Computer Methods and Programs in Biomedicine 257 (2024) 108466

Contents lists available at ScienceDirect

Computer Methods and Programs in Biomedicine

journal homepage: https://www.sciencedirect.com/journal/computer-methods-and-programs-in-biomedicine

Data-driven reduced order surrogate modeling for coronary in-stent restenosis

Jianye Shi [a,*], Kiran Manjunatha [a], Felix Vogt [b], Stefanie Reese [a]

[a] *Institute of Applied Mechanics, RWTH Aachen University, Germany*
[b] *Department of Cardiology, Vascular Medicine and Intensive Care, RWTH Aachen University, Germany*

ARTICLE INFO

Keywords:
In-stent restenosis (ISR)
Drug-eluting stents (DES)
Surrogate model
3D convolutional neural networks
Autoencoder
Non-intrusive model order reduction

ABSTRACT

Background: The intricate process of coronary in-stent restenosis (ISR) involves the interplay between different mediators, including platelet-derived growth factor, transforming growth factor-β, extracellular matrix, smooth muscle cells, endothelial cells, and drug elution from the stent. Modeling such complex multiphysics phenomena demands extensive computational resources and time.

Methods: This paper proposes a novel non-intrusive data-driven reduced order modeling approach for the underlying multiphysics time-dependent parametrized problem. In the offline phase, a 3D convolutional autoencoder, comprising an encoder and decoder, is trained to achieve dimensionality reduction. The encoder condenses the full-order solution into a lower-dimensional latent space, while the decoder facilitates the reconstruction of the full solution from the latent space. To deal with the 5D input datasets (3D geometry + time series + multiple output channels), two ingredients are explored. The first approach incorporates time as an additional parameter and applies 3D convolution on individual time steps, encoding a distinct latent variable for each parameter instance within each time step. The second approach reshapes the 3D geometry into a 2D plane along a less interactive axis and stacks all time steps in the third direction for each parameter instance. This rearrangement generates a larger and complete dataset for one parameter instance, resulting in a singular latent variable across the entire discrete time-series. In both approaches, the multiple outputs are considered automatically in the convolutions. Moreover, Gaussian process regression is applied to establish correlations between the latent variable and the input parameter.

Results: The constitutive model reveals a significant acceleration in neointimal growth between $30-60$ days post percutaneous coronary intervention (PCI). The surrogate models applying both approaches exhibit high accuracy in pointwise error, with the first approach showcasing smaller errors across the entire evaluation period for all outputs. The parameter study on drug dosage against ISR rates provides noteworthy insights of neointimal growth, where the nonlinear dependence of ISR rates on the peak drug flux exhibits intriguing periodic patterns. Applying the trained model, the rate of ISR is effectively evaluated, and the optimal parameter range for drug dosage is identified.

Conclusion: The demonstrated non-intrusive reduced order surrogate model proves to be a powerful tool for predicting ISR outcomes. Moreover, the proposed method lays the foundation for real-time simulations and optimization of PCI parameters.

1. Introduction

Modeling coronary in-stent restenosis (ISR) not only facilitates understanding of the intricate interplay between different mediators but also serves as a predictive tool for estimating the outcomes of percutaneous coronary intervention (PCI) [1–3]. For instance, real-time simulation of the ISR process during PCI enhances the practical utility and aids in optimizing parameters for effective decision-making, including stent geometry, degrees of stent expansion, and drug-elution rate. Despite rapid developments in hardware and numerics, achieving real-time high-fidelity solutions with both high temporal and high spatial resolutions for the multiphysical ISR process remains computationally challenging.

Reduced order modeling (ROM) is a useful alternative to the comprehensive full-order model (FOM). The central objective of ROM is to replace the FOM with a significantly smaller dimension, reducing the

---

* Corresponding author.
E-mail address: jianye.shi@ifam.rwth-aachen.de (J. Shi).

https://doi.org/10.1016/j.cmpb.2024.108466
Received 31 March 2024; Received in revised form 17 September 2024; Accepted 15 October 2024

computational workload while maintaining a controlled compromise in accuracy [4]. A well-established and widely-used ROM approach is rooted in projection-based reduced basis (RB) methods [5]. In this methodology class, a low-dimensional approximation of the solution manifold (reduced subspace) is spanned by a linear combination of bases. The FOM is then projected into this low-dimensional reduced subspace to obtain the RB model [6]. Proper orthogonal decomposition (POD) [7] stands out as one of the most popular techniques for obtaining the optimal RB by decomposing the snapshot matrix based on the singular value decomposition (SVD).

Despite their proven effectiveness and robustness for linear problems, projection-based RB methods generally do not offer significant computational advantages for complex nonlinear problems that have a non-affine dependence on the parameters [4,8,9], since the computation of the projection coefficients depends on the dimension of the FOM. In this context, hyper-reduction strategies might be used to recover an affine expansion of the nonlinear differential operator, among them the empirical interpolation method (EIM) [10], the discrete empirical interpolation method (DEIM) [7,11,12], subspace-angle interpolation [13,14], missing point estimation [15], Gauss–Newton with approximated tensors (GNAT) [16], and energy conserving sampling and weighting (ECSW) [17,18]. However, these methods are generally intrusive in nature, where access to equations or solvers of the original FOM is needed and the generalization to nonlinear problems is not straightforward.

Recently, data-driven non-intrusive ROM for nonlinear time-dependent problems has received considerable attention in the model order reduction (MOR) community [4,8,19–32]. These non-intrusive approaches have distinct advantages: they do not require access to the equations governing the original FOM and can handle different types of partial differential equations (PDEs) that define the underlying mechanisms of different systems. The efficiency and accuracy of non-intrusive ROM have been demonstrated in various engineering fields, including fluid dynamics [4,6,8,26–28,33,34], fluid–structure interaction [19,35,36], structural dynamics [29,37,38], continuum constitutive modeling [20], porous media [39], chemically reacting flow [9], computational biomechanics of the musculoskeletal system [40], and cardiac electrophysiology [41,42]. However, despite extensive research in the aforementioned fields, there is currently no predictive non-intrusive reduced order surrogate models addressing the pathophysiology responsible for coronary ISR within the research community. It is urgent to address this deficiency.

To address the temporal evolution of datasets, various strategies have been employed such as recurrent neural networks (RNNs). For instance, in the work of Eivazi et al. [43], autoencoders combined with long short-term memory (LSTM) networks were successfully applied for nonlinear MOR in unsteady flow simulations. Althoug LSTM is powerful to capture long-term dependencies in sequential data, convolutional networks are applied in our work, as they offer advantages in handling high-dimensional spatial data more efficiently, particularly when working with 3D geometries, while also being less computationally expensive. Additionally, convolutional autoencoders are well-suited for extracting spatial features and are more flexible in managing non-sequential data.

This paper aims to develop a data-driven non-intrusive reduced order surrogate model for coronary ISR using 3D convolutional neural networks. A system of coupled advection–reaction–diffusion type partial differential equations (PDEs) is formulated to trace the pathophysiology and the evolution of the influential factors associated with ISR, including platelet-derived growth factor (PDGF), transforming growth factor-β (TGF-β), extracellular matrix (ECM), density of smooth muscle cells (SMC), endothelial cells (EC), and drug concentration. The growth of neointima and the mechanical response of the arterial wall are constitutively described using continuum mechanics within framework of large deformation. The final system is characterized by a total of nine degrees of freedom (three displacements, five biological mediators and drug concentration). High-fidelity datasets are generated through the finite element method, introducing variability in the parameter associated with the drug-releasing rate.

Two approaches are explored to deal with the 5D input dataset (3D geometry + 1D time series + multiple output channels), with comparison of their efficiency and accuracy. The first approach involves treating time as an additional variable, applying 3D convolution to each time step. This implies encoding one latent variable for each input parameter at each time step. The second approach reshapes the 3D geometry into 2D along one less-interactive axis, stacking the time-series in the third direction to form a larger input dataset. This indicates that for one input parameter, only one latent variable is needed across all discrete time-series. The necessity of these two approaches lies mainly in their ability to handle different aspects of the problem's temporal complexity. The first approach is more flexible in dealing with significant temporal changes, while the second approach is more efficient in capturing the overall temporal trend. To establish correlations between the input parameters and the latent space, Gaussian process regression (GPR) is applied for both approaches. While other (machine learning) methods like FNNs or LSTMs could be used to map latent variables to input parameters, GPR offers some key advantages. GPR helps interpret the latent space by establishing a smooth, probabilistic relationship between latent variables and input parameters, allowing for accurate predictions on new, unseen data. Additionally, GPR provides uncertainty quantification, making it a more interpretable and reliable model, especially when dealing with complex, non-linear relationships.

The remainder of this paper is organized as follows: in Section 2.1, generation of the high-fidelity datasets used in this study is introduced, beginning with an introduction of the pathophysiology of in-stent restenosis. We then delve into the evolution of biological mediators, i.e., PDGF, TGF-β, ECM, SMCs, EC, and drug concentration, followed by constitutive description of growth, choice of material parameters, model setup and initial–boundary conditions. Section 2.2 provides an overview of the deep learning techniques employed in this study, including the network architecture and training process of the underlying data-driven non-intrusive MOR using 3D convolutional neural networks. We present in Section 3 the results and discuss in Section 4 the prediction capability of the surrogate model. Finally, Section 5 concludes the paper by summarizing the findings and outlining limitations and potential future research related to this study.

2. Methods

2.1. In-stent restenosis and modeling

This section covers the pathophysiology of ISR, coupled PDEs governing biological mediators and mechanical response, growth description from a continuum mechanics standpoint, and a summary of the numerical model, including geometry, material parameters, and initial/boundary conditions.

2.1.1. Pathophysiology of in-stent restenosis

The pathophysiology of ISR is characterized by the rapid renarrowing of the artery due to excessive neointimal tissue growth within the stented segment. This contrasts with the virgin stenosis caused by long-term progression of atherosclerotic disease [44]. This short-term progression occurs in weeks to months following stent implantation and is linked to vascular injury sustained during PCI. Stent abrasion during PCI leads to denudation of the endothelial monolayer on arterial walls, while under-dilated stents can induce blood flow vortices around struts, causing oscillatory wall shear stress and endothelial damage. Arterial wall injuries may also result from the degree of arterial overstretch. For a comprehensive understanding of this intricate process, refer to our previous works [1–3,45,46], which provide detailed elaborations and illustrations of the ISR procedure, including influencing factors, wall mediators, and their interactions.

2.1.2. Evolution of biological mediators: governing PDEs

To model the pathophysiologic process, a set of evolution equations is established for the five biological mediators (PDGF, TGF-β, ECM, SMCs, EC) present within the arterial wall, coupled with the drug concentration and growth kinematics. The cellular mediator SMCs of the arterial wall are expressed in cell densities (ρ_S, unit: cells/mm^3), whereas the extracellular mediators, including PDGF, TGF$-\beta$, ECM, and the drug, are quantified in terms of their concentrations (c_P, c_T, c_C, and c_D, unit: mol/mm^3). The monolayer ECs are characterized by cell densities per unit area (ρ_E, unit: cells/mm^2). The equations describing the evolution of mediators within the arterial wall are formulated in Eulerian form. Note that the existing modeling framework does not encompass the fluid–solid interaction between the blood flow and the arterial wall.

For the incorporation of mediator transport and interaction, advection–reaction–diffusion type equations derived from our prior studies [1–3] are employed and succinctly outlined here without extensive elaboration. Readers keen on detailed insights into equation construction and associated quantities are directed to the referenced work.

$$\text{PDGF: } \left.\frac{\partial c_P}{\partial t}\right|_{\boldsymbol{x}} + \operatorname{div}\left(c_P\, \boldsymbol{v}\right) = \underbrace{\operatorname{div}\left(D_P \ \operatorname{grad} c_P\right)}_{\text{diffusion}} + \underbrace{\eta_P\, \rho_S\, c_T}_{\text{autocrine secretion by SMCs}} - \underbrace{\varepsilon_P\, f_T\, \rho_S\, c_P}_{\text{receptor internalization}}, \tag{1}$$

$$\text{TGF}-\beta\text{: } \left.\frac{\partial c_T}{\partial t}\right|_{\boldsymbol{x}} + \operatorname{div}\left(c_T\, \boldsymbol{v}\right) = \underbrace{\operatorname{div}\left(D_T \ \operatorname{grad} c_T\right)}_{\text{diffusion}} - \underbrace{\varepsilon_T\, \rho_S\, c_T}_{\text{receptor internalization}}, \tag{2}$$

$$\text{ECM: } \left.\frac{\partial c_C}{\partial t}\right|_{\boldsymbol{x}} + \operatorname{div}\left(c_C\, \boldsymbol{v}\right) = \underbrace{\eta_E\, \rho_S \left(1 - c_C / c_{C,th}\right)}_{\text{secretion by synthetic SMCs}} - \underbrace{\varepsilon_E\, c_P\, c_C}_{\text{MMP-induced degradation}}, \tag{3}$$

$$\begin{aligned}\text{SMC: } \left.\frac{\partial \rho_S}{\partial t}\right|_{\boldsymbol{x}} + \operatorname{div}\left(\rho_S\, \boldsymbol{v}\right) = & -\underbrace{\operatorname{div}\left(\chi_C \left(1 - c_C / c_{C,th}\right) \rho_S \ \operatorname{grad} c_P\right)}_{\text{chemotaxis}} \\ & + \underbrace{\operatorname{div}\left(\chi_H\, f_P\, \rho_S \ \operatorname{grad} c_C\right)}_{\text{haptotaxis}} \\ & + \underbrace{\eta_S\, f_T\, f_D\, c_P\, \rho_S \left(1 - c_C / c_{C,th}\right)}_{\text{proliferation}},\end{aligned} \tag{4}$$

$$\begin{aligned}\text{EC: } \left.\frac{\partial \rho_E}{\partial t}\right|_{\boldsymbol{x}} + \operatorname{div}\left(\rho_E\, \boldsymbol{v}\right) = & \operatorname{div}\left(D_E \ \operatorname{grad} \rho_E\right) + \underbrace{\eta_E\, f_{E1}\, \rho_E \left(1 - \rho_E / \rho_{E,eq}\right)}_{\text{proliferation}} \\ & - \underbrace{\varepsilon_E\, f_{E2}\, \rho_E}_{\text{apoptosis}},\end{aligned} \tag{5}$$

$$\text{Drug: } \left.\frac{\partial c_D}{\partial t}\right|_{\boldsymbol{x}} + \operatorname{div}\left(c_D\, \boldsymbol{v}\right) = \underbrace{\operatorname{div}\left(D_D \ \operatorname{grad} c_D\right)}_{\text{diffusion}} - \underbrace{\varepsilon_D\, \rho_S\, c_D}_{\text{receptor internalization}}. \tag{6}$$

The scaling functions f_T (Eqs. (1) and (4)), f_P and f_D (Eq. (4)) describe the degree of influence of the mediator concentration on the corresponding coupled quantity, defined as

$$\begin{aligned} f_T(c_T) &:= \frac{1}{1 + e^{l_T \left(c_T - c_{T,th}\right)}}, \quad f_P(c_P) := \frac{1}{1 + e^{-l_P \left(c_P - c_{P,th}\right)}}, \\ f_D(c_D) &:= \frac{\beta\,(c_D)^{\alpha} + A^{\alpha}}{(c_D)^{\alpha} + A^{\alpha}}. \end{aligned} \tag{7}$$

The influence of drug concentration on the EC proliferation and apoptosis is controlled by the scaling functions f_{E1} and f_{E2} (Eq. (5)), respectively, which are defined as

$$f_{E1}(c_D) := 1 - \frac{1}{100}\left(\frac{A_E\, c_D^{\beta}}{c_D^{\beta} + B_E^{\beta}}\right), \quad f_{E2}(c_D) := 1 - \exp\left(-l_E\, c_D\right). \tag{8}$$

Other quantities that shown up in the equations (Eqs. (1)–(6)) and functions (Eqs. (7) and (8)) mentioned above are constant material or fitting parameters.

2.1.3. Continuum mechanical modeling of arterial response and growth

The mechanical response of the arterial wall primarily depends on the medial and adventitial layers. Each layer is assumed to consist of two families of collagen fibers embedded within an isotropic ground matrix. Constitutively, we assume that the growth process within this matrix is driven by smooth muscle cells (SMCs). Furthermore, the collagen, and consequently the extracellular matrix, significantly impacts the compliance of the arterial wall, which should be also considered in the free energy function.

Within the framework of large deformation, the deformation gradient $\mathbf{F}$ is multiplicatively decomposed into an elastic part and a growth part, i.e,

$$\mathbf{F} = \mathbf{F}_e \mathbf{F}_g = \mathbf{F}_e \mathbf{R}_g \mathbf{U}_g := \mathbf{F}_* \mathbf{U}_g, \tag{9}$$

with the corresponding polar decomposition of the growth deformation gradient tensor $\mathbf{F}_g = \mathbf{R}_g \mathbf{U}_g$. The relationship between the global right Cauchy–Green tensor $\mathbf{C} = \mathbf{F}^{\mathrm{T}}\mathbf{F}$ and the intermediate right Cauchy–Green tensor $\mathbf{C}_*$ can be established according to

$$\mathbf{C}_* = \mathbf{F}_*^{\mathrm{T}} \mathbf{F}_* = \mathbf{U}_g^{-1} \mathbf{C} \mathbf{U}_g^{-1}. \tag{10}$$

Having this, the Helmholtz free energy per unit volume in the reference configuration can be decomposed into an isotropic part associated with the isotropic ground matrix, and an anisotropic part corresponding to the collagen fibers:

$$\psi := \psi_{iso}(\mathbf{C}_*, \mathbf{U}_g) + \psi_{ani}(\mathbf{C}, \mathbf{H}_1, \mathbf{H}_2, c_C^0), \tag{11}$$

with the structural tensors $\mathbf{H}_1, \mathbf{H}_2$ defining the orientation of the initial collagen fibers arranged in two distinct helices, and c_C^0 being the local ECM concentration in the reference configuration. Specifically, the isotropic part adheres to the compressible Neo-Hookean hyperelastic material behavior, while the anisotropic part demonstrates exponential dependence on the stretch in the direction of the collagen fibers. Details on the definition of the structural tensors, choice for the free energy functions, and mappings between various quantities can be found in our previous work [1,2].

The constitutive relation for growth can be simply described using the right stretch tensor $\mathbf{U}_g$ related to the growth deformation gradient $\mathbf{F}_g$, which resides solely in the reference configuration. The constitutive definition of the growth stretch tensor $\mathbf{U}_g$ is formulated to depend on the referential SMC density ρ_S^0 and is proposed to follow a specific form for positions with

- distinct collagen orientations with negligible dispersion ($\kappa = 0$):

$$\mathbf{U}_g := \mathbf{I} + \left(\frac{\rho_S^0}{\rho_{S,eq}} - 1\right) \mathbf{n}_g \otimes \mathbf{n}_g, \quad \mathbf{n}_g := \frac{\mathbf{a}_1^0 \times \mathbf{a}_2^0}{\|\mathbf{a}_1^0 \times \mathbf{a}_2^0\|}; \tag{12}$$

- diffuse collagen orientations with dispersion ($\kappa \neq 0$):

$$\mathbf{U}_g := \left(\frac{\rho_S^0}{\rho_{S,eq}}\right)^{1/3} \mathbf{I}. \tag{13}$$

Here, $\rho_{S,eq}$ denotes the homeostatic SMC density of a healthy artery. The vectors $\mathbf{a}_1^0$ and $\mathbf{a}_2^0$ represent the directions of the initial collagen fibers defined in the reference configuration. In both constitutive scenarios, it can be demonstrated that the determination of the growth tensor is equivalent to the relation between the current SMC density and the equilibrium SMC density in 3D, i.e., $\det(\mathbf{U}_g) = \rho_S^0 / \rho_{S,eq} = \det(\mathbf{F}_g) := J_g$.

2.1.4. Model setup, boundary conditions, material parameters and numerical solution

Model definition and finite element discretization:

To investigate the spatiotemporal progression of mediators following stent implantation and create training datasets, a simplified stented arterial quadrant is chosen, as illustrated in Fig. 1(a). This initial

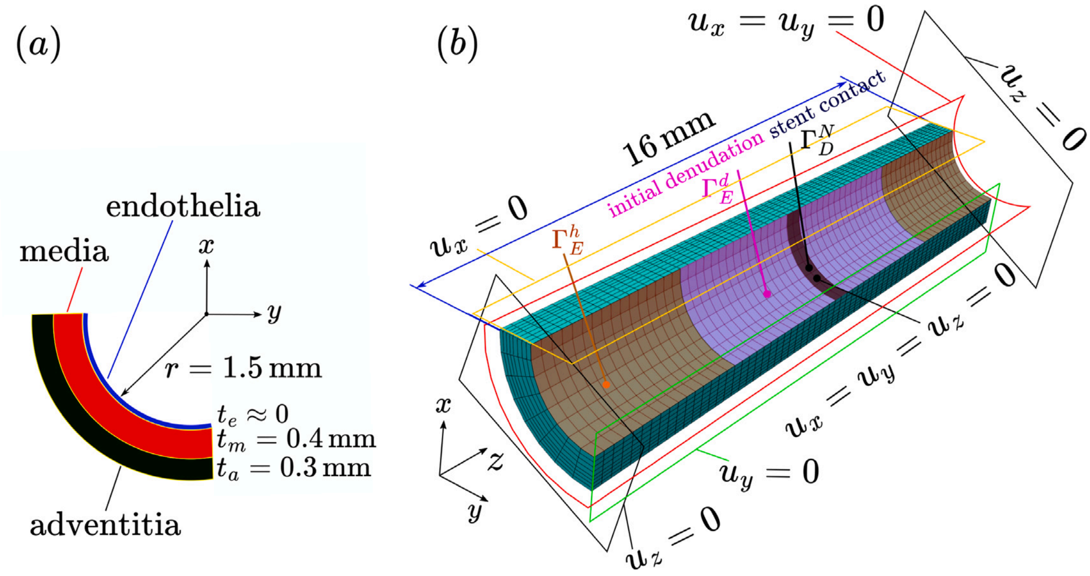


Fig. 1. Left: schematic illustration of coronary artery structure, including adventitia, media, and endothelia layers. Right: Finite element discretization and boundary conditions.

boundary value problem (IBVP) serves as the basis for modeling the complex interaction between mediators and drug diffusion within the stented artery. The bulk response of the arterial wall is characterized by two concentric layers (media and adventitia), while the endothelium is represented as a surface on the luminal side.

The spatial discretization of the geometry is accomplished using trilinear hexahedral elements. Along the longitudinal direction, 64 elements are employed, while 8 elements are used along the circumference, and 4 along the radial direction for each layer of the arterial wall, as depicted in Fig. 1(b). Projected surface meshes on the luminal side define the stent apposition region, as well as the denuded and healthy endothelial regions. Temporal discretization of the problem is achieved with a time step Δt set to 1 [day] for a total modeling time of 180 days.

Initial and boundary conditions:

The symmetry of the arterial wall is specified by the displacement boundary conditions, as outlined in Fig. 1. The end faces are fixed against longitudinal displacement, while the abluminal side of the vessel wall is restricted from movement in the circumferential and radial directions. To emulate the interaction between the stent and artery, fixed boundary conditions and drug fluxes are defined on the contact region Γ_D^N. In terms of biological variables, the initial concentration of ECM and SMC densities are prescribed to be those of a healthy homeostatic vessel in equilibrium. PDGF, TGF$-\beta$, and the drug are assumed to be initially absent in the vessel wall. Furthermore, the initial value for endothelial cells is set to be zero on the denuded region Γ_E^d and matches the value of a healthy artery elsewhere Γ_E^h. No Dirichlet boundary condition is prescribed for the biological mediators.

Regarding the Neumann boundary conditions, PDGF and TGF-β enter the arterial wall as a consequence of platelet aggregation after endothelial denudation via stent implantation. This effect can be modeled by prescribing influxes along the normal $\boldsymbol{n}$ at the denudation region Γ_E^d, which are exposed directly to the blood flow. These influxes are directly prescribed as time-varying profiles, dependent on the load-factor $l_f^1(t)$ and the peak influxes q_P^{ref} and q_T^{ref} of PDGF and TGF-β, respectively. Homogeneous Neumann boundary conditions are applied to the remaining uninjured region of the vessel wall. Similarly, as mentioned earlier, the drug is assumed to enter the vessel wall at the contact region Γ_D^N, where the drug influx is again prescribed as a time-varying profile, dependent on the load-factor $l_f^2(t)$ and the peak influx q_D^{ref}. It is worth noting that additional drug can enter the vessel wall due to downstream deposition from blood flow and lipophilicity of the drug. These effects are excluded in current modeling for simplicity. ECM and SMCs are considered to be restrained within the arterial wall, and hence homogeneous Neumann boundary conditions are assumed across the entire boundary of the system. Likewise, there is no influx for ECs from the boundary lines into the inner area of the lumen. Details regarding the boundary and initial conditions are summarized in Table 1.

Table 1
Initial and boundary conditions.

Name	Variable	Initial	Dirichlet	Neumann
PDGF	c_P	0 in Ω	–	$q_P(t) = l_f^1(t)\, q_P^{\mathrm{ref}}$ on Γ_E^d
TGF-β	c_T	0 in Ω	–	$q_T(t) = l_f^1(t)\, q_T^{\mathrm{ref}}$ on Γ_E^d
ECM	c_C	$c_{C,eq}$ in Ω	–	–
SMC	ρ_S	$\rho_{S,eq}$ in Ω	–	–
EC	ρ_E	$\begin{cases} 0 & \text{on } \Gamma_E^d \\ \rho_{E,eq} & \text{on } \Gamma_E^h \end{cases}$	–	–
Drug	c_D	0 in Ω	–	$q_D(t) = l_f^2(t)\, q_D^{\mathrm{ref}}$ on Γ_{D}^N
Disp.	$\mathbf{u}$	0 in Ω	$\mathbf{u} = \bar{\mathbf{u}}$ on Γ_u	–

The load-factor $l_f^1(t)$ and $l_f^2(t)$ introduced earlier are designed to account for the time-dependent effect of ECs-recovery and drug-elution and permeability of the vessel to blood contents. Studies, such as [47], have reported that the presence of sirolimus-based drugs on the endothelium leads to a decrease in endothelial cell viability and an increase in wall permeability to blood contents. The chosen time-dependent load-factor profiles are structured in the following manner [2]:

$$l_f^1(t) := \left(1 - \exp\left(-l_B\, t\right)\right)\left(1 - \frac{\rho_E}{\rho_{E,eq}}\right), \tag{14}$$

$$l_f^2(t) := \exp\left(-\frac{t}{t_c\left(\exp\left(t_p/t_c\right) - 1\right)}\right)\left(1 - \exp\left(-\frac{t}{t_c}\right)\right), \tag{15}$$

respectively, where the parameter l_B refers to the slope of the influx profiles and the curve $l_f^1(t)$ has a similar course as f_{E2}(Eq. (8))$_2$ when the endothelium is absent. In Eq. (15), t_p and t_c are time-like parameters which control the curvature of the influx profile, where t_p indicates the time required to achieve peak influx value of the drug. The chosen values for l_B, t_p and t_c in this study are $1/30\,[\mathrm{day}^{-1}]$, $5\,[\mathrm{days}]$ and $5\,[\mathrm{days}]$, respectively.

Table 2
List of parameters related to the biological mediators.

Parameter	Description	Value	Units
PDGF			
D_P	Diffusivity	0.1	[mm^2/day]
η_P	Autocrine secretion coefficient	1.0×10^{-6}	[mm^3/cell/day]
ε_P	Receptor internalization coefficient	1.0×10^{-7}	[mm^3/cell/day]
$c_{P,th}$	PDGF threshold for the scaling function	1.0×10^{-15}	[mol/mm^3]
l_P	Steepness coefficient for PDGF dependent scaling	1.0×10^{16}	[mm^3/mol]
TGF-β			
D_T	Diffusivity	0.1	[mm^2/day]
ε_T	Receptor internalization coefficient	1.0×10^{-7}	[mm^3/cell/day]
$c_{T,th}$	TGF-β threshold for the scaling function	1.0×10^{-16}	[mol/mm^3]
l_T	Steepness coefficient TGF$-\beta$ dependent scaling	1.0×10^{16}	[mm^3/mol]
ECM			
η_C	Collagen secretion coefficient	1.0×10^{-7}	[mol/cell/day]
ε_C	Collagen degradation coefficient	1.0×10^{21}	[mm^3/mol/day]
$c_{C,eq}$	Collagen concentration in a healthy artery	7.0×10^{-9}	[mol/mm^3]
$c_{C,th}$	Collagen secretion threshold	7.0007×10^{-9}	[mol/mm^3]
SMC			
χ_C	Chemotactic sensitivity	1.0×10^{11}	[mm^5/mol/day]
χ_H	Haptotactic sensitivity	1.0×10^{6}	[mm^5/mol/day]
η_S	Proliferation coefficient	1.0×10^{14}	[mm^3/cell/day]
$\rho_{S,eq}$	SMC density of a healthy artery	3.7×10^{5}	[cells/mm^3]
EC			
D_E	Diffusivity	0.01	[mm^2/day]
η_E	Proliferation coefficient	0.1	[day]$^{-1}$
$\rho_{E,eq}$	EC density on healthy vascular endothelia	500	[cells/mm^2]
A_E	Maximum efficacy of the drug against EC proliferation	65.4%	[−]
B_E	Drug concentration for half efficacy	9.19	[nM]
β	Hill coefficient for drug-induced EC proliferation inhibition	1.46	[−]
ε_E	Apoptosis coefficient	0.01	[day]$^{-1}$
l_E	Drug-dose-dependent apoptosis parameter	0.2	[nM]$^{-1}$
Drug			
D_D	Diffusivity	0.1	[mm^2/day]
ε_D	Receptor internalization coefficient	1.0×10^{-7}	[mm^3/cell/day]

Table 3
List of parameters related to the mechanical response.

Parameter	Description	Value {media, adventitia}	Units
μ	Shear modulus for the matrix	$\{0.02, 0.008\}$	[MPa]
Λ	Lamé parameter for the matrix	10	[MPa]
$\bar{k}_1$	Stress-like parameter for collagen fibers	$\{0.112, 0.362\}$	[MPa]
k_2	Exponential coefficient for collagen fibers	$\{20.61, 7.089\}$	[−]
α_a	Collagen orientation angle w.r.t circumference	$\{41, 50.1\}$	[°]

Material parameters:

The biological material parameters are listed in Table 2. These values pertain to the rates of production, degradation, proliferation, and differentiation, as well as the diffusion coefficients of the biological species, initial concentrations, and threshold values considered. The mechanical material parameters are listed in Table 3. Details on the choice and determination of material parameters are provided in our previous work [1,2].

Numerical solution:

In the FEM implementation, the governing coupled differential equations are transformed to the Lagrangian description by neglecting the advection term. The weak form of all balance equations, excluding the EC density equation, are solved within the arterial wall domain using trilinear hexahedral elements as mentioned above. The EC density field is addressed on the lumen surface, discretized with bilinear quadrilaterals projected from the bulk arterial wall mesh. This projected surface mesh is also employed to impose flux (inhomogeneous Neumann) boundary conditions. Temporal discretization utilizes the fully-implicit backward-Euler method.

For the SMC density field, the streamline-upwind Petrov–Galerkin (SUPG) method is incorporated to ensure robust finite element computations, particularly at high chemotactic and haptotactic velocities of the SMCs, as shown in our previous work [2]. The system matrices are interconnected at common nodes shared between the bulk hexahedrons and surface quadrilaterals. The implemented framework is integrated into the commercial finite element program FEAP through its user-defined element interface. Tangential stiffness matrices in each global Newton iteration of the nonlinear problem are derived through algorithmic differentiation using the Wolfram Mathematica package AceGen [48]. The solution of the linear system of equations in every global Newton iteration is achieved with the parallelized PARDISO solver [49] through a user-defined subroutine.

2.2. Non-intrusive data-driven ROM

In this section, our proposed method for non-intrusive ROM using 3D convolutional autoencoder (CAE) is introduced. Two distinct approaches pertaining to the treatment of input datasets are elucidated, including their network architectures and training processes.

2.2.1. General framework of autoencoders

An autoencoder (AE) is a type of neural network (NN) designed to learn the identity mapping, denoted as $\boldsymbol{h} : \boldsymbol{x} \mapsto \tilde{\boldsymbol{x}}$, where $\tilde{\boldsymbol{x}}$ is an approximation of $\boldsymbol{x}$, and $\boldsymbol{h} : \mathbb{R}^{n_x} \mapsto \mathbb{R}^{n_x}$. Herein, the AE comprises an encoder, $\boldsymbol{h}_{\text{enc}} : \boldsymbol{x} \mapsto \hat{\boldsymbol{x}}$ with $\boldsymbol{h}_{\text{enc}} : \mathbb{R}^{n_x} \mapsto \mathbb{R}^{n_{\hat{x}}}$, mapping a high-dimensional vector/tensor $\boldsymbol{x}$ to a low-dimensional code $\hat{\boldsymbol{x}}$ (also referred to latent variable or latent vector, etc.), and a decoder, $\boldsymbol{h}_{\text{dec}} : \hat{\boldsymbol{x}} \mapsto \tilde{\boldsymbol{x}}$ with $\boldsymbol{h}_{\text{enc}} : \mathbb{R}^{n_{\hat{x}}} \mapsto \mathbb{R}^{n_x}$, mapping the low-dimensional code $\hat{\boldsymbol{x}}$ back to an approximation of the original vector/tensor $\tilde{\boldsymbol{x}}$. Thereby, the complete AE takes the form

$$\boldsymbol{h} : \boldsymbol{x} \mapsto \boldsymbol{h}_{\text{dec}} \circ \boldsymbol{h}_{\text{enc}}(\boldsymbol{x}). \tag{16}$$

If the approximation $\boldsymbol{h}(\boldsymbol{x}) \approx \boldsymbol{x}$ holds across a dataset $\boldsymbol{x} \in \{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \dots, \boldsymbol{x}^{(i)}, \dots, \boldsymbol{x}^{(m)}\}$, it implies that the low-dimensional code $\boldsymbol{h}_{\text{enc}}(\boldsymbol{x}^{(i)})$ contains sufficient information for precise recovery of $\boldsymbol{x}^{(i)}$ through the decoder $\boldsymbol{h}_{\text{dec}}$. This property provides a natural rationale for employing AE in ROM, especially when the dataset aligns with high-fidelity data, such as snapshots.

Various NN architectures can be employed for the encoding-decoding task, including multilayer perceptrons (MLP). In this context, the encoder's architecture comprises a sequence of fully connected layers, with the last layer, often referred to as the bottleneck layer, capturing the compressed state of the high-order systems. These compressed states serve as representations of the reduced order states of the input. Subsequently, the decoder utilizes these reduced states from the bottleneck layer (i.e., the input layer for the decoder) to reconstruct the input in the original dimensional space. The decoder is implemented as another network of fully connected layers.

The MLP-based AE has been proven to be advantageous when handling one-dimensional vector inputs, especially when the correlations between local spatial characteristics are not required [28]. However, when addressing higher-dimensional inputs with the need to capture local features, the MLP-based AE might face challenges in accuracy and training. Moreover, it is important to note that the MLP-based AE operates as a fully black-box model.

In our work, inspired by applications in picture or video processing, we introduce the CAE to overcome this limitation by specifically aiming to address higher-dimensional inputs and capture local features. Convolutional layers emerge as advantageous in this context, since they can handle multi-channel spatially distributed input data and correlate the spatial properties. Furthermore, the weight-sharing characteristic of convolutional neural networks (CNNs) enhances their computational efficiency when compared to fully connected layers.

2.2.2. Convolutional autoencoder

CAE differs from MLP-based AE by using convolutional layers for the primary dimensionality reduction. In CAEs, the encoder processes features through a sequence of convolutional layers, leading to a bottleneck layer through a single fully-connected layer. During decoding, transposed convolutions, loosely referred to as deconvolutions, are employed to upsample the bottleneck layer back to the original higher-dimensional space. To provide a clearer understanding of the 3D NN architecture used in the upcoming section, the fundamental operations and notations (e.g. kernel, stride, pooling, channel, hyperparameters, learnable parameters, etc.) employed in convolutional and transposed convolutional layers in 2D are shortly introduced. The principles outlined for 2D layers can be easily extended to 3D layers without any conceptual differences.

In convolutional layers, units are structured as feature maps (denoted as $\mathcal{H}$), where each unit establishes connections with local patches of the feature maps from the previous layer. This connection is realized through discrete convolutions defined by a set of kernels (denoted as $\mathcal{W}$), followed by a nonlinear activation function and a pooling operation. Hence, in 2D, the feature map at layer l can be interpreted as a 3-dimensional tensor $\mathcal{H}^l \in \mathbb{R}^{n^l_{\text{chan}} \times n^l_1 \times n^l_2}$ with element $\mathcal{H}^l_{i,j,k}$ signifying a unit within channel i at row j and column k, while the filter kernel can be considered as a 4-dimensional tensor $\mathcal{W}^l \in \mathbb{R}^{n^l_{\text{filter}} \times n^{l-1}_{\text{chan}} \times k^l_1 \times k^l_2}$ with element $\mathcal{W}^l_{i,j,m,n}$ connecting a unit in channel i of the output to a unit in channel j of the input, with an offset of m rows and n columns between the output unit and the input unit. The number of filters in the kernel is denoted by n_{filter}, and the kernel length is characterized by k_1 and k_2. Thereby, the mathematical connection between layer $(l-1)$ and layer l is established as follows [9]:

$$\mathcal{H}^l_{i,j,k} = \phi_l\left(\sum_{r,m,n} \mathcal{H}^{l-1}_{r,\,(j-1)\times s+m,\,(k-1)\times s+n}\,\mathcal{W}^l_{i,r,m,n} + \mathcal{B}^l_{i,j,k}\right), \tag{17}$$

for all channels i, where $\mathcal{B}^l_{i,j,k}$ and ϕ_l are the bias and activation function, respectively. One important hyperparameter in Eq. (17) is the stride size s (could be different for rows and columns), which determines the downsampling rate of each convolution. If $s > 1$, the dimension of the next feature map is reduced by a factor of s in that direction. The filters $\mathcal{W}$ and the biases $\mathcal{B}$ are learnable parameters, whereas the kernel length $[k_1, k_2]$, the number of filters n_{filter}, and the stride s are hyperparameters. If necessary, after one convolutional operation, pooling layers (such as average pooling, max pooling, etc.) can be applied to the output. This process not only offers a statistical evaluation (average, maximum, etc.) of the neighboring units at specific locations but also downsamples the output.

Conversely, a deconvolutional layer performs the inverse operation of convolution, which is employed in constructing decoding layers. Its fundamental role involves element-wise multiplication of each input value by a filter. Consequently, the filters learned in the deconvolutional layers serve as the basis for reconstructing the shape of the inputs, taking into account the desired shape of the output. Like convolutional layers, a deconvolutional layer may incorporate multiple filters, and the stacking of several deconvolutional layers allows for the creation of deep architectures in CAEs [29]. Similarly, after a deconvolutional operation, unpooling layers (such as nearest neighbor, bed of nails, and max unpooling) can be applied to the output, generally resulting in upsampling. For a more detailed understanding of (de-)convolution arithmetic in deep learning, the readers are referred to [9,29,50–53].

2.3. Data generation, network architecture and training

The model presented in Section 2.1 clarifies the intricate nature of coronary ISR, including the evolution equations and the wide parameter space, as well as the initial and boundary condition. In the surrogate modeling, one can theoretically study all the possible parameters at once. This leads to unmanageable mount of data and training burden. A common process is to concentrate on the most important parameter(s). To assess the applicability and accuracy of the proposed data-driven non-intrusive surrogate method, we constrain the variation to a single parameter, specifically the peak influx of drug q_D^{ref} as defined in the penultimate row of Table 1. This strategic choice of drug-elution dosage is important for the clinical application, since it is rooted in its critical role in influencing the evolution of biological mediators and the degree of ISR. By narrowing down the parameter space to focus on drug-elution dosages, we aim to not only understand their impact but also to identify an optimal value for the peak drug influx, minimizing the degree of ISR post PCI.

Referring to the spatial discretization outlined in Section 2.1.4, where $64 \times 8 \times 8$ hexahedral elements are assigned for PDGF, TGF-β, ECM, SMCs, and drug concentration along the longitudinal, circumferential, and radial directions, respectively. Simultaneously, 64×8 quadrilateral elements are designated for ECs on the lumen surface (Fig. 1). The values on the grid points are collected to construct the input dataset. Note that the values on the Gaussian points can also be used for the construction of datasets. The temporal discretization involves a total of 180 time steps, with each step representing one day. As a result, the complete dataset for one parameter takes the form of a 5D tensor with dimensions $(180, 9, 9, 65, 9)$, where the numbers represent time series, radial, circumferential, longitudinal dimensions, and the number of output channels (u_x, u_y, u_z, c_P, c_T, c_C, ρ_S, ρ_E, and c_D), respectively. For the generation of the parameter space, a set of 50 instances is designed for q_D^{ref}. These parameter instances are

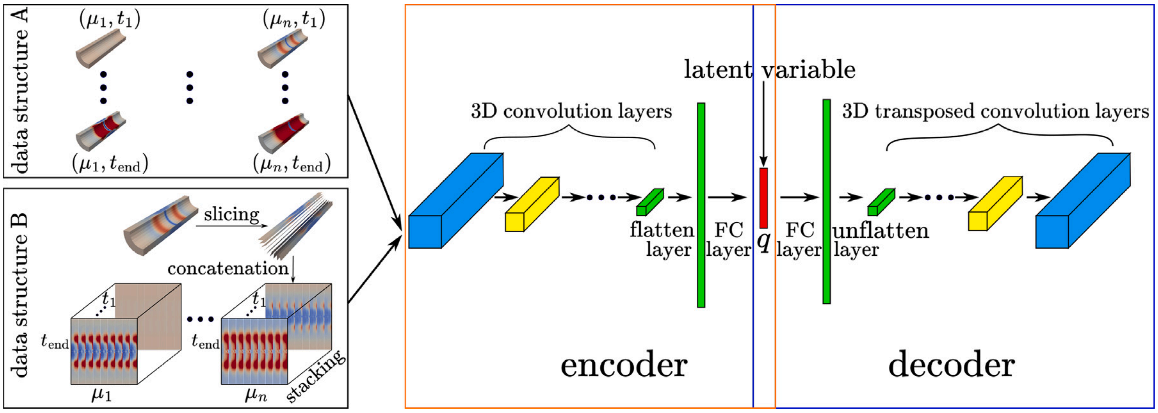


Fig. 2. Network architecture of the 3D deep convolutional autoencoder (CAE). In the downsampling process, the encoder transforms the input into the low-dimensional latent variable $\boldsymbol{q}$ using 3D convolution, flatten and dense/fully connected (FC) layers. Subsequently, in the upsampling process, the decoder reconstructs the high-dimensional solution from the latent variable $\boldsymbol{q}$ applying FC, reshape/unflatten and 3D transposed convolution layers. The 3D CAE is fed with two distinct types of input datasets: in the case of data structure A, the time is treated as an additional parameter; in the case of data structure B, the 3D geometry is firstly reshaped into a 2D plane along the circumferential axis and all time steps are stacked for each parameter instance.

evenly distributed in the range from 0 to 50 fmol/mm^2/day. In general, having more training data allows a CAE to better learn and reflect the surrogate model, resulting in more accurate and generalizable predictions. However, generating large volumes of training data and the corresponding training require heavy computational costs, which can impact overall efficiency if not managed carefully. Balancing this trade-off is challenging, as the value of a surrogate model also depends on its intended use. For cases involving numerous evaluations, the initial investment in generating training data can be enlarged. Conversely, if only a few evaluations are needed, the cost of generating data and training might outweigh the efficiency gains.

For handling the input datasets, two strategies are investigated. In the first approach, time is treated as an additional parameter, and 3D convolution is applied to individual time steps, creating distinct latent variables for each parameter instance within each time step, as illustrated in Fig. 2 using data structure A. The second approach involves initially reshaping the 3D geometry into a 2D plane by concatenating the slices along the circumferential axis, and stacking all time steps in the third direction for each parameter instance. This reconfiguration results in a more extensive and comprehensive dataset for one parameter instance, yielding a singular latent variable across the entire discrete time series, as illustrated in Fig. 2 using data structure B. The high-fidelity data undergoes normalization before being fed into the neural network. In both approaches, convolutions automatically account for multiple outputs. The neural network architecture primarily varies in the dimensions of input and latent variables due to the difference in the size of individual datasets between data structures A and B.

The detailed network architectures for the CAEs for both approaches are summarized in Tables 4 and 5. Specifically, the encoder incorporates zero-padding layers, strategically placed before the convolutional layers, to achieve a more efficient dimensional reduction. This is followed by multiple 3D convolution layers with uniform kernel sizes of 9 in all directions, diverse strides, and increasing channel dimensions. For data structures A and B, a latent space dimension larger than 10 and 25 respectively, has proven sufficient for achieving high accuracy. It is worth noting that while higher-dimensional latent spaces generally offer improved accuracy, they may also lead to diminishing returns or overfitting beyond a certain point. The decoder is adjusted to harmonize the output dimension with the original input, which encompasses dense, reshape/unflatten, 3D transposed convolution and cropping layers with corresponding mirroring parameters.

The training process involves a maximum number of 10 000 epochs, a batch size of 10, and a learning rate of $\eta = 10^{-4}$. Early stopping, with a patience of 10, is implemented to avoid overfitting. The input data is partitioned into training and validation sets with an 8:2 proportion. The ELU (Exponential Linear Unit) non-linear activation function are applied throughout the network, with the exception of the last convolutional layer in the decoder, where no activation function is employed—a common practice in CAEs [9,26]. Adam optimizer is applied and the training loss is evaluated using mean squared error (MSE). The NN architecture is implemented using TensorFlow with the Keras API. The training process takes place on the RWTH Aachen GPU cluster workstation equipped with dual NVIDIA Tesla V100 GPUs.

To study the convergence behavior concerning the dimension of the latent vectors, the relationship between the mean relative error (MRE) of the entire testing samples and the latent space dimension is depicted in Fig. 3, where the first 2000 iterations are displayed. For this purpose, the unified global architecture was designed with slight local compatibility adjustments on the number of neurons in the FC layer and 3D CNN layers. In Eq. (18), the quantities $\boldsymbol{u}_h(\boldsymbol{\mu}_{\text{test},i})$ and $\tilde{\boldsymbol{u}}_h(\boldsymbol{\mu}_{\text{test},i})$ are the discretized high-fidelity solution and CAE surrogate solution, respectively. Notably, for both scenarios, there is a consistent decrease in MRE as the latent dimension increases. However, it is worth noting that the MRE exhibits more substantial fluctuations for data structure B in general. Larger loss fluctuations in data structure B can be attributed to its increased complexity from reshaping 3D spatial data and stacking time-series into a higher-dimensional format. Additionally, optimization challenges and the performance of hyperparameters may further contribute to these fluctuations. For both cases, the errors converge to the order of 10^{-5} after approximately 5000 iterations for the chosen latent dimensions outlined in Tables 4 and 5, corresponding to green curves with $n = 12$ and $n = 30$ with training durations of 18.5 h and 6.6 h for data structures A and B, respectively. It is noted that despite the efficiency of CNNs in handling small to medium-sized datasets, significant challenges can arise when dealing with higher-dimensional data, such as those for complex simulations with fine spatial and temporal resolutions. These challenges include increased memory requirements, extended training times, and the need for a larger latent space to adequately capture essential features.

$$\varepsilon_{rel} = \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \left(\frac{\|\boldsymbol{u}_h(\boldsymbol{\mu}_{\text{test},i}) - \tilde{\boldsymbol{u}}_h(\boldsymbol{\mu}_{\text{test},i})\|_2}{\|\boldsymbol{u}_h(\boldsymbol{\mu}_{\text{test},i})\|_2} \right) \tag{18}$$

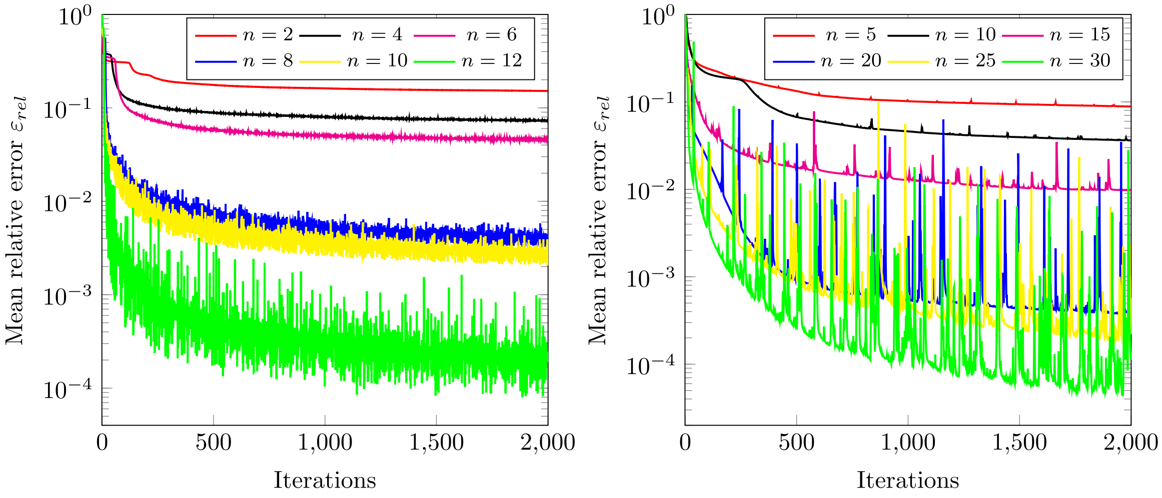
After training, GPR is employed to establish correlations between the latent variables and the parameter space. Although alternative regression methods such as artificial neural networks (ANN) are also applicable, GPR is chosen for its inherent flexibility, especially in extending correlations between vector-valued inputs and outputs. In

Table 4
The CAE network architecture for data structure A.

Encoder					
Layer	Input dimension	Output dimension	Kernel size	Number of filters	Strides
ZeroPadding3D	(9, 9, 65, 9)	(16, 16, 72, 9)	–	–	–
Conv3D	(16, 16, 72, 9)	(16, 16, 72, 16)	(9, 9, 9)	16	(1, 1, 1)
Conv3D	(16, 16, 72, 16)	(8, 8, 36, 32)	(9, 9, 9)	32	(2, 2, 2)
Conv3D	(8, 8, 36, 32)	(4, 4, 18, 64)	(9, 9, 9)	64	(2, 2, 2)
Conv3D	(4, 4, 18, 64)	(2, 2, 9, 128)	(9, 9, 9)	128	(2, 2, 2)
Conv3D	(2, 2, 9, 128)	(1, 1, 3, 256)	(9, 9, 9)	256	(2, 2, 3)
Flatten	(1, 1, 3, 256)	768	–	–	–
Dense	768	$\dim(q) = 12$	–	–	–
Decoder					
Layer	Input dimension	Output dimension	Kernel size	Number of filters	Strides
Dense	$\dim(q) = 12$	768	–	–	–
Reshape	768	(1, 1, 3, 256)	–	–	–
Conv3DTranspose	(1, 1, 3, 256)	(2, 2, 9, 256)	(9, 9, 9)	256	(2, 2, 3)
Conv3DTranspose	(2, 2, 9, 256)	(4, 4, 18, 128)	(9, 9, 9)	128	(2, 2, 2)
Conv3DTranspose	(4, 4, 18, 128)	(8, 8, 36, 64)	(9, 9, 9)	64	(2, 2, 2)
Conv3DTranspose	(8, 8, 36, 64)	(16, 16, 72, 32)	(9, 9, 9)	32	(2, 2, 2)
Conv3DTranspose	(16, 16, 72, 32)	(16, 16, 72, 9)	(9, 9, 9)	9	(1, 1, 1)
Cropping3D	(16, 16, 72, 9)	(9, 9, 65, 9)	–	–	–

Table 5
The CAE network architecture for data structure B.

Encoder					
Layer	Input shape	Output shape	Kernel size	Number of filters	Strides
ZeroPadding3D	(180, 81, 65, 9)	(180, 81, 72, 9)	–	–	–
Conv3D	(180, 81, 72, 9)	(180, 81, 72, 64)	(9, 9, 9)	64	(1, 1, 1)
Conv3D	(180, 81, 72, 64)	(60, 27, 24, 128)	(9, 9, 9)	128	(3, 3, 3)
Conv3D	(60, 27, 24, 128)	(20, 9, 8, 256)	(9, 9, 9)	256	(3, 3, 3)
Conv3D	(20, 9, 8, 256)	(10, 3, 4, 512)	(9, 9, 9)	512	(2, 3, 2)
Conv3D	(10, 3, 4, 512)	(5, 3, 2, 1024)	(9, 9, 9)	1024	(2, 1, 2)
Flatten	(5, 3, 2, 1024)	30 720	–	–	–
Dense	30 720	3000	–	–	–
Dense	3000	$\dim(q) = 30$	–	–	–
Decoder					
Layer	Input shape	Output shape	Kernel size	Number of filters	Strides
Dense	$\dim(q) = 30$	3000	–	–	–
Dense	3000	30 720	–	–	–
Reshape	30 720	(5, 3, 2, 1024)	–	–	–
Conv3DTranspose	(5, 3, 2, 1024)	(10, 3, 4, 1024)	(9, 9, 9)	1024	(2, 1, 2)
Conv3DTranspose	(10, 3, 4, 1024)	(20, 9, 8, 512)	(9, 9, 9)	512	(2, 3, 2)
Conv3DTranspose	(20, 9, 8, 512)	(60, 27, 24, 256)	(9, 9, 9)	256	(3, 3, 3)
Conv3DTranspose	(60, 27, 24, 256)	(180, 81, 72, 128)	(9, 9, 9)	128	(3, 3, 3)
Conv3DTranspose	(180, 81, 72, 128)	(180, 81, 72, 9)	(9, 9, 9)	9	(1, 1, 1)
Cropping3D	(180, 81, 72, 9)	(180, 81, 65, 9)	–	–	–



(a) data structure A.　(b) data structure B.

Fig. 3. Mean relative error — iteration diagrams.

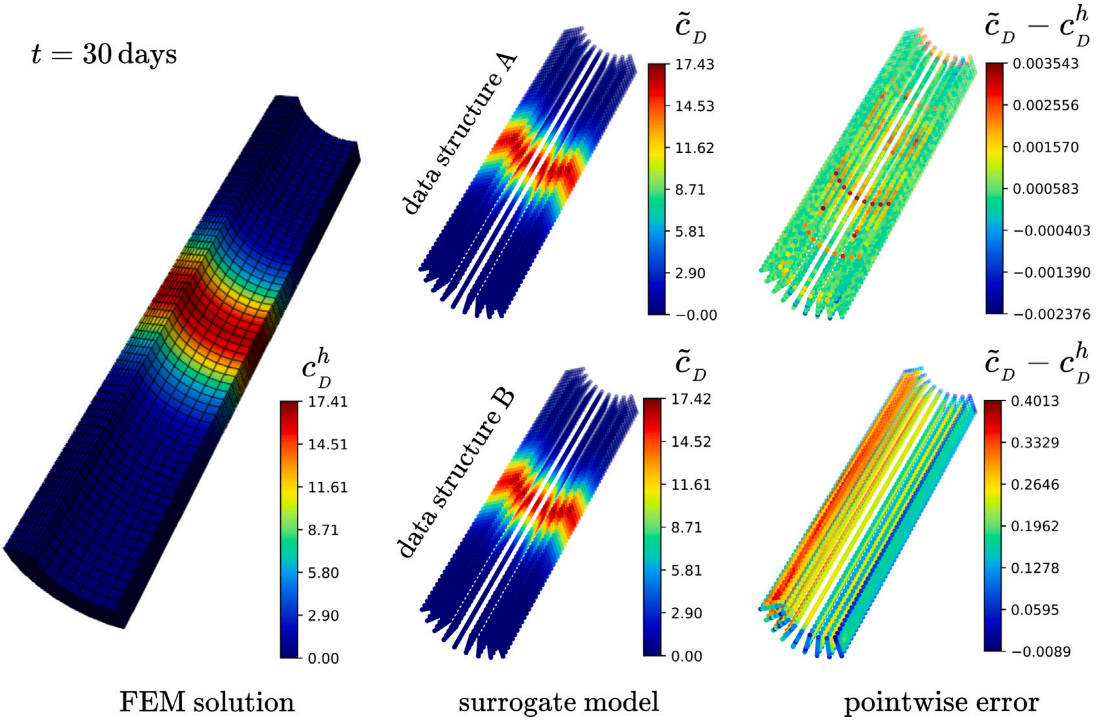


Fig. 4. Comparison of FOM and CAE-ROM solutions for the drug concentration c_D for the testing-parameter instance $\mu_{\text{test}} = (q_D^{\text{ref}} = 7.0\,\text{fmol/mm}^2/\text{day}, t = 30\,\text{days})$.

this work, the implementation of GPR utilizes the scikit-learn library and incorporates the Radial Basis Function (RBF) kernel to enhance the model's capabilities for capturing complex relationships within the data.

3. Results

In this section, we evaluate the performance of the proposed CAE-ROM. Our primary focus is on examining the accuracy of the method, specifically by comparing pointwise errors between the high-fidelity FEM solution and the CAE-ROM solution for both approaches. To illustrate, we choose the parameter for the peak influx of the drug $q_D^{\text{ref}} = 7.0\,\text{fmol/mm}^2/\text{day}$ (the choice of which is arbitrary) and examine the solutions of different outputs at various time steps.

Fig. 4 presents the solutions for the drug concentration 30 days after PCI applying both FOM and CAE-ROM. With data structure A, predominant errors emerge at the two ends and in the transition zones from the stent position to the region of lower drug concentration. When using data structure B, the errors are notably higher compared to data structure A. Interestingly, as observed in the lower right image, the errors appear to be concentrated slice by slice. Concerning the ECM concentration, depicted in Fig. 5, with the application of data structure A, errors are mainly located at the two ends and in the middle of the geometry, with relatively smaller errors observed in the transition zones. Conversely, when employing data structure B, the errors are concentrated primarily in the middle of the geometry. Regarding the EC density, defined exclusively on the inner surface of the arterial wall, comparable levels of errors are observed for both approaches, as illustrated in Fig. 6. Notably, errors primarily manifest at the surface of the arterial wall. It is important to mention that the presence of ECs inside the arterial wall of the surrogate model, deemed unphysical in our model, is a consequence of the characteristics of the CAE model. This occurrence is attributed to the model's treatment of all outputs at nodes simultaneously, without distinguishing between 3D solid elements and 2D surface elements. The displacement field in the longitudinal direction (u_z) and the smooth muscle cell density (ρ_S) exhibit significantly lower errors when employing both approaches, as depicted in Figs. 7 and 8. For error probes of other output quantities, refer to Appendix, Figs. A.10–A.13.

Upon global comparison of the results (refer to Figs. 4–8 and figures in Appendix), both approaches exhibit high accuracy, with data structure A consistently demonstrating smaller errors throughout the evaluation.

4. Discussion

With the trained and validated surrogate model in place, evaluation for new input parameters becomes straightforward. Our objective here is to estimate the optimal drug dosage, specifically, the peak value of flux for the drug q_D^{ref}, for prohibiting ISR. To achieve this, we partition the parameter space into finer instances, noting that some instances may not require recomputation since they overlap with the testing instances. We then apply the GPR to the newly emerged instances and decode the corresponding latent variables to obtain the full solution, focusing on assessing the rate of ISR at $30, 60$, and 90 days after PCI.

For evaluation, we exclusively utilize the first approach using data structure A, as it has demonstrated superior accuracy. Additionally, as discussed in Section 2.3, data structure A offers the advantage of the trained latent variable being correlated with both input parameters and time. This correlation allows for a continuous interpolation of time, achieved through GPR or alternative methods such as Dynamic Mode Decomposition (DMD), among others.

To estimate the degree of ISR after PCI, we rely on the constitutive definition of the growth tensor, represented by Eqs. (12) and (13). In both cases, the relationship between the actual SMC concentration and the equilibrium concentration is characterized by the determinant of the growth deformation gradient $\det \mathbf{F}_g$. Hence, we adopt the average value of this relationship for SMC evaluated across all nodal points as a simplified yet representative metric for the degree of ISR. It is worth mentioning that while the average measure offers valuable insights into global trends, there is a potential limitation as it may underestimate the severity at local positions, particularly in the context of complex geometries.

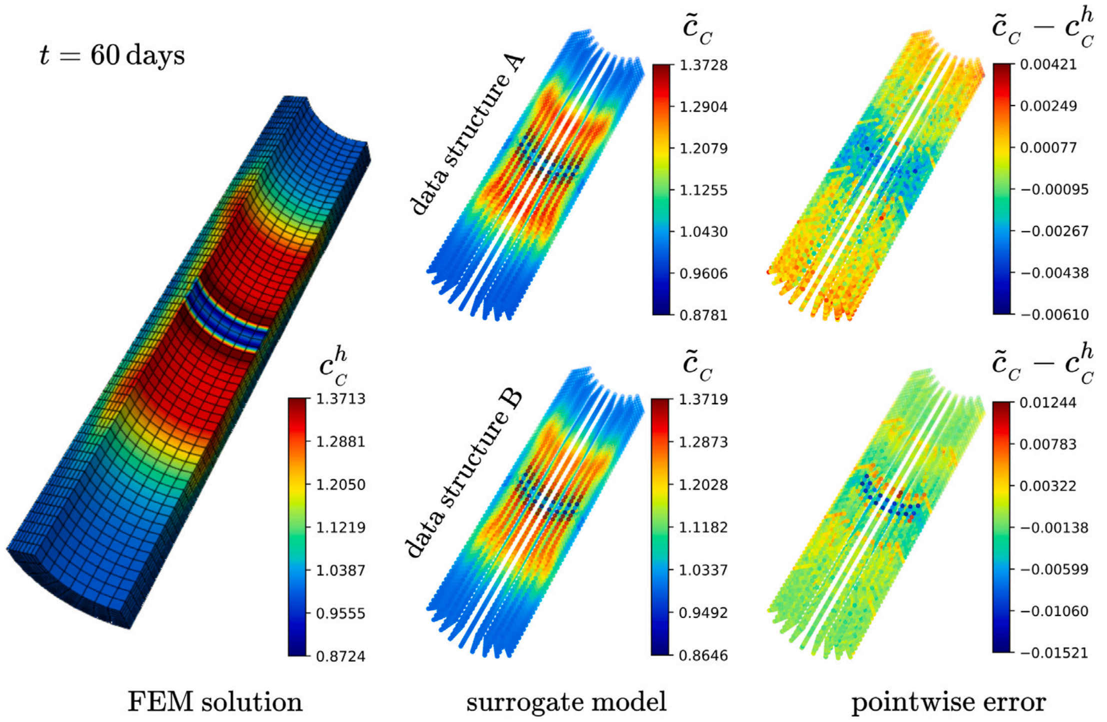


Fig. 5. Comparison of FOM and CAE-ROM solutions for the ECM concentration c_C for the testing-parameter instance $\mu_{\text{test}} = (q_D^{\text{ref}} = 7.0\,\text{fmol/mm}^2/\text{day}, t = 60\,\text{days})$.

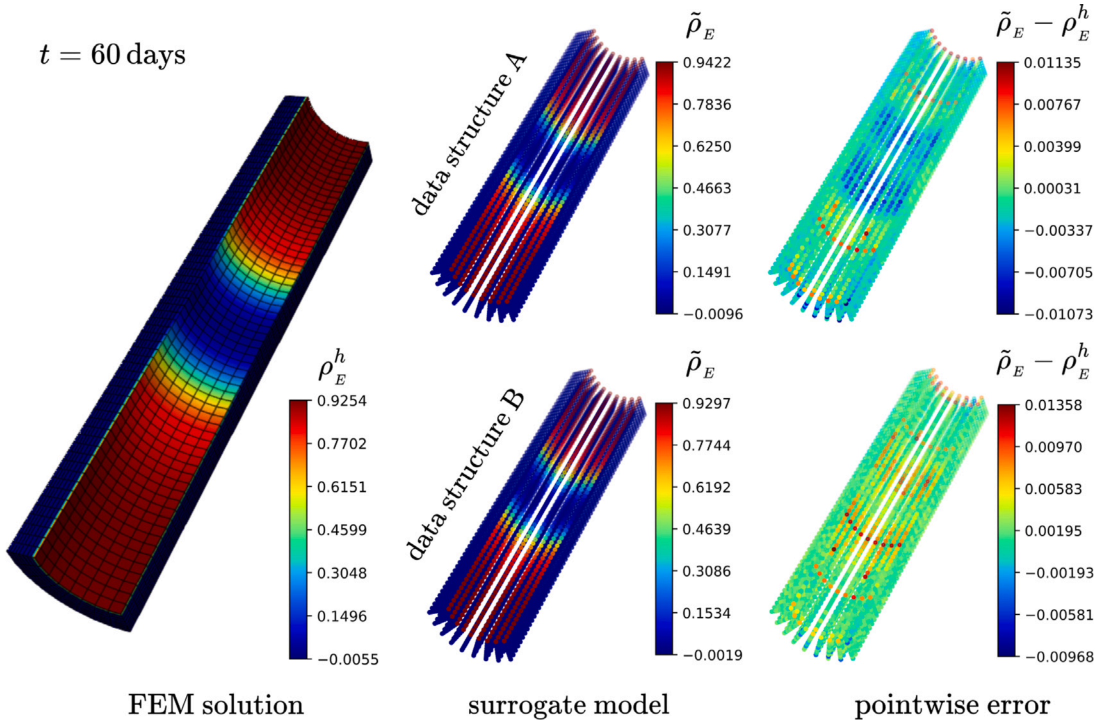


Fig. 6. Comparison of FOM and CAE-ROM solutions for the EC density ρ_E for the testing-parameter instance $\mu_{\text{test}} = (q_D^{\text{ref}} = 7.0\,\text{fmol/mm}^2/\text{day}, t = 60\,\text{days})$.

A scatter plot depicting the drug dosage in terms of the peak value drug flux q_D^{ref} against the degree of ISR is shown in Fig. 9. The plot reveals a clear nonlinearity in the temporal evolution of the degree of ISR. Specifically, during the period of 30–60 days after PCI, the neointimal growth rate is considerably faster than in subsequent periods, indicating significant neointimal evolution within this period. Moreover, the dependence of the degree of ISR on the peak value drug flux q_D^{ref} is also nonlinear. For lower dosages ($<11\,\text{fmol/mm}^2/\text{day}$), the ISR rate appears more irregular. However, in the range of $10-50\,\text{fmol/mm}^2/\text{day}$, it exhibits some periodic patterns. Notably, optimal values are identified in some specific ranges, including $11-13$, $16-17$, $22-24$, $30-32$, and $40-50\,\text{fmol/mm}^2/\text{day}$. These findings emphasize the

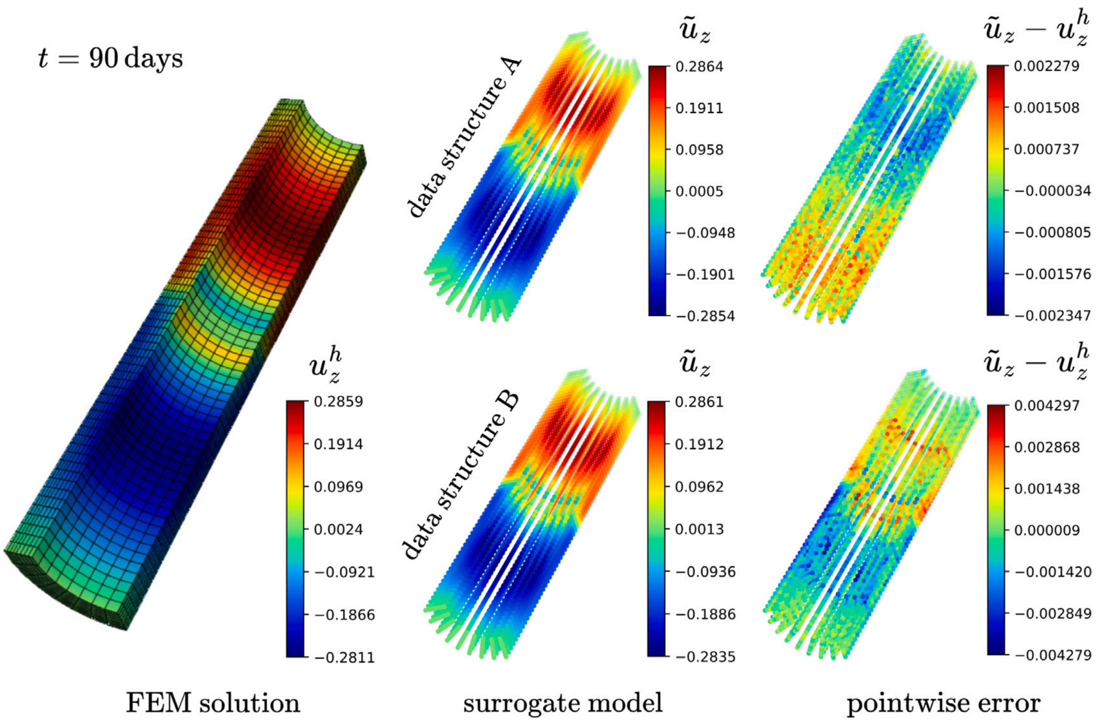


Fig. 7. Comparison of FOM and CAE-ROM solutions for the displacement field u_z for the testing-parameter instance $\boldsymbol{\mu}_{\text{test}} = (q_p^{\text{ref}} = 7.0\,\text{fmol/mm}^2/\text{day}, t = 90\,\text{days})$.

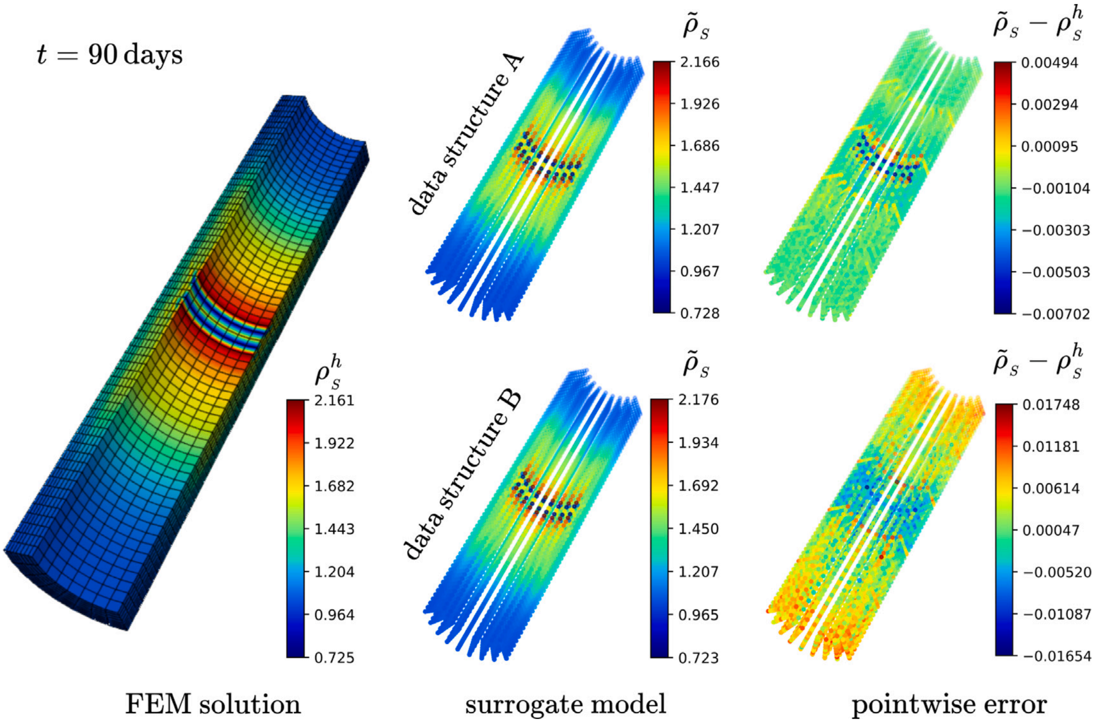


Fig. 8. Comparison of FOM and CAE-ROM solutions for the SMC density ρ_s for the testing-parameter instance $\boldsymbol{\mu}_{\text{test}} = (q_p^{\text{ref}} = 7.0\,\text{fmol/mm}^2/\text{day}, t = 90\,\text{days})$.

nonlinear nature of the interplay between drug dosage and biological mediators, shedding light on potential optimal ranges for therapeutic interventions.

5. Conclusion

In this work, we proposed a novel non-intrusive data-driven ROM approach for the intricate dynamics of coronary ISR, a challenging

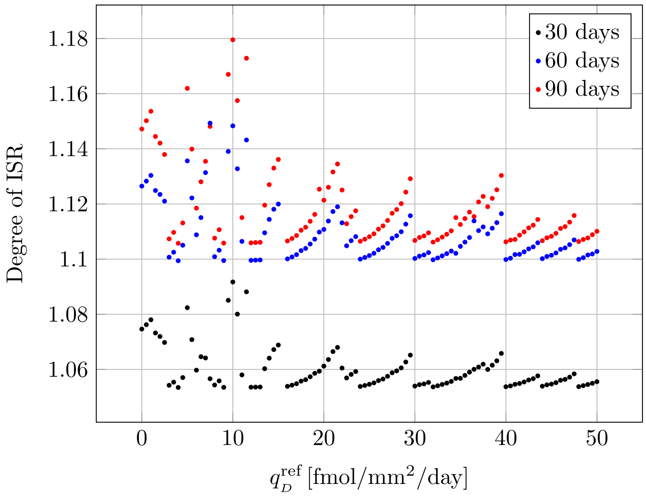


Fig. 9. Estimation of the degree of ISR using the trained CAE-ROM with data type A.

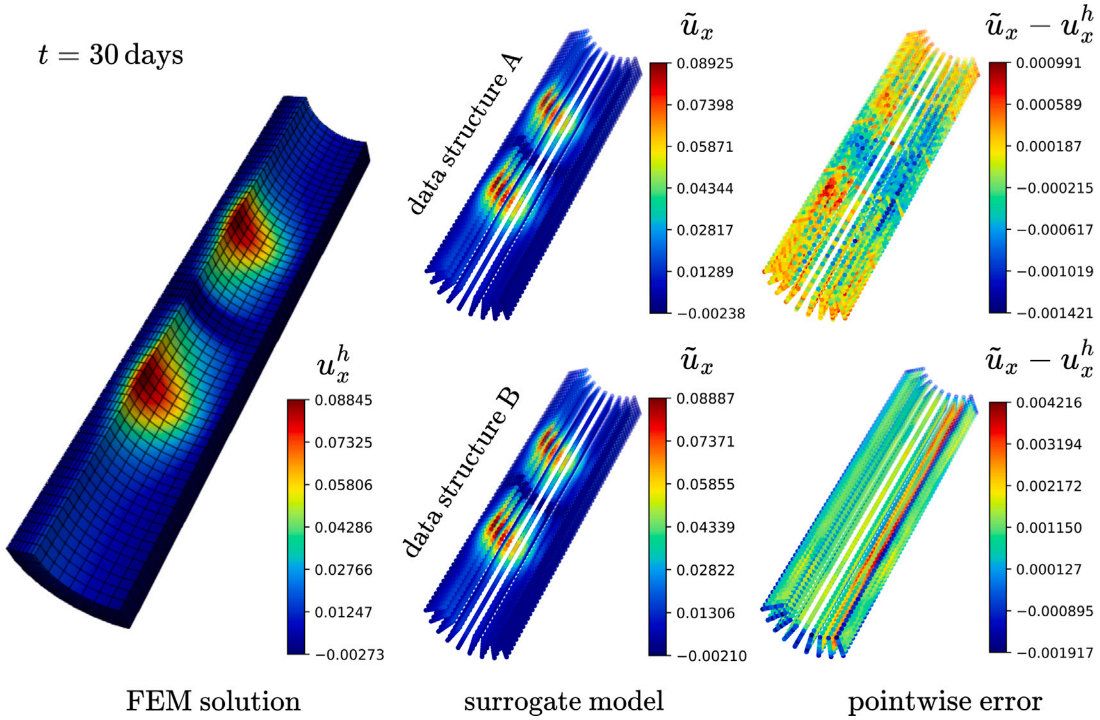


Fig. A.10. Comparison of FOM and CAE-ROM solutions for the displacement field u_x for the testing-parameter instance $\boldsymbol{\mu}_{\text{test}} = (q_D^{\text{ref}} = 7.0\,\text{fmol/mm}^2/\text{day}, t = 30\,\text{days})$.

multiphysics time-dependent parametrized problem. In our methodology, the dimensionality reduction was effectively achieved by applying a 3D convolutional autoencoder. Two distinct approaches were employed to handle the complex 5D input datasets, which encompass 3D geometry, time series, and multiple output channels. The comparative study demonstrates that both approaches exhibit high accuracy in pointwise error, with the first approach showcasing smaller errors across the entire evaluation period for all outputs. The parameter study on drug dosage against ISR rates provides noteworthy insights, revealing a significant acceleration in neointimal growth between 30 – 60 days post PCI. Furthermore, the nonlinear dependence of ISR rates on the peak drug flux highlights intriguing periodic patterns within the dosage range of 10 to 50 fmol/mm^2/day. Optimal values are identified in specific dosage ranges, providing valuable insights for therapeutic treatment. The demonstrated non-intrusive reduced order surrogate model proves to be a powerful tool for predicting ISR outcomes. The straightforward investigation on other input parameters enables full consideration of patient-specific scenarios. Moreover, the proposed method lays the foundation for real-time simulations and optimization of PCI parameters.

We note that one limitation of our current model is that its efficiency and accuracy heavily rely on structured datasets. While CAEs can also handle unstructured data through reshaping, this process often results in the loss of critical information related to geometry and data connectivity, particularly in such intricate patient-specific scenarios. Additionally, reshaping can reduce the CAE's ability to learn effectively from the data, often requiring a significantly larger dataset to compensate. To address this challenge, the incorporation of geometric deep learning, for instance, graph neural networks (GNN), is proposed to better capture and utilize geometrical information for complex geometries with unstructured mesh [32,54,55]. Continued research and development in these areas are necessary to unlock the full

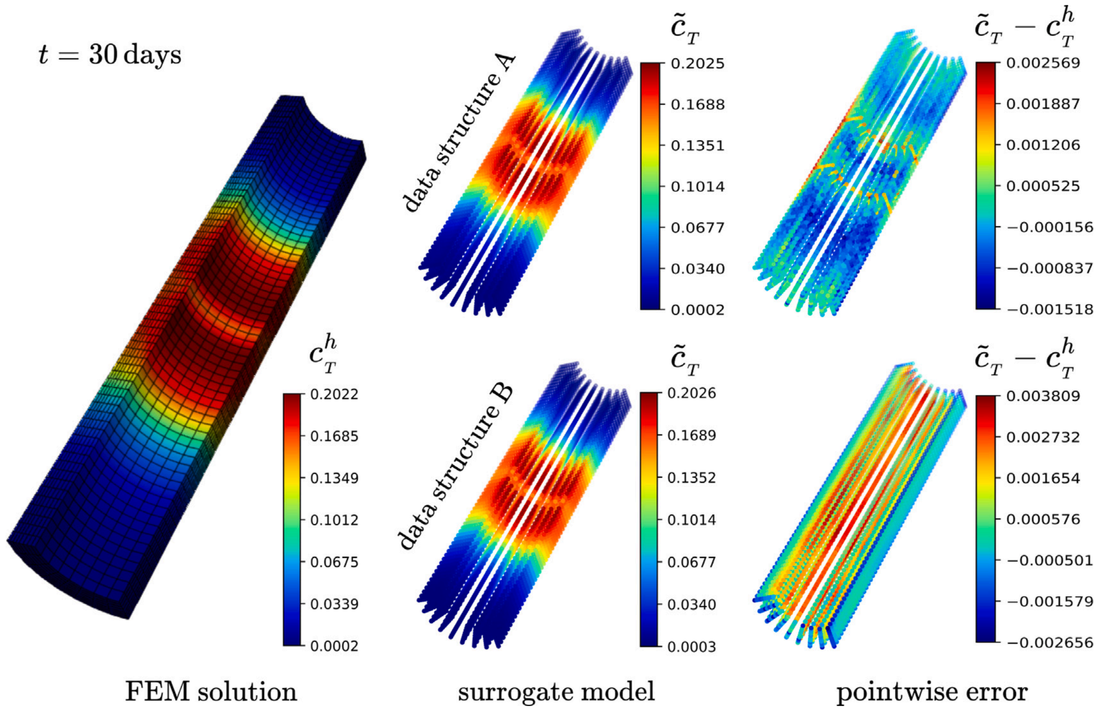


Fig. A.11. Comparison of FOM and CAE-ROM solutions for the TGF concentration c_T for the testing-parameter instance $\mu_{\text{test}} = (q_D^{\text{ref}} = 7.0\,\text{fmol/mm}^2/\text{day}, t = 30\,\text{days})$.

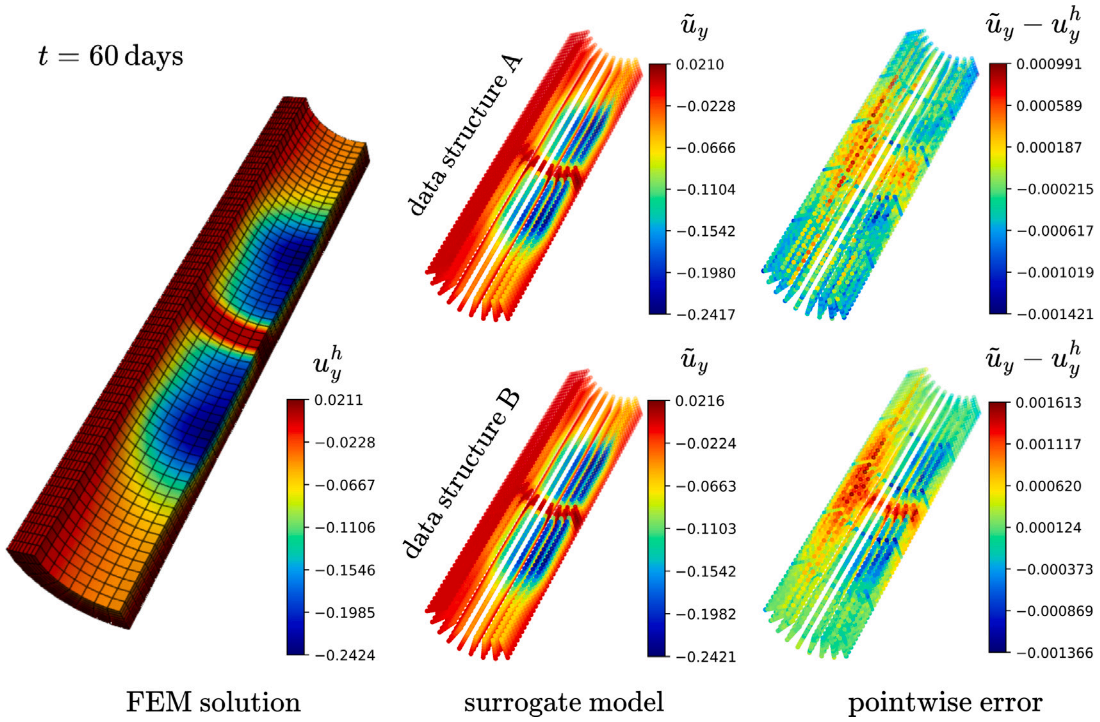


Fig. A.12. Comparison of FOM and CAE-ROM solutions for the displacement field u_y for the testing-parameter instance $\mu_{\text{test}} = (q_D^{\text{ref}} = 7.0\,\text{fmol/mm}^2/\text{day}, t = 60\,\text{days})$.

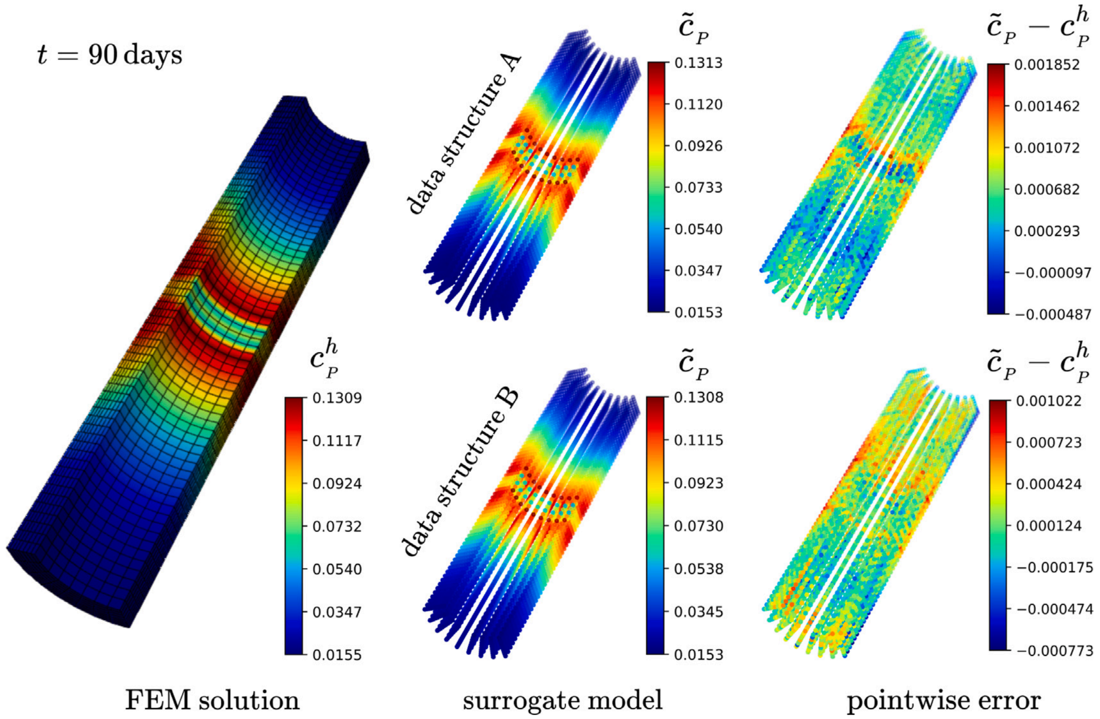


Fig. A.13. Comparison of FOM and CAE-ROM solutions for the PDGF concentration c_P for the testing-parameter instance $\boldsymbol{\mu}_{\text{test}} = (q_D^{\text{ref}} = 7.0\,\text{fmol/mm}^2/\text{day}, t = 90\,\text{days})$.

potential of deep learning-based surrogate modeling for cardiovascular disease.

CRediT authorship contribution statement

Jianye Shi: Writing – review & editing, Writing – original draft, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Kiran Manjunatha:** Writing – review & editing, Formal analysis, Conceptualization. **Felix Vogt:** Writing – review & editing, Methodology, Investigation, Conceptualization. **Stefanie Reese:** Writing – review & editing, Supervision, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the German Research Foundation (DFG) (grant numbers 465213526, 395712048, 403471716). Jianye Shi would like to extend his appreciation for the computational resource provided by the RWTH GPU Cluster.

Appendix. Accuracy probe for other outputs

See Figs. A.10–A.13.

References

[1] K. Manjunatha, M. Behr, F. Vogt, S. Reese, A multiphysics modeling approach for in-stent restenosis: Theoretical aspects and finite element implementation, Comput. Biol. Med. 150 (2022) 106166, http://dx.doi.org/10.1016/j.compbiomed.2022.106166, URL https://www.sciencedirect.com/science/article/pii/S0010482522008745.

[2] K. Manjunatha, N. Schaaps, M. Behr, F. Vogt, S. Reese, Computational modeling of in-stent restenosis: Pharmacokinetic and pharmacodynamic evaluation, Comput. Biol. Med. 167 (2023) 107686, http://dx.doi.org/10.1016/j.compbiomed.2023.107686.

[3] J. Shi, K. Manjunatha, M. Behr, F. Vogt, S. Reese, A physics-informed deep learning framework for modeling of coronary in-stent restenosis, Biomech. Model. Mechanobiol. (2014) http://dx.doi.org/10.1007/s10237-023-01796-1, URL https://doi.org/10.1007/s10237-023-01796-1.

[4] J. Hesthaven, S. Ubbiali, Non-intrusive reduced order modeling of nonlinear problems using neural networks, J. Comput. Phys. 363 (2018) 55–78, http://dx.doi.org/10.1016/j.jcp.2018.02.037, URL https://www.sciencedirect.com/science/article/pii/S0021999118301190.

[5] P. Benner, S. Gugercin, K. Willcox, A survey of projection-based model reduction methods for parametric dynamical systems, SIAM Rev. 57 (4) (2015) 483–531, http://dx.doi.org/10.1137/130932715, arXiv:https://doi.org/10.1137/130932715.

[6] J. Duan, J.S. Hesthaven, Non-intrusive data-driven reduced-order modeling for time-dependent parametrized problems, J. Comput. Phys. 497 (2024) 112621, http://dx.doi.org/10.1016/j.jcp.2023.112621, URL https://www.sciencedirect.com/science/article/pii/S0021999123007167.

[7] S. Ritzert, D. Macek, J.-W. Simon, S. Reese, An adaptive model order reduction technique for parameter-dependent modular structures, Comput. Mech. (2023) http://dx.doi.org/10.1007/s00466-023-02404-w.

[8] R. Halder, K.J. Fidkowski, K.J. Maki, Non-intrusive reduced-order modeling using convolutional autoencoders, Internat. J. Numer. Methods Engrg. 123 (21) (2022) 5369–5390, http://dx.doi.org/10.1002/nme.7072, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.7072 URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.7072.

[9] K. Lee, K.T. Carlberg, Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders, J. Comput. Phys. 404 (2020) 108973, http://dx.doi.org/10.1016/j.jcp.2019.108973, URL https://www.sciencedirect.com/science/article/pii/S0021999119306783.

[10] M. Barrault, Y. Maday, N.C. Nguyen, A.T. Patera, An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations, C. R. Math. 339 (9) (2004) 667–672, http://dx.doi.org/10.1016/j.crma.2004.08.006, URL https://www.sciencedirect.com/science/article/pii/S1631073X04004248.

[11] S. Chaturantabut, D.C. Sorensen, Nonlinear model reduction via discrete empirical interpolation, SIAM J. Sci. Comput. 32 (5) (2010) 2737–2764, http://dx.doi.org/10.1137/090766498.

[12] F. Negri, A. Manzoni, D. Amsallem, Efficient model reduction of parametrized systems by matrix discrete empirical interpolation, J. Comput. Phys. 303 (2015) 431–454, http://dx.doi.org/10.1016/j.jcp.2015.09.046, URL https://www.sciencedirect.com/science/article/pii/S0021999115006543.

[13] D. Amsallem, C. Farhat, Interpolation method for adapting reduced-order models and application to aeroelasticity, AIAA J. 46 (7) (2008) 1803–1813, http://dx.doi.org/10.2514/1.35374, arXiv:https://doi.org/10.2514/1.35374.

[14] D. Amsallem, M.J. Zahr, C. Farhat, Nonlinear model order reduction based on local reduced-order bases, Internat. J. Numer. Methods Engrg. 92 (10) (2012) 891–916, http://dx.doi.org/10.1002/nme.4371, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.4371 URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.4371.

[15] P. Astrid, S. Weiland, K. Willcox, T. Backx, Missing point estimation in models described by proper orthogonal decomposition, IEEE Trans. Autom. Control 53 (10) (2008) 2237–2251, http://dx.doi.org/10.1109/TAC.2008.2006102.

[16] K. Carlberg, C. Farhat, J. Cortial, D. Amsallem, The GNAT method for nonlinear model reduction: Effective implementation and application to computational fluid dynamics and turbulent flows, J. Comput. Phys. 242 (2013) 623–647, http://dx.doi.org/10.1016/j.jcp.2013.02.028, URL https://www.sciencedirect.com/science/article/pii/S0021999113001472.

[17] C. Farhat, T. Chapman, P. Avery, Structure-preserving, stability, and accuracy properties of the energy-conserving sampling and weighting method for the hyper reduction of nonlinear finite element dynamic models, Internat. J. Numer. Methods Engrg. 102 (5) (2015) 1077–1110, http://dx.doi.org/10.1002/nme.4820, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.4820 URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.4820.

[18] J. Maierhofer, D.J. Rixen, Model order reduction using hyperreduction methods (DEIM, ECSW) for magnetodynamic FEM problems, Finite Elem. Anal. Des. 209 (2022) 103793, http://dx.doi.org/10.1016/j.finel.2022.103793, URL https://www.sciencedirect.com/science/article/pii/S0168874X22000671.

[19] D. Xiao, P. Yang, F. Fang, J. Xiang, C. Pain, I. Navon, Non-intrusive reduced order modelling of fluid–structure interactions, Comput. Methods Appl. Mech. Engrg. 303 (2016) 35–54, http://dx.doi.org/10.1016/j.cma.2015.12.029, URL https://www.sciencedirect.com/science/article/pii/S0045782516300068.

[20] R. Chakir, J. Hammond, A non-intrusive reduced basis method for elastoplasticity problems in geotechnics, J. Comput. Appl. Math. 337 (2018) 1–17, http://dx.doi.org/10.1016/j.cam.2017.12.044, URL https://www.sciencedirect.com/science/article/pii/S0377042718300128.

[21] Z. Wang, D. Xiao, F. Fang, R. Govindan, C.C. Pain, Y. Guo, Model identification of reduced order fluid dynamics systems using deep learning, Internat. J. Numer. Methods Fluids 86 (4) (2018) 255–268, http://dx.doi.org/10.1002/fld.4416, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/fld.4416 URL https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.4416.

[22] O. San, R. Maulik, Neural network closures for nonlinear model order reduction, Adv. Comput. Math. 44 (6) (2018) 1717–1750, http://dx.doi.org/10.1007/s10444-018-9590-z.

[23] S. Pawar, S.M. Rahman, H. Vaddireddy, O. San, A. Rasheed, P. Vedula, A deep learning enabler for nonintrusive reduced order modeling of fluid flows, Phys. Fluids 31 (8) (2019) 085101, http://dx.doi.org/10.1063/1.5113494, arXiv:https://pubs.aip.org/aip/pof/article-pdf/doi/10.1063/1.5113494/14800114/085101_1_online.pdf.

[24] T. Murata, K. Fukami, K. Fukagata, Nonlinear mode decomposition with convolutional neural networks for fluid dynamics, J. Fluid Mech. 882 (2020) A13, http://dx.doi.org/10.1017/jfm.2019.822.

[25] P. Wu, J. Sun, X. Chang, W. Zhang, R. Arcucci, Y. Guo, C.C. Pain, Data-driven reduced order model with temporal convolutional neural network, Comput. Methods Appl. Mech. Engrg. 360 (2020) 112766, http://dx.doi.org/10.1016/j.cma.2019.112766, URL https://www.sciencedirect.com/science/article/pii/S0045782519306589.

[26] S. Fresca, L. Dede', A. Manzoni, A comprehensive deep learning-based approach to reduced order modeling of nonlinear time-dependent parametrized PDEs, J. Sci. Comput. 87 (2) (2021) http://dx.doi.org/10.1007/s10915-021-01462-7.

[27] N.T. Mücke, S.M. Bohté, C.W. Oosterlee, Reduced order modeling for parameterized time-dependent PDEs using spatially and memory aware deep learning, J. Comput. Sci. 53 (2021) 101408, http://dx.doi.org/10.1016/j.jocs.2021.101408, URL https://www.sciencedirect.com/science/article/pii/S1877750321000934.

[28] P. Pant, R. Doshi, P. Bahl, A. Barati Farimani, Deep learning for reduced order modelling and efficient temporal evolution of fluid simulations, Phys. Fluids 33 (10) (2021) 107101, http://dx.doi.org/10.1063/5.0062546, arXiv:https://pubs.aip.org/aip/pof/article-pdf/doi/10.1063/5.0062546/16118714/107101_1_online.pdf.

[29] S. Nikolopoulos, I. Kalogeris, V. Papadopoulos, Non-intrusive surrogate modeling for parametrized time-dependent partial differential equations using convolutional autoencoders, Eng. Appl. Artif. Intell. 109 (2022) 104652, http://dx.doi.org/10.1016/j.engappai.2021.104652, URL https://www.sciencedirect.com/science/article/pii/S0952197621004541.

[30] A. Abdedou, A. Soulaimani, Reduced-order modeling for stochastic large-scale and time-dependent flow problems using deep spatial and temporal convolutional autoencoders, Adv. Model. Simul. Eng. Sci. 10 (1) (2023) 7, http://dx.doi.org/10.1186/s40323-023-00244-0.

[31] S. Lee, S. Lee, K. Jang, H. Cho, S. Shin, Data-driven nonlinear parametric model order reduction framework using deep hierarchical variational autoencoder, Eng. Comput. (2024) http://dx.doi.org/10.1007/s00366-023-01916-6.

[32] F. Pichi, B. Moya, J.S. Hesthaven, A graph convolutional autoencoder approach to model order reduction for parametrized PDEs, J. Comput. Phys. 501 (2024) 112762, http://dx.doi.org/10.1016/j.jcp.2024.112762, URL https://www.sciencedirect.com/science/article/pii/S0021999124000111.

[33] R. Maulik, B. Lusch, P. Balaprakash, Reduced-order modeling of advection-dominated systems with recurrent neural networks and convolutional autoencoders, Phys. Fluids 33 (3) (2021) 037106, http://dx.doi.org/10.1063/5.0039986, arXiv:https://pubs.aip.org/aip/pof/article-pdf/doi/10.1063/5.0039986/15666155/037106_1_online.pdf.

[34] S. Dutta, M.W. Farthing, E. Perracchione, G. Savant, M. Putti, A greedy non-intrusive reduced order model for shallow water equations, J. Comput. Phys. 439 (2021) 110378, http://dx.doi.org/10.1016/j.jcp.2021.110378, URL https://www.sciencedirect.com/science/article/pii/S0021999121002734.

[35] S. Lee, K. Jang, S. Lee, H. Cho, S. Shin, Parametric model order reduction by machine learning for fluid–structure interaction analysis, Eng. Comput. (2023) http://dx.doi.org/10.1007/s00366-023-01782-2.

[36] C. Dupont, F. De Vuyst, A.-V. Salsac, Data-driven kinematics-consistent model-order reduction of fluid–structure interaction problems: application to deformable microcapsules in a Stokes flow, J. Fluid Mech. 955 (2023) A2, http://dx.doi.org/10.1017/jfm.2022.1005.

[37] J. Kneifl, D. Grunert, J. Fehr, A nonintrusive nonlinear model reduction method for structural dynamical problems based on machine learning, Internat. J. Numer. Methods Engrg. 122 (17) (2021) 4774–4786, http://dx.doi.org/10.1002/nme.6712, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.6712 URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.6712.

[38] S. Fresca, G. Gobat, P. Fedeli, A. Frangi, A. Manzoni, Deep learning-based reduced order models for the real-time simulation of the nonlinear dynamics of microstructures, Internat. J. Numer. Methods Engrg. 123 (20) (2022) 4749–4777, http://dx.doi.org/10.1002/nme.7054, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.7054 URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.7054.

[39] T. Kadeethum, F. Ballarin, Y. Choi, D. O'Malley, H. Yoon, N. Bouklas, Non-intrusive reduced order modeling of natural convection in porous media using convolutional autoencoders: Comparison with linear subspace techniques, Adv. Water Resour. 160 (2022) 104098, http://dx.doi.org/10.1016/j.advwatres.2021.104098, URL https://www.sciencedirect.com/science/article/pii/S0309170821002499.

[40] J. Kneifl, D. Rosin, O. Avci, O. Röhrle, J. Fehr, Low-dimensional data-based surrogate model of a continuum-mechanical musculoskeletal system based on non-intrusive model order reduction, Arch. Appl. Mech. 93 (9) (2023) 3637–3663, http://dx.doi.org/10.1007/s00419-023-02458-5.

[41] S. Fresca, A. Manzoni, L. Dedè, A. Quarteroni, Deep learning-based reduced order models in cardiac electrophysiology, PLOS ONE 15 (10) (2020) 1–32, http://dx.doi.org/10.1371/journal.pone.0239416.

[42] S. Fresca, A. Manzoni, L. Dedè, A. Quarteroni, POD-enhanced deep learning-based reduced order models for the real-time simulation of cardiac electrophysiology in the left atrium, Front. Phys. 12 (2021) http://dx.doi.org/10.3389/fphys.2021.679076, URL https://www.frontiersin.org/articles/10.3389/fphys.2021.679076.

[43] H. Eivazi, H. Veisi, M.H. Naderi, V. Esfahanian, Deep neural networks for nonlinear model order reduction of unsteady flows, Phys. Fluids 32 (10) (2020) 105104, http://dx.doi.org/10.1063/5.0020526.

[44] M.S. Kim, L.S. Dean, In-stent restenosis, Cardiovasc. Therap. 29 (3) (2011) 190–198, http://dx.doi.org/10.1111/j.1755-5922.2010.00155.x.

[45] S. Reese, K. Manjunatha, J. Shi, M. Sesa, Multiphysical modeling of soft tissue-stent interaction, in: Conference: 10th Edition of the International Conference on Computational Methods for Coupled Problems in Science and Engineering, 2023, http://dx.doi.org/10.23967/c.coupled.2023.023.

[46] J. Shi, K. Manjunatha, S. Reese, Deep learning-based surrogate modeling of coronary in-stent restenosis, Proc. Appl. Math. Mech. (2023) e202300090, http://dx.doi.org/10.1002/pamm.202300090.

[47] A. Barilli, R. Visigalli, R. Sala, G.C. Gazzola, A. Parolari, E. Tremoli, S. Bonomini, A. Simon, E.I. Closs, V. Dall'Asta, O. Bussolati, In human endothelial cells rapamycin causes mTORC2 inhibition and impairs cell viability and function, Cardiovasc. Res. 78 (3) (2008) 563–571, http://dx.doi.org/10.1093/cvr/cvn024.

[48] J. Korelc, Automation of primal and sensitivity analysis of transient coupled problems, Comput. Mech. 44 (5) (2009) 631–649, http://dx.doi.org/10.1007/s00466-009-0395-2.

[49] O. Schenk, K. Gärtner, W. Fichtner, A. Stricker, PARDISO: a high-performance serial and parallel sparse linear solver in semiconductor device simulation, Future Gener. Comput. Syst. 18 (1) (2001) 69–78, http://dx.doi.org/10.1016/S0167-739X(00)00076-5, URL https://www.sciencedirect.com/science/article/pii/S0167739X00000765 I. High Performance Numerical Methods and Applications. II. Performance Data Mining: Automated Diagnosis, Adaption, and Optimization.

[50] M.D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, 2013, arXiv:1311.2901.

[51] M.D. Zeiler, D. Krishnan, G.W. Taylor, R. Fergus, Deconvolutional networks, in: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010, pp. 2528–2535, http://dx.doi.org/10.1109/CVPR.2010.5539957.

[52] H. Noh, S. Hong, B. Han, Learning deconvolution network for semantic segmentation, in: 2015 IEEE International Conference on Computer Vision, ICCV, IEEE Computer Society, Los Alamitos, CA, USA, 2015, pp. 1520–1528, http://dx.doi.org/10.1109/ICCV.2015.178, URL https://doi.ieeecomputersociety.org/10.1109/ICCV.2015.178.

[53] V. Dumoulin, F. Visin, A guide to convolution arithmetic for deep learning, 2018, arXiv:1603.07285.

[54] A. Gruber, M. Gunzburger, L. Ju, Z. Wang, A comparison of neural network architectures for data-driven reduced-order modeling, Comput. Methods Appl. Mech. Engrg. 393 (2022) 114764, http://dx.doi.org/10.1016/j.cma.2022.114764, URL https://www.sciencedirect.com/science/article/pii/S004578252200113X.

[55] S. Barwey, V. Shankar, V. Viswanathan, R. Maulik, Multiscale graph neural network autoencoders for interpretable scientific machine learning, J. Comput. Phys. 495 (2023) 112537, http://dx.doi.org/10.1016/j.jcp.2023.112537, URL https://www.sciencedirect.com/science/article/pii/S0021999123006320.