made by

$$V_{\text{heart}}^{\text{tot}}, E_{\text{LA}}^{\text{pass}}, t_{\text{L}}^{\text{av}}, E_{\text{RA}}^{\text{pass}}, t_{\text{R}}^{\text{av}}, E_{\text{RV}}^{\text{act}}, E_{\text{RV}}^{\text{pass}}, R_{\text{AR}}^{\text{SYS}}, R_{\text{VEN}}^{\text{SYS}}.$$

To define the sampling spaces $\mathcal{P}_{\mathcal{M}}$ and $\mathcal{P}_{\mathcal{C}}$, we consider 50% to 200% ranges of the baseline values reported in Tabs. 1 and 2 for $V_{\text{heart}}^{\text{tot}}, E_{\text{LA}}^{\text{pass}}, a_{\text{XB}}, E_{\text{RA}}^{\text{pass}}, E_{\text{RV}}^{\text{act}}, E_{\text{RV}}^{\text{pass}}, R_{\text{AR}}^{\text{SYS}}$ and $R_{\text{VEN}}^{\text{SYS}}$, 50% to 150% for σ_{f} and C . We vary $\alpha \in (40^\circ, 80^\circ)$ and $t_{\text{L}}^{\text{av}}, t_{\text{R}}^{\text{av}} \in (0.08, 0.24)$ s.

To train the ANN-based model according to Eq. (4), we employ as mentioned the algorithm that we proposed in [47]. More precisely, we approximate the trained ODE with a time step size of $\Delta t = 5$ ms and the objective functional with a twice as wide time step. To train the ANN, we employ 2000 iterations of the Levenberg-Marquardt algorithm.

A.4 Hyperparameters tuning

To tune the hyperparameters, we adopt a k -fold cross-validation procedure as described in Sec. 2.3. Specifically, we consider a given hyperparameters setting and we partition the training dataset in $k = 5$ non-overlapping subsets. Then, by retaining a single subset as validation data, we train, on the remaining $k - 1$ subsets, three different models (with three different random initializations of the ANN weights) and we keep the model attaining the lowest validation error. This procedure is repeated k times, once for each subset. Finally, we average the errors over the k trained models, and we compare the resulting average errors for different hyperparameter settings. More precisely, to compare the performances of the models obtained with different hyperparameter settings, we consider the validation errors associated with the QoIs reported in Tab. 3, in terms of relative errors. Moreover, we assess the generalization skills of the trained models by computing the ratio between validation errors and training errors. Indeed, a ratio that is much larger than one indicates overfitting. By following these criteria, we get to the final configurations of Tab. 4.

A.5 Global sensitivity analysis

To perform global sensitivity analysis, we consider all parameters $\mathbf{p}_{\mathcal{M}}$ and $\mathbf{p}_{\mathcal{C}}$, which are reported in Tabs. 1 and 2, respectively. To define the sampling spaces $\mathcal{P}_{\mathcal{M}}$ and $\mathcal{P}_{\mathcal{C}}$, we consider 80% to 120% ranges of the baseline values in Tabs. 1 and 2, except for $V_{\text{heart}}^{\text{tot}} \in (193, 593)$ mL and for $t_{\text{L}}^{\text{av}}, t_{\text{R}}^{\text{av}} \in (0.04, 0.28)$ s.

We employ the Saltelli's method to sample the parametric space $\mathcal{P}_{\mathcal{M}} \times \mathcal{P}_{\mathcal{C}}$. A naive sampling of the space would require an exponential increase with respect to the number of parameters to guarantee a prescribed accuracy; conversely, Saltelli's method leads to a linear increase. More precisely, the number of samples required is $N(2D + 2)$, where D is the number of parameters and N is a user defined setting. In this paper we set $N = 1000$, for a total of 74'000 samples, that guarantees small confidence intervals around the first-order Sobol indices and the total-effect Sobol indices reported in Figs. 7 and 8, respectively. We use one cluster node endowed with 20 cores (four Intel Xeon E5-2640 v4, 2.4 GHz), which is available at MOX, Dipartimento di Matematica, to run global sensitivity analysis.

A.6 Bayesian parameter estimation

We perform Bayesian parameter estimation by means of the MCMC method. We consider 500 samples per chain, a jump period of 10 samples and a burn-in equal to 1000, for a total number of

20 chains. Indeed, we use one cluster node endowed with 20 cores (four Intel Xeon E5-2640 v4, 2.4 GHz), which is available at MOX, Dipartimento di Matematica, where each core manages a single chain and then all final results are collected together at the end of the simulation.

References

- [1] *ASME Standard V&V 40 - Assessing credibility of computational modeling through verification and validation: Application to medical devices*. 2018.
- [2] R. C. Aster, B. Borchers, and C. H. Thurber. *Parameter estimation and inverse problems*. Elsevier, 2018.
- [3] C. Augustin, M. Gsell, E. Karabelas, and G. Plank. “Physiologically valid 3D-0D closed loop model of the heart and circulation—Modeling the acute response to altered loading and contractility”. In: *arXiv preprint arXiv:2009.08802* (2020).
- [4] J. Bayer, R. Blake, G. Plank, and N. Trayanova. “A novel rule-based algorithm for assigning myocardial fiber orientation to computational heart models”. In: *Annals of Biomedical Engineering* 40.10 (2012), pp. 2243–2254.
- [5] D. Bonomi, A. Manzoni, and A. Quarteroni. “A matrix DEIM technique for model reduction of nonlinear parametrized problems in cardiac mechanics”. In: *Computer Methods in Applied Mechanics and Engineering* 324 (2017), pp. 300–326.
- [6] S. Brooks. “Markov chain Monte Carlo method and its application”. In: *Journal of the royal statistical society: series D (the Statistician)* 47.1 (1998), pp. 69–100.
- [7] L. Cai, L. Ren, Y. Wang, W. Xie, G. Zhu, and H. Gao. “Surrogate models based on machine learning methods for parameter estimation of left ventricular myocardium”. In: *Royal Society Open Science* 8.1 (2021), p. 201121.
- [8] P. Colli Franzone, L. Pavarino, and S. Scacchi. *Mathematical cardiac electrophysiology*. Vol. 13. Springer, 2014.
- [9] C. M. Costa, E. Hoetzl, B. M. Rocha, A. J. Prassl, and G. Plank. “Automatic parameterization strategy for cardiac electrophysiology simulations”. In: *Computing in Cardiology 2013*. IEEE, 2013, pp. 373–376.
- [10] S. Coveney, C. Corrado, J. E. Oakley, R. D. Wilkinson, S. A. Niederer, and R. H. Clayton. “Bayesian Calibration of Electrophysiology Models using Restitution Curve Emulators”. In: *Frontiers in Physiology* (2021), p. 1120.
- [11] Y. Dabiri, A. Van der Velden, K. L. Sack, J. S. Choy, G. S. Kassab, and J. M. Guccione. “Prediction of left ventricular mechanics using machine learning”. In: *Frontiers in physics* 7 (2019), p. 117.
- [12] L. Dedè, A. Quarteroni, and F. Regazzoni. “Mathematical and numerical models for the cardiac electromechanical function”. In: *Atti della Accademia Nazionale dei Lincei, Classe di Scienze Fisiche, Matematiche e Naturali. Rendiconti Lincei - Matematica e Applicazioni* 32.2 (2021), pp. 233–272.
- [13] P. Di Achille, A. Harouni, S. Khamzin, O. Solovyova, J. J. Rice, and V. Gurev. “Gaussian process regressions for inverse problems and parameter searches in models of ventricular mechanics”. In: *Frontiers in physiology* 9 (2018), p. 1002.
- [14] *FDA Recognized Consensus Standards*. accessed: June 12th, 2021.

- [15] M. Fink, S. Niederer, E. Cherry, F. Fenton, J. Koivumäki, G. Seemann, R. Thul, H. Zhang, F. Sachse, D. Beard, E. Crampin, and N. Smith. “Cardiac cell modelling: observations from the heart of the cardiac physiome project”. In: *Progress in Biophysics and Molecular Biology* 104.1 (2011), pp. 2–21.
- [16] H. Finsberg, G. Balaban, S. Ross, T. F. Håland, H. H. Odland, J. Sundnes, and S. Wall. “Estimating cardiac contraction through high resolution data assimilation of a personalized mechanical model”. In: *Journal of computational science* 24 (2018), pp. 85–90.
- [17] S. Fresca, A. Manzoni, L. Dedè, and A. Quarteroni. “Deep learning-based reduced order models in cardiac electrophysiology”. In: *PloS one* 15.10 (2020), e0239416.
- [18] S. Fresca, A. Manzoni, L. Dedè, and A. Quarteroni. “POD-Enhanced Deep Learning-Based Reduced Order Models for the Real-Time Simulation of Cardiac Electrophysiology in the Left Atrium”. In: *Frontiers in Physiology* 12 (2021), p. 1431.
- [19] A. Gerbi, L. Dedè, and A. Quarteroni. “A monolithic algorithm for the simulation of cardiac electromechanics in the human left ventricle”. In: *Mathematics in Engineering* 1 (2018), pp. 1–37.
- [20] J. Herman and W. Usher. “SALib: An open-source Python library for Sensitivity Analysis”. In: *The Journal of Open Source Software* 2.9 (2017).
- [21] M. Hirschvogel, M. Bassilious, L. Jagschies, S. Wildhirt, and M. Gee. “A monolithic 3D-0D coupled closed-loop model of the heart and the vascular system: experiment-based parameter estimation for patient-specific cardiac mechanics”. In: *International journal for numerical methods in biomedical engineering* 33 (2017), e2842.
- [22] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley. “Stochastic variational inference”. In: *Journal of Machine Learning Research* 14.5 (2013).
- [23] J. H. Holland. “Genetic algorithms”. In: *Scientific american* 267.1 (1992), pp. 66–73.
- [24] T. Homma and A. Saltelli. “Importance measures in global sensitivity analysis of nonlinear models”. In: *Reliability Engineering & System Safety* 52.1 (1996), pp. 1–17.
- [25] K. T. Hu and G. E. Orient. “The 2014 sandia verification and validation challenge: Problem statement”. In: *Journal of Verification, Validation and Uncertainty Quantification* 1.1 (2016).
- [26] D. Hurtado, S. Castro, and P. Madrid. “Uncertainty quantification of 2 models of cardiac electromechanics”. In: *International Journal for Numerical Methods in Biomedical Engineering* 33.12 (2017), e2894.
- [27] Z. M. G. Inc. *Zygote Solid 3D heart Generation II Development Report*. Technical Report. Zygote Media Group Inc., 2014.
- [28] S. Kucherenko, S. Tarantola, and P. Annoni. “Estimation of global sensitivity indices for models with dependent variables”. In: *Computer physics communications* 183.4 (2012), pp. 937–946.
- [29] S. Land, S. A. Niederer, J. M. Aronsen, E. K. Espe, L. Zhang, W. E. Louch, I. Sjaastad, O. M. Sejersted, and N. P. Smith. “An analysis of deformation-dependent electromechanical coupling in the mouse heart”. In: *The Journal of physiology* 590.18 (2012), pp. 4553–4569.
- [30] F. Levrero-Florencio, F. Margara, E. Zacur, and et al. “Sensitivity analysis of a strongly-coupled human-based electromechanical cardiac model: Effect of mechanical parameters on physiologically relevant biomarkers”. In: *Computer methods in applied mechanics and engineering* 361 (2020), p. 112762.

- [31] S. Longobardi, A. Lewalle, S. Coveney, I. Sjaastad, E. K. S. Espe, W. E. Louch, C. J. Musante, A. Sher, and S. A. Niederer. “Predicting left ventricular contractile function via Gaussian process emulation in aortic-banded rats”. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 378.2173 (2020), p. 20190334.
- [32] S. Marchesseau, H. Delingette, M. Sermesant, R. Cabrera-Lozoya, C. Tobon-Gomez, P. Moireau, R. Figueras i Ventura, K. Lekadir, A. Hernandez, M. Garreau, E. Donal, C. Leclercq, S. Duckett, K. Rhode, C. Rinaldi, A. Frangi, R. Razavi, D. Chapelle, and N. Ayache. “Personalization of a cardiac electromechanical model using reduced order unscented Kalman filtering from regional volumes”. In: *Medical Image Analysis* 17.7 (2013), pp. 816–829.
- [33] L. Marx, M. A. F. Gsell, A. Rund, F. Caforio, A. J. Prassl, G. Toth-Gayor, T. Kuehne, C. M. Augustin, and G. Plank. “Personalization of electro-mechanical models of the pressure-overloaded left ventricle: fitting of Windkessel-type afterload models”. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 378.2173 (2020), p. 20190342.
- [34] M. D. Morris. “Factorial sampling plans for preliminary computational experiments”. In: *Technometrics* 33.2 (1991), pp. 161–174.
- [35] J. Nocedal and S. Wright. *Numerical optimization*. second. New York, NY, USA: Springer Science & Business Media, 2006.
- [36] W. L. Oberkampf, T. G. Trucano, and C. Hirsch. “Verification, validation, and predictive capability in computational engineering and physics”. In: *Appl. Mech. Rev.* 57.5 (2004), pp. 345–384.
- [37] A. Olivier, D. Giovanis, B. Aakash, M. Chauhan, L. Vandanapu, and M. Shields. “UQpy: A general purpose Python package and development environment for uncertainty quantification”. In: *Journal of Computational Science* 47 (2020), p. 101204.
- [38] S. Pagani. “Reduced-order models for inverse problems and uncertainty quantification in cardiac electrophysiology”. PhD thesis. Politecnico di Milano, 2017.
- [39] S. Pagani and A. Manzoni. “Enabling forward uncertainty quantification and sensitivity analysis in cardiac electrophysiology by reduced order modeling and machine learning”. In: *International Journal for Numerical Methods in Biomedical Engineering* (2021), e3450.
- [40] P. Pathmanathan, S. Chapman, D. Gavaghan, and J. Whiteley. “Cardiac electromechanics: the effect of contraction model on the mathematical problem and accuracy of the numerical scheme”. In: *The Quarterly Journal of Mechanics & Applied Mathematics* 63.3 (2010), pp. 375–399.
- [41] P. Pathmanathan, J. M. Cordeiro, and R. A. Gray. “Comprehensive uncertainty quantification and sensitivity analysis for cardiac action potential models”. In: *Frontiers in physiology* 10 (2019), p. 721.
- [42] R. Piersanti, P. C. Africa, M. Fedele, C. Vergara, L. Dede’, A. F. Corno, and A. Quarteroni. “Modeling cardiac muscle fibers in ventricular and atrial electrophysiology simulations”. In: *Computer Methods in Applied Mechanics and Engineering* 373 (2021), p. 113468.
- [43] E. Plischke, E. Borgonovo, and C. L. Smith. “Global sensitivity measures from given data”. In: *European Journal of Operational Research* 226.3 (2013), pp. 536–550.
- [44] A. Quarteroni, L. Dedè, A. Manzoni, and C. Vergara. *Mathematical modelling of the human cardiovascular system: data, numerical approximation, clinical applications*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2019.

- [45] A. Quarteroni, A. Manzoni, and F. Negri. *Reduced Basis Methods for Partial Differential Equations. An Introduction*. Vol. 92. Springer, 2016.
- [46] F. Regazzoni, L. Dedè, and A. Quarteroni. “Biophysically detailed mathematical models of multiscale cardiac active mechanics”. In: *PLoS computational biology* 16 (2020), e1008294.
- [47] F. Regazzoni, L. Dedè, and A. Quarteroni. “Machine learning for fast and reliable solution of time-dependent differential equations”. In: *Journal of Computational Physics* 397 (2019), p. 108852.
- [48] F. Regazzoni, L. Dedè, and A. Quarteroni. “Machine learning of multiscale active force generation models for the efficient simulation of cardiac electromechanics”. In: *Computer Methods in Applied Mechanics and Engineering* 370 (2020), p. 113268.
- [49] F. Regazzoni and A. Quarteroni. “Accelerating the convergence to a limit cycle in 3D cardiac electromechanical simulations through a data-driven 0D emulator”. In: *Computers in Biology and Medicine* 135 (2021), p. 104641.
- [50] F. Regazzoni, M. Salvador, P. C. Africa, M. Fedele, L. Dede’, and A. Quarteroni. “A cardiac electromechanics model coupled with a lumped parameters model for closed-loop blood circulation. Part I: model derivation”. In: *arXiv preprint arXiv:2011.15040* (2020).
- [51] F. Regazzoni, M. Salvador, P. C. Africa, M. Fedele, L. Dede’, and A. Quarteroni. “A cardiac electromechanics model coupled with a lumped parameters model for closed-loop blood circulation. Part II: numerical approximation”. In: *arXiv preprint arXiv:2011.15051* (2020).
- [52] F. Regazzoni and A. Quarteroni. “An oscillation-free fully partitioned scheme for the numerical modeling of cardiac active mechanics”. In: *Computer Methods in Applied Mechanics and Engineering* 373 (2021), p. 113506.
- [53] A. Saltelli. “Making best use of model evaluations to compute sensitivity indices”. In: *Computer physics communications* 145.2 (2002), pp. 280–297.
- [54] A. A. Sher, K. Wang, A. Wathen, P. J. Maybank, G. R. Mirams, D. Abramson, D. Noble, and D. J. Gavaghan. “A local sensitivity analysis method for developing biological models with identifiable parameters: Application to cardiac ionic channel modelling”. In: *Future Generation Computer Systems* 29.2 (2013), pp. 591–598.
- [55] N. Smith, D. Nickerson, E. Crampin, and P. Hunter. “Multiscale computational modelling of the heart”. In: *Acta Numerica* 13 (2004), pp. 371–431.
- [56] I. M. Sobol’. “On sensitivity estimation for nonlinear mathematical models”. In: *Matematicheskoe modelirovanie* 2.1 (1990), pp. 112–118.
- [57] K. Tøndel, S. Land, S. A. Niederer, and N. P. Smith. “Quantifying inter-species differences in contractile function through biophysical modelling”. In: *The Journal of Physiology* 593.5 (2015), pp. 1083–1111.
- [58] G. J. Tortora and B. H. Derrickson. *Principles of anatomy and physiology*. John Wiley & Sons, 2008.
- [59] N. A. Trayanova. “Whole-heart modeling applications to cardiac electrophysiology and electromechanics”. In: *Circulation Research* 108 (2011), pp. 113–128.
- [60] K. ten Tusscher and A. Panfilov. “Alternans and spiral breakup in a human ventricular tissue model”. In: *American Journal of Physiology-Heart and Circulatory Physiology* 291 (2006), H1088–H1100.

- [61] T. P. Usyk, I. J. LeGrice, and A. D. McCulloch. “Computational model of three-dimensional cardiac electromechanics”. In: *Computing and Visualization in Science* 4.4 (2002), pp. 249–257.
- [62] E. J. Vigmond, C. Clements, D. M. McQueen, and C. S. Peskin. “Effect of bundle branch block on cardiac output: a whole heart simulation study”. In: *Progress in biophysics and molecular biology* 97.2-3 (2008), pp. 520–542.
- [63] V. Y. Wang, H. Lam, D. B. Ennis, B. R. Cowan, A. A. Young, and M. P. Nash. “Modelling passive diastolic mechanics with quantitative MRI of cardiac structure and function”. In: *Medical image analysis* 13.5 (2009), pp. 773–784.
- [64] T. Washio, K. Yoneda, J. Okada, T. Kariya, S. Sugiura, and T. Hisada. “Ventricular fiber optimization utilizing the branching structure”. In: *International Journal for Numerical Methods in Biomedical Engineering* (2015), e02753.
- [65] N. Westerhof, J.-W. Lankhaar, and B. E. Westerhof. “The arterial Windkessel”. In: *Medical & Biological Engineering & Computing* 47.2 (2009), pp. 131–141.
- [66] J. Xi, P. Lamata, J. Lee, P. Moireau, D. Chapelle, and N. Smith. “Myocardial transversely isotropic material parameter estimation from in-silico measurements based on a reduced-order unscented Kalman filter”. In: *Journal of the mechanical behavior of biomedical materials* 4.7 (2011), pp. 1090–1102.
- [67] C. Xu and G. Z. Gertner. “Uncertainty and sensitivity analysis for models with correlated parameters”. In: *Reliability Engineering & System Safety* 93.10 (2008), pp. 1563–1573.

MOX Technical Reports, last issues

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

- 65/2021** Mazzieri, I.; Muhr, M.; Stupazzini, M.; Wohlmuth, B.
Elasto-acoustic modelling and simulation for the seismic response of structures: The case of the Tahtali dam in the 2020 Izmir earthquake
- 67/2021** Salvador, M.; Regazzoni, F.; Pagani, S.; Dede', L.; Trayanova, N.; Quarteroni, A.
The role of mechano-electric feedbacks and hemodynamic coupling in scar-related ventricular tachycardia
- 66/2021** Antonietti, P.F.; Botti, M.; Mazzieri, I.
On mathematical and numerical modelling of multiphysics wave propagation with polygonal Discontinuous Galerkin methods
- 64/2021** Clarotto, L.; Allard, D.; Menafoglio, A.
A new class of alpha-transformations for the spatial analysis of Compositional Data
- 63/2021** Rosafalco, L.; Torzoni, M.; Manzoni, A.; Mariani, S.; Corigliano, A.
Online structural health monitoring by model order reduction and deep learning algorithms
- 62/2021** Lupo Pasini, M.; Burcul, M.; Reeve, S.; Eisenbach, M.; Perotto, S.
Fast and accurate predictions of total energy for solid solution alloys with graph convolutional neural networks
- 60/2021** Rosafalco, L.; Manzoni, A.; Mariani, S.; Corigliano, A.
Fully convolutional networks for structural health monitoring through multivariate time series classification
- 61/2021** Buchwald, S.; Ciaramella, G.; Salomon, J.; Sugny, D.
A greedy reconstruction algorithm for the identification of spin distribution
- 59/2021** Stella, S.; Regazzoni, F.; Vergara, C.; Dede', L.; Quarteroni, A.
A fast cardiac electromechanics model coupling the Eikonal and the nonlinear mechanics equations
- 58/2021** Tassi, T.; Zingaro, A.; Dede', L.
Enhancing numerical stabilization methods for advection dominated differential problems by Machine Learning algorithms