Received: 14 May 2024 | Revised: 1 October 2024 | Accepted: 17 October 2024

DOI: 10.1002/pamm.202400067

RESEARCH ARTICLE

Fast simulation of coronary in-stent restenosis: A non-intrusive data-driven reduced order surrogate model

Jianye Shi | **Kiran Manjunatha** | **Stefanie Reese**

Institute of Applied Mechanics (IFAM), RWTH Aachen University, Aachen, Germany

Correspondence
Jianye Shi, Institute of Applied Mechanics (IFAM), RWTH Aachen University, Mies-van-der-Rohe-Str. 1, 52074 Aachen Germany.
Email: jianye.shi@ifam.rwth-aachen.de

Funding information
German Research Foundation, Grant/Award Numbers: 465213526, SPP2311

Abstract

Modeling and simulation of coronary artery disease (CAD) is of great importance for supporting and predicting the outcome of percutaneous coronary intervention (PCI). However, an in silico model generally requires heavy computational resources. An effective reduced order surrogate model is indispensable in this context. This study aims to develop a non-intrusive data-driven reduced order surrogate model for coronary in-stent restenosis (ISR) incorporating anti-inflammatory drugs embedded in the drug-eluting stents. The constitutive model includes a detailed multiphysics approach based on partial differential equations (PDEs), which include descriptions of platelet aggregation, growth-factor release, cellular motility and drug deposition. Dimensionality reduction is carried out based on a 3D convolutional autoencoder, which comprises an encoder and decoder. The former condenses the full-order solution into a lower-dimensional latent space, while the latter recovers the full solution from the latent space. Special attention is paid to handle the multidimensional outputs and network architecture.

KEYWORDS
convolutional autoencoder, data-driven modeling, in-stent restenosis, reduced order modeling, surrogate model

1 | MODELING OF IN-STENT RESTENOSIS

1.1 | Evolution of biological mediators and drug concentration

The pathophysiology of in-stent restenosis (ISR) includes the complex interplay between different driving mediators including platelet-derived growth factor (PDGF), transforming growth factor (TGF)-β, extracellular matrix (ECM), smooth muscle cells (SMCs), endothelial cells (ECs), and drug concentration.

PDGF, a family of proteins interconnected by disulfide bonds, plays an important role in vascular remodeling processes, particularly in neointimal hyperplasia following arterial wall injury due to their dual nature as both mitogens and chemoattractants. TGF-β, a group of growth factors composing homodimeric or heterodimeric polypeptides, exhibits multifaceted regulatory properties, which plays a crucial role in cell proliferation, differentiation, and apoptosis. Within

the arterial wall, ECM, primarily composed of collagen, serves as a structural scaffold, initiating essential biochemical and biomechanical signals necessary for tissue morphogenesis, cell differentiation, and homeostasis. SMCs, also known as myocytes, are highly significant cellular constituents of the arterial wall responsible for modulating vascular resistance and blood flow modulation [1]. ECs act as a non-permeable barrier against blood flow interacting with subintimal cellular and extracellular constituents. During stent implantation, the endothelial layer is stripped away, exposing the ECM and SMCs to blood flow. The diminished EC population impedes the production of nitric oxide (NO) and prostacyclin, reducing the anti-platelet activity of remaining ECs. Consequently, platelets and fibrinogen are swiftly deposited at sites of endothelial denudation, instigating the vessel wall's inflammatory response, involving cytokines, chemokines, and cellular adhesion molecules [2]. Drug-eluting stents are widely employed to mitigate the inflammatory response triggered by factors such as endothelial monolayer denudation post-stent placement [3]. Therefore, it is crucial to take into account the evolution of drug concentration within the arterial wall.

To include mediator transport and interaction, the set of advection-reaction-diffusion equations developed in our previous research is adopted. Here, we provide a concise overview of these equations without extensive explanation. Interested readers seeking comprehensive understanding of equation derivation and related parameters are encouraged to refer to our previous work [2, 4–6].

The mediators PDGF, TGF $-\beta$, ECM, SMCs, ECs, and the drug concentration are denoted by c_P, c_T, c_C, ρ_S, ρ_E, and c_D, respectively. Note that the quantities c_P, c_T, c_C, and c_D represent the concentration, while ρ_S and ρ_E denote the density.

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

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

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

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

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

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

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

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

The influence of drug concentration on the EC proliferation and apoptosis is controlled by the scaling functions f_{E1} and f_{E2} (Equation 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 (Equations 1–6) and functions (Equations 7 and 8) mentioned above are constant material or fitting parameters.

1.2 | Constitutive growth model

To describe the evolution of neointima, the deformation gradient $\mathbf{F}$ is decomposed in a multiplicative manner into an elastic part and a growth part, that is,

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

with the corresponding polar decomposition of the growth deformation gradient tensor $\mathbf{F}_g = \mathbf{R}_g\mathbf{U}_g$.

It is common to describe the constitutive relation for the growth using the right stretch tensor $\mathbf{U}_g$. Furthermore, the relation between the global right Cauchy Green tensor $\mathbf{C}$ and the intermediate counterpart $\mathbf{C}_*$ can be established according to

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

The generalized structural tensors $\mathbf{H}_1 = \kappa\,\mathbf{I} + (1-3\kappa)\,\mathbf{a}_1^0 \otimes \mathbf{a}_1^0$ and $\mathbf{H}_2 = \kappa\,\mathbf{I} + (1-3\kappa)\,\mathbf{a}_2^0 \otimes \mathbf{a}_2^0$ with the dispersion parameter $0 \le \kappa \le \frac{1}{3}$, the right Cauchy Green tensor $\mathbf{C}$, the intermediate right Cauchy Green tensor $\mathbf{C}_*$, the growth stretch tensor $\mathbf{U}_g$ and the internal variables are used to define the Helmholtz free energy per unit volume in the reference configuration [2]. Vectors $\mathbf{a}_1^0$ and $\mathbf{a}_2^0$ are the directions of the initial collagen fibers defined in the reference configuration. Constitutively, the right growth stretch tensor $\mathbf{U}_g$ is defined as a function of the SMC density ρ_S and takes the form for positions with

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

$$\mathbf{U}_g := \mathbf{I} + \left(\frac{\rho_S^0}{\rho_{S,eq}} - 1\right)\mathbf{n}_g \otimes \mathbf{n}_g, \quad \mathbf{n}_g := \frac{\mathbf{a}_1^0 \times \mathbf{a}_2^0}{\|\mathbf{a}_1^0 \times \mathbf{a}_2^0\|}; \tag{11}$$

- diffuse collagen orientations with dispersion ($\kappa \neq 0$) in general 3D-case:

$$\mathbf{U}_g := \left(\frac{\rho_S^0}{\rho_{S,eq}}\right)^{1/3}\mathbf{I}, \tag{12}$$

where ρ_S^0 is the actual SMC density pulled back to the reference configuration and $\rho_{S,eq}$ is the homeostatic SMC density of a healthy artery. It is noted that the mechanobiological model presented in this study is a one-way coupled model. The soft tissue growth is formulated as the inelastic part of the total deformation gradient, which is a function of the smooth

muscle cell density. While the model captures the general geometrical changes due to growth, it does not include feedback mechanisms such as mechanical loading (e.g., wall shear stress or strain measures) [7, 8] in the evolution equations of the biological agents.

2 | NON-INTRUSIVE DATA-DRIVEN REDUCED ORDER MODEL

Having the high-fidelity numerical model at hand, data-driven non-intrusive reduced order modeling (ROM) can be explored. In this study, we employ a 3D convolutional autoencoder (CAE) to achieve dimensionality reduction.

Generally, an autoencoder (AE) is a neural network (NN) designed to learn the identity mapping: $\boldsymbol{h} : \boldsymbol{x} \mapsto \tilde{\boldsymbol{x}}$, where $\tilde{\boldsymbol{x}}$ approximates $\boldsymbol{x}$, and $\boldsymbol{h} : \mathbb{R}^{n_x} \mapsto \mathbb{R}^{n_x}$. In this context, the AE consists 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 high-dimensional vector/tensor $\boldsymbol{x}$ to low-dimensional codes $\hat{\boldsymbol{x}}$, and a decoder, $\boldsymbol{h}_{\text{dec}} : \hat{\boldsymbol{x}} \mapsto \tilde{\boldsymbol{x}}$ with $\boldsymbol{h}_{\text{enc}} : \mathbb{R}^{n_{\hat{x}}} \mapsto \mathbb{R}^{n_x}$, reconstructing the input $\tilde{\boldsymbol{x}}$ from the low-dimensional code $\hat{\boldsymbol{x}}$. Thus, the complete AE can be represented as

$$\boldsymbol{h} : \boldsymbol{x} \mapsto \boldsymbol{h}_{\text{dec}} \circ \boldsymbol{h}_{\text{enc}}(\boldsymbol{x}). \tag{13}$$

When 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 indicates that the low-dimensional code $\boldsymbol{h}_{\text{enc}}(\boldsymbol{x}^{(i)})$ contains sufficient information for accurately recovering $\boldsymbol{x}^{(i)}$ through the decoder $\boldsymbol{h}_{\text{dec}}$. This property justifies the use of AE in ROM, particularly when dealing with datasets aligned with high-fidelity data like snapshots.

Various NN architectures, including multilayer perceptrons (MLP), can be employed for the encoding–decoding task. In the case of MLP-based AE, the encoder consists of fully connected layers, with the last layer serving as the bottleneck layer capturing compressed representations of high-order systems. These compressed states represent the reduced order states of the input, utilized by the decoder, implemented as another network of fully connected layers.

While MLP-based AE is advantageous for handling one-dimensional vector inputs, especially when local spatial correlations are unnecessary, it may encounter challenges in accuracy and training when dealing with higher-dimensional inputs requiring the capture of local features. Additionally, MLP-based AE functions as a fully black-box model.

In our work, inspired by applications in picture or video processing, we introduce the CAE to address these limitations, specifically targeting higher-dimensional inputs and local feature capture. Convolutional layers are advantageous for handling multi-channel spatially distributed input data and correlating spatial properties efficiently due to the weight-sharing characteristic of convolutional neural networks (CNNs) [9].

In CAEs, unlike MLP-based AE, convolutional layers are used for primary dimensionality reduction. The encoder processes features through a sequence of convolutional layers, leading to a bottleneck layer (latent variable) via a single fully-connected layer. During decoding, transposed convolutions, or deconvolutions, are employed to upsample the bottleneck layer back to the original higher-dimensional space.

To elucidate the 3D NN architecture used in the subsequent section, fundamental operations and notations (e.g., kernel, stride, pooling, channel, hyperparameters, learnable parameters, etc.) employed in convolutional and transposed convolutional layers in 2D are briefly introduced. These principles extend seamlessly to 3D layers without conceptual differences.

In convolutional layers, feature maps (represented as $\mathcal{H}$) are structured into units. Each unit establishes connections with local patches of the feature maps from the preceding layer. These connections are realized through discrete convolutions defined by a set of kernels (denoted as $\mathcal{W}$), followed by a nonlinear activation function and a pooling operation. In a 2D scenario, the feature map at layer l can be conceptualized as a 3-dimensional tensor $\mathcal{H}^l \in \mathbb{R}^{n^l_{\text{chan}} \times n^l_1 \times n^l_2}$. Here, $\mathcal{H}^l_{i,j,k}$ represents a unit within channel i at row j and column k. Meanwhile, the filter kernel is visualized as a 4-dimensional tensor $\mathcal{W}^l \in \mathbb{R}^{n^l_{\text{filter}}} \times n^{l-1}_{\text{chan}} \times k^l_1 \times k^l_2$. Each element $\mathcal{W}^l_{i,j,m,n}$ connects 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. Thus, the mathematical connection between layer $(l-1)$ and layer l can be expressed as follows:

$$\mathcal{H}^l_{i,j,k} = \phi_l\left(\sum_{r,m,n} \mathcal{H}^{l-1}_{r,(j-1)\times s+m,(k-1)\times s+n}\, \mathcal{W}^l_{i,r,m,n} + \mathcal{B}^l_{i,j,k}\right), \tag{14}$$

for all channels i, where $\mathcal{B}^{l}_{i,j,k}$ and ϕ_l represent the bias and activation function, respectively. One crucial hyperparameter in Equation 14 is the stride size s (which may vary for rows and columns), determining 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 biases $\mathcal{B}$ are learnable parameters, while the kernel length $[k_1, k_2]$, the number of filters n_{filter}, and the stride s are hyperparameters. If required, pooling layers (such as average pooling, max pooling, etc.) can be applied to the output after one convolutional operation. This process not only provides a statistical evaluation (average, maximum, etc.) of the neighboring units at specific locations but also downsamples the output.

On the contrary, a deconvolutional layer executes the reverse process of convolution, playing an essential role in constructing decoding layers. Its primary function involves element-wise multiplication of each input value by a filter. Consequently, the filters learned in the deconvolutional layers act as the foundation for reconstructing the shape of the inputs, while considering the desired shape of the output. Similar to convolutional layers, a deconvolutional layer can incorporate multiple filters, and the stacking of several deconvolutional layers facilitates the development of deep architectures in CAEs. Furthermore, following a deconvolutional operation, unpooling layers (such as nearest neighbor, bed of nails, and max unpooling) can be applied to the output, typically resulting in upsampling.

3 | MODEL SETUP, DATA GENERATION, NETWORK ARCHITECTURE, AND TRAINING

The constitutive model outlined in Section 1 serves as the high-fidelity full-order model. To gauge the feasibility and precision of the proposed data-driven non-intrusive approach, we limit the variability to a single design parameter, specifically the maximum influx of drug q_D^{ref} in the Neumann boundary condition for the drug concentration. This deliberate focus on drug-elution dosage is crucial for clinical relevance, as it significantly influences the dynamics of biological mediators. By narrowing our attention to drug-elution dosages, our objective is not only to comprehend their effects but also to pinpoint an optimal value for the peak drug influx, thus minimizing ISR post PCI.

The FE model comprises $64 \times 8 \times 8$ hexahedral elements, where degrees of freedom are allocated for PDGF, TGF-β, ECM, SMCs, and drug concentration along the longitudinal, circumferential, and radial directions, respectively. Additionally, 64×8 quadrilateral elements are assigned for ECs on the lumen surface. Data is collected from the grid points to form the input dataset, although values from Gaussian points can also be utilized. Temporal discretization spans 180 time steps, with each step representing a day. Consequently, the dataset for one parameter assumes the form of a 4D tensor with dimensions $(9, 9, 65, 9)$, where the tuple denote radial, circumferential, longitudinal sizes, and the number of output channels (9 in total: u_x, u_y, u_z, c_p, c_T, c_C, ρ_S, ρ_E, and c_D), respectively. To establish the parameter space, 50 instances are generated for q_D^{ref}, evenly distributed from 0 to 50, $\text{fmol}/\text{mm}^2/\text{day}$.

For processing the total input datasets, time is treated as an additional parameter, and 3D convolution is applied to individual time steps, creating distinct latent variables for each parameter instance at every time step. Before inputting into the neural network, the high-fidelity data undergoes normalization to optimize trainability. Figure 1 depicts the CAE network architecture.

The comprehensive network architecture for the CAEs is outlined in Table 1. In particular, the encoder integrates zero-padding layers strategically positioned before the convolutional layers to enhance more effective dimensional reduction. Subsequently, multiple 3D convolution layers are employed, each featuring uniform kernel sizes of 9 in all directions, varied strides, and progressively increasing channel dimensions.

During training, a maximum of 10 000 epochs are executed, utilizing a batch size of 10 and a learning rate of $\eta = 10^{-4}$. Early stopping with a patience of 10 is implemented to prevent overfitting. The input data is divided into training and validation sets with an 8:2 ratio. Throughout the network, the ELU (Exponential Linear Unit) non-linear activation function is applied, except for the final convolutional layer in the decoder, where no activation function is utilized. Adam optimizer is employed, and mean squared error (MSE) is used to evaluate the training loss. The NN architecture is constructed using TensorFlow with the Keras API, and the training process is conducted on the RWTH Aachen GPU cluster workstation, equipped with dual NVIDIA Tesla V100 GPUs.

Following training, Gaussian process regression (GPR) is employed to establish correlations between the latent variables and the parameter space. While alternative regression methods like artificial neural networks (ANN) are viable, GPR is selected for its inherent flexibility, particularly in extending correlations between vector-valued inputs and outputs. In

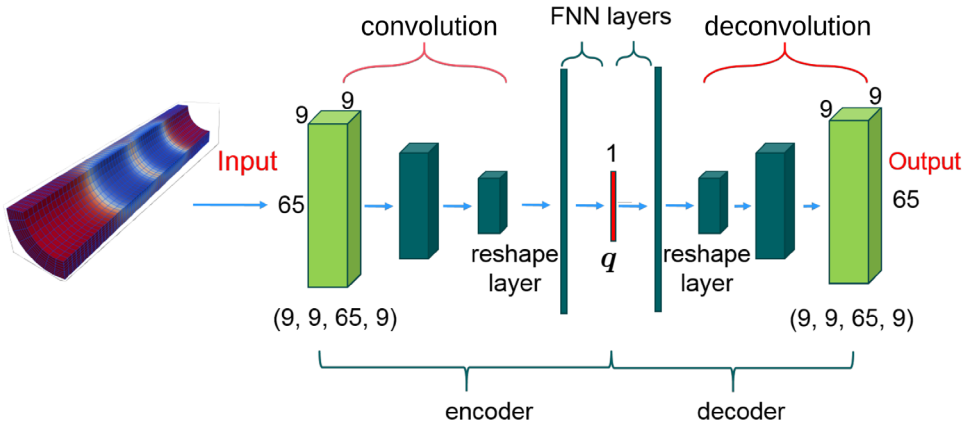

FIGURE 1 Network architecture of the 3D deep CAE. In the downsampling process, the encoder transforms the input into the low-dimensional latent variable $\boldsymbol{q}$ using 3D convolution, FNN layers. Subsequently, in the upsampling process, the decoder reconstructs the high-dimensional solution from the latent variable $\boldsymbol{q}$ applying FNN, reshape/unflatten and 3D transposed convolution layers. CAE, convolutional autoencoder; FNN, flatten and dense/fully connected.

TABLE 1 CAE network architecture.

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	$\dim(\boldsymbol{q}) = 12$	–	–	–
Decoder layer	**Input dimension**	**Output dimension**	**Kernel size**	**Number of filters**	**Strides**
Dense	$\dim(\boldsymbol{q}) = 12$	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)	–	–	–

Abbreviations: CAE, convolutional autoencoder.

this study, GPR implementation utilizes the scikit-learn library and incorporates the radial basis function (RBF) kernel to enhance the model's capability in capturing complex relationships within the data.

4 | RESULTS AND DISCUSSION

To exam the accuracy of the CAE-ROM method, the SMC density 60 days post PCI is compared, as depicted in Figure 2. The global comparison reveals that the proposed ROM demonstrates impressive accuracy. A minor discrepancy is noticeable in

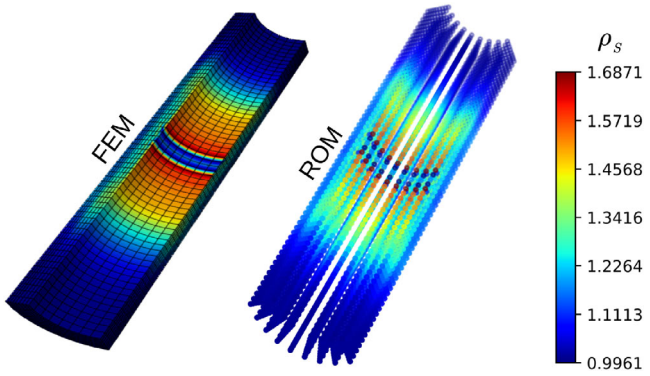

FIGURE 2 Comparison of full-order FEM and CAE-ROM solutions for the SMC density ρ_s 60 days post PCI. CAE, convolutional autoencoder; ROM, reduced order modeling; SMC, smooth muscle cell.

the contact area with the ring-shaped stent. Having the trained and validated surrogate model established, the assessment of new input parameters becomes straightforward.

5 | CONCLUSION

In this study, we introduced an innovative non-intrusive data-driven reduced order modeling approach tailored to the intricate dynamics of coronary in-stent restenosis, a complex multiphysics time-dependent parametrized problem. Our methodology effectively achieved dimensionality reduction with high accuracy by employing a 3D convolutional autoencoder. The straightforward exploration of other input parameters enables comprehensive consideration of patient-specific scenarios. Additionally, the proposed method lays the groundwork for real-time simulations and optimization of PCI parameters. However, it is worth noting a drawback of our current method: the reliance on structured datasets by the applied convolutional neural networks may lead to inefficiencies and inaccuracies when confronted with unstructured domains, such as real intricate patient-specific geometries. To tackle this challenge, we propose the integration of geometric deep learning techniques, such as graph neural networks, to better capture and utilize geometrical information for complex geometries with unstructured meshes.

ACKNOWLEDGMENTS

Financial support provided by the German Research Foundation (DFG) for the subproject "In-stent restenosis in coronary arteries - in silico investigations based on patient-specific data and meta modeling" (project number 465213526) of SPP2311 is gratefully acknowledged.

Open access funding enabled and organized by Projekt DEAL.

ORCID

Jianye Shi https://orcid.org/0000-0002-7855-2695

REFERENCES

1. Manjunatha, K., Behr, M., Vogt, F., & Reese, S. (2022). A multiphysics modeling approach for in-stent restenosis: Theoretical aspects and finite element implementation. *Computers in Biology and Medicine*, *150*, 106166.
2. Manjunatha, K., Schaaps, N., Behr, M., Vogt, F., & Reese, S. (2023). Computational modeling of in-stent restenosis: Pharmacokinetic and pharmacodynamic evaluation. *Computers in Biology and Medicine*, *167*, 107686.
3. Parry, T., Thyagarajan, R., Argentieri, D., Falotico, R., Siekierka, J., & Tallarida, J. T. (2006). Effects of drug combinations on smooth muscle cell proliferation: An isobolographic analysis. *European Journal of Pharmacology*, *532*, 38–43.
4. Reese, S., Manjunatha, K., Shi, J., & Sesa, M. (2023). Multiphysical modeling of soft tissue-stent interaction. *10th edition of the International Conference on Computational Methods for Coupled Problems in Science and Engineering*, Coupled Problems. https://doi.org/10.23967/c.coupled.2023.023
5. Shi, J., Manjunatha, K., Behr, M., Vogt, F., & Reese, S. (2024). A physics-informed deep learning framework for modeling of coronary in-stent restenosis. *Biomechanics and Modeling in Mechanobiology*, *23*, 615–629.

6. Shi, J., Manjunatha, K., & Reese, S. (2023). Deep learning-based surrogate modeling of coronary in-stent restenosis. *PAMM*, *23*, e202300090. https://doi.org/10.1002/pamm.202300090
7. Uhlmann, K., & Balzani, D. (2023). Chemo-mechanical modeling of smooth muscle cell activation for the simulation of arterial walls under changing blood pressure. *Biomechanics and Modeling in Mechanobiology*, *22*, 1049–1065.
8. Meike, G., Peter, W., & Michele, M. (2023). Arterial tissues and their inflammatory response to collagen damage: A continuum in silico model coupling nonlinear mechanics, molecular pathways, and cell behavior. *Computers in Biology and Medicine*, *158*, 106811.
9. Fresca, S., Dede', L., & Manzoni, A. (2021). A comprehensive deep learning-based approach to reduced order modeling of nonlinear time-dependent parametrized PDEs. *Journal of Scientific Computing*, *87*, 61.

How to cite this article: Shi, J., Manjunatha, K., & Reese, S. (2024). Fast simulation of coronary in-stent restenosis: A non-intrusive data-driven reduced order surrogate model. *Proceedings in Applied Mathematics and Mechanics*, e202400067. https://doi.org/10.1002/pamm.202400067