Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 394 (2022) 114871
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

Simulation of the 3D hyperelastic behavior of ventricular myocardium using a finite-element based neural-network approach

Wenbo Zhang[a], David S. Li[a,b], Tan Bui-Thanh[c], Michael S. Sacks[a,b,c,*]

[a] *James T. Willerson Center for Cardiovascular Modeling and Simulation, Oden Institute for Computational Engineering and Sciences, The University of Texas at Austin, 201 East 24th St, Stop C0200, Austin, TX 78712, USA*
[b] *Department of Biomedical Engineering, The University of Texas at Austin, 201 East 24th St, Stop C0200, Austin, TX 78712, USA*
[c] *Department of Aerospace Engineering and Engineering Mechanics, and Oden Institute for Computational Engineering and Sciences, The University of Texas at Austin, 201 East 24th St, Stop C0200, Austin, TX 78712, USA*

Received 25 January 2022; received in revised form 7 March 2022; accepted 9 March 2022
Available online 1 April 2022

Abstract

High-fidelity cardiac models using attribute-rich finite element based models have been developed to a very mature stage. However, such finite-element based approaches remain time consuming, which have limited their clinical use. There remains a need for alternative methods for novel cardiac simulation methods capable of high fidelity simulations in clinically relevant time frames. Surrogate models are one approach, which traditionally use a data-driven approach for training, requiring the generation of a sufficiently large number of simulation results as the training dataset. Alternatively, a physics-informed neural network can be trained by minimizing the PDE residuals or energy potentials. However, this approach does not provide a general method to easily using existing finite element models. To address these challenges, we developed a hybrid approach that seamlessly bridged a neural network surrogate model with a differentiable finite element domain representation (NNFE). Given its importance in cardiac simulations, we applied this approach to simulations of the hyperelastic mechanical behavior of ventricular myocardium from recent 3D kinematic constitutive model (J Mech Behav Biomed Mater, 2020 doi: 10.1016/j.jmbbm.2019.103508). We utilized a cuboidal domain and conducted numerical studies of individual myocardium specimens discretized by a finite element mesh and assigned with experimentally obtained myofiber architectures. Both parameterized Dirichlet and Neumann boundary conditions were studied. We developed a second-order Newton optimization method, instead of using a stochastic gradient descent method, to train the neural network efficiently. The resulting trained neural network surrogate model demonstrated excellent agreement with the corresponding "ground truth" finite element solutions over the entire physiological deformation range. More importantly, the NNFE approach provided a significantly decreased computational time for a range of finite element mesh sizes for online predictions. For example, as the finite element mesh size increased from 2744 to 175615 elements, the NNFE computational time increased from 0.1108 s to 0.1393 s, while the "ground truth" FE model increased from 4.541 s to 719.9 s. These results suggest that NNFE run times can be significantly reduced compared with the traditional large-deformation based finite element solution methods. The trade-off is to train the NNFE off-line within a range of anticipated physiological responses. However, training time would only have to be performed once before any number of application uses. Moreover, since the NNFE is an analytical function its computational performance will be amplified when the corresponding problem becomes more complex.

Keywords: Soft tissues; Machine learning; Surrogate modeling

* Corresponding author.
E-mail address: msacks@oden.utexas.edu (M.S. Sacks).

https://doi.org/10.1016/j.cma.2022.114871
0045-7825/

Nomenclature

$\bar{\mathbf{C}}$	Isochoric part of the right Cauchy–Green tensor
$\bar{I}_1$	First invariant of $\bar{\mathbf{C}}$
$\bar{I}_{4\mathrm{f}}$	Fourth pseudo invariant of $\mathbf{C}$ for the fiber direction
$\bar{I}_{4\mathrm{s}}$	Fourth pseudo invariant of $\mathbf{C}$ for the sheet direction
$\bar{I}_{8\mathrm{fn}}$	Eighth pseudo invariant of $\mathbf{C}$ for fiber–normal interaction
$\bar{I}_{8\mathrm{fs}}$	Eighth pseudo invariant of $\mathbf{C}$ for fiber–sheet interaction
$\bar{I}_{8\mathrm{sn}}$	Eighth pseudo invariant of $\mathbf{C}$ for sheet–normal interaction
Γ_t	Neumann boundary
Γ_u	Dirichlet boundary
$\mathbf{b}$	Neural network biases
$\mathbf{C}$	Right Cauchy–Green tensor
$\mathbf{F}$	Deformation gradient
$\mathbf{f}_0$	Fiber direction in the reference configuration
$\mathbf{h}$	Hidden states of the neural network
$\mathbf{M}$	Input parameters for surrogate model
$\mathbf{N}$	Finite element basis functions
$\mathbf{n}_0$	Normal direction in the reference configuration
$\mathbf{P}$	First Piola–Kirchhoff stress
$\mathbf{s}_0$	Sheet direction in the reference configuration
$\mathbf{T}$	Prescribed traction field on the Neumann boundary
$\mathbf{U}$	Nodal displacements
$\mathbf{u}$	Displacement
$\mathbf{u}_0$	Prescribed displacement on the Dirichlet boundary
$\mathbf{U}_{\mathrm{FE}}$	Finite element solution
$\mathbf{U}_{\mathrm{NN}}$	Neural network prediction
$\mathbf{W}$	Neural network weights
$\mathbf{X}$	Coordinates in the reference domain
$\mathbf{x}$	Coordinates in the current domain
$\mathcal{D}_{\mathrm{tr}}$	Training dataset
$\mathcal{D}_{\mathrm{val}}$	Validation dataset
Ω	Current configuration
Ω_0	Reference configuration
ϕ	Activation function
Π	Total potential energy
Ψ	Strain energy density function
Ψ_{f}	Fiber contribution of strain energy density function
Ψ_{int}	Interaction contribution of strain energy density function
Ψ_{m}	Matrix contribution of strain energy density function
Ψ_{vol}	Volumetric contribution of strain energy density function
θ	Neural network parameters
e_{L_∞}	Average L_∞ error of the neural network predictions
J	The Jacobian determinant of the deformation gradient

1. Introduction

The human heart is an inherently complex organ and is often cited as a paradigm of functional efficiency. Nevertheless, it can experience a wide range of pathologies often presented as patient-specific lesions. Imaging based diagnostic methods have made great strides in recent years in the accurate diagnosis of cardiac disease. Yet our ability to predict disease progression and treatment outcomes still rests on our understanding of cardiac function. It is thus of no surprise that computational methods continue to demonstrate great utility in improving our understanding of cardiac pathophysiology and treatment [1–3]. The finite element method (FEM) has been widely used to develop high-fidelity cardiac models as it can easily simulate such important aspects as detailed cardiac geometry at multiple scales (e.g., ventricular and atrial geometry, local myofiber structure), time varying boundary conditions, and related functional aspects [4].

A critical part of developing high-fidelity cardiac models is the accurate simulation of the full three-dimensional (3D) mechanical behavior of the myocardium. This is especially important in the prediction of onset and time progression of cardiac diseases such as response to pulmonary hypertension and left ventricular infarction [5–7]. The development of mathematically rigorous and robust estimators of myocardial mechanical behavior has been the subject of intense research [8–19]. This has been a challenging task as myocardium possesses an unusually complex structural and functional hierarchy. These occur at various length scales and dictate local anisotropic mechanical behaviors, with the spatial variations in fiber structure giving rise to functional heterogeneity [20]. To address these issues in a full 3D context, we have previously developed a novel approach to determine the optimal parameters for candidate continuum constitutive models of myocardium [19,21]. This approach utilized optimal experimental design of the full 3D kinematic approach coupled to an inverse model to perform robust parameter estimation. Our findings indicated a previously unreported coupling behavior via shearing of myofibers and extracellular collagen fibers. This resulted in development of an extended invariant based material model that included additional terms to describe fiber–normal and sheet–normal coupling in the constitutive model. The modified model accurately reproduced all optimal loading paths and exhibited improved predictive capabilities.

Clearly, such approaches are crucial for improved understanding and performance in cardiac modeling in healthy, diseased, and treatment scenarios. However, even with increasingly accurate models of myocardial mechanical behaviors, electromechanical coupling [22–26], and biophysical contraction [27–30]—all integrated into high-fidelity finite element based simulations, they remain computationally too intensive to explore patient outcomes in clinically relevant time-frames. Thus, while high-fidelity cardiac simulations have been developed both in general and for specific aspects (e.g. cardiac mechanics) emerging simulation challenges require development of alternative computational approaches that can provide the requisite simulations with improved computational efficiency.

In considering how to approach this problem, it should be noted that at a fundamental mathematical level all finite element methods based cardiac mechanics simulations attempt to solve the associated non-linear hyperelastic myocardium mechanical behaviors. This involves a parametric map from the input (i.e., the model, boundary conditions, etc.) to the output, via solution of the underlying governing partial differential equations (PDEs). Each evaluation of the parametric map thus involves finding the PDE solution for a given instance of input model parameters. Given the need for a very large number of clinical scenarios for a single patient specific therapy, the need for computational efficiency becomes obvious. There are many potential approaches possible to reduce the computational challenges for high-fidelity cardiac simulations associated with evaluations of the parametric map. Projection-based model reduction methods extract the reduced bases from the snapshots of the simulation results to reduce the complexity of the original methods [31,32]. Alternatively, the parametric map can be approximated by a surrogate model using polynomials [33], Gaussian process models [34,35], or polynomial chaos expansion [36,37]. For high-dimensional problems, Gaussian process and polynomial chaos expansion models are intractable due to the curse of dimensionality. While these approaches are on a rigorous mathematical foundation and have demonstrated promising improvements in computational efficiency, none of these approaches have yet shown the necessary simulation speed improvements required for routine clinical applications.

Continued improvements in computational power, software, and massive data availability over the past two decades have led to an increasing interest in neural networks (NN) as universal function approximators [38]. While commonly associated with image processing and informatics, NN have also been used to directly represent PDE solutions by minimizing the residual norm [39] or via a variational functional [40] using physics-informed neural networks (PINNs). PINNs are also applied to finite deformation hyperelasticity [41]. The relationship with the finite element method has also been studied for rectified linear unit (ReLU) and linear finite elements [42].

Neural networks have been used to approximate the parametric map to form an analytical function when numerous evaluations of the parametric map are required. To this end, utilizing neural networks to reduce the computational cost has a long and rich history in the context of structural analysis and design optimization [43]. This approach, often termed a "neural network surrogate model" (NNSM) for parametric PDEs, can be trained by a data-driven approach or a physics-informed approach [44–49]. A data-driven approach needs the finite element (FE) solutions for a set of realizations of the inputs parameters, collected as the training dataset by solving the corresponding FE equations. Dimension reduction methods can be used to represent the inputs and outputs for the neural network surrogate model in the dominant subspace [46]. Neural networks can also enhance the accuracy of a reduced order model [47].

Alternatively, physics-based approaches seek to *directly* minimize the residual norm or the variational functional of the governing PDE without the need to generate and train on a wide range of FEM solutions. In other words, the physics-based approaches attempt to solve without the labels or the FE solutions. In the present work we utilized a hybrid approach that couples a neural network to a differentiable finite element domain map (a neural network-finite element or NNFE) model to approximate the parametric map. In our approach we essentially shift the computational expense offline, allowing for extremely rapid simulations of passive mechanical behaviors.

The physics-based approach for NNFE can be based on fully-connected networks (FCNs) [50] or convolutional neural networks (CNNs) [45,51]. FCNs directly approximate the PDE solutions while the CNNs use a rectangular grid to discretize the solution field, and have been developed and demonstrate superior capability for capturing local variations of the solution field. However, the potential of CNNs is limited by the regular domain where the convolutional layers are defined. For example, the convolutional operators in 2D are defined for the rectangular domain. The restriction of the CNNs on the regular domain can be alleviated by applying a coordinate map [52]. Since it assumes the reference domain is regular, this limitation is still not fully resolved for a problem defined on a geometry with complex topology. Thus, FCNs remain the NN topological design of choice in the present study. We also note that in the implementation by Zhu et al. [45,51] the differential operators were evaluated based on a smooth finite difference method. In comparison with the finite difference method, the FEM can easily tackle arbitrary complex geometries and boundary conditions. In general, FCNs that directly approximate the solution fields do not need conventional domain discretization. There are also efforts based on PINNs, using non-uniform rational basis spline (NURBS) basis functions to map the reference domain to physical domain [53]. However, this is not an isoparametric mapping approach since the solution field is approximated by a FCN, while the geometry is approximated by NURBS. In the present study, we utilized an isoparametric FE discretization approach. This allowed us to leverage mature finite element meshing tools and quadrature schemes. This approach also facilitated utilization of FE meshes developed from clinical imaging data for cardiac modeling, which is our main application goal. These studies suggest that *physics-based* surrogate modeling approaches are a promising alternative to conventional finite element approaches for many-query problems wherein a minimal application computation time is critical.

As a first step in exploring the feasibility of a NNFE modeling approach for cardiac mechanics simulations, we developed a specialized implementation for the simulation of the passive behavior of myocardium. This was done using a previously developed advanced model material model under generalized 3D deformations, using local myofiber structures to define local material axes [8,19,21,53]. This approach utilized well defined boundary conditions as a means to rigorously evaluate the approach prior to larger scale simulations. Actual specimen-specific material parameters and local myofiber structures were utilized to increase the realism of the simulations, with results compared to "ground truth" FE simulations using identical FE meshed domains. We then addressed several numerical challenges arising from the training algorithms, scaling and transfer learning behaviors, and the effects of varying neural network architectures.

2. Methods

2.1. Overview

We present a comprehensive computational pipeline to develop an efficient neural network finite element based representation of the solutions of hyperelastic PDEs that describe the 3D passive response of myocardium under physiological deformations. We utilized a model developed in [21] for a set of four 1 cm^3 cuboidal specimens, each evaluated using precisely defined fully 3D kinematic boundary conditions, constitutive model form, and parameter

set that integrated local 3D myofiber structural information. This approach provided for a well defined, rich passive myocardium mechanical behavior dataset for NNFE model development. To develop the NNFE approach we utilized a fully-connected neural network. A novel aspect of our NNFE approach is that the neural network output is linked to the problem domain defined over a finite element mesh. The resulting NNFE model was trained by minimizing an energy functional for each of the four specimens' material parameters sets. We then demonstrated the efficacy and robustness of this approach as compared to "ground truth" finite element solutions performed on identical meshed domains. We then demonstrate scalability of the approach, as well as the use of transfer learning to reduce training time.

2.2. 3D hyperelastic constitutive models of myocardium

We start by summarizing our anisotropic hyperelastic material model of myocardium [21]. This approach was based on earlier work by Holzapfel et al. [16,54], and utilized an extended form to fully capture unique coupling phenomena observed in 3D triaxial deformation. First, consider the domain Ω_0 occupied by myocardium with the reference configuration defined as $\mathbf{X}$. We denote the current configuration $\mathbf{x}$ and the displacement $\mathbf{u} = \mathbf{x} - \mathbf{X}$. Then we denote the deformation gradient by $\mathbf{F} = \partial\mathbf{x}/\partial\mathbf{X}$, and the right Cauchy–Green tensor $\mathbf{C} = \mathbf{F}^{\mathrm{T}}\mathbf{F}$, and the Jacobian determinant $J = \det\mathbf{F}$. The isochoric part of the right Cauchy–Green tensor is given by $\bar{\mathbf{C}} = J^{-2/3}\mathbf{C}$. We model the myocardium as a nearly incompressible hyperelastic materials by penalizing the volumetric deformation. To avoid the computationally expensive mixed formulation along with the augmented Lagrangian method to enforce incompressibility, we adapted the formulation in [55] using the invariants of $\bar{\mathbf{C}}$ to describe the isotropic contribution to the strain energy density function and pseudo-invariants of $\mathbf{C}$ to describe the anisotropic contribution to the strain energy density function. Note that in [55] it was demonstrated that the formulations that treat the isotropic part and the anisotropic parts differently can be up to 20 times faster than the augmented Lagrangian method to enforce incompressibility, with only a 5% difference in results from the augmented Lagrangian method. This approach also simplifies the formulation for training and avoids the complexity involving with the augmented Lagrangian method.

Next, we define the first invariant of $\bar{\mathbf{C}}$

$$\bar{I}_1 = \mathrm{Tr}\,\bar{\mathbf{C}}. \tag{1}$$

We denote the material axes of myocardium in the local Cartesian material coordinate system in the reference configuration as the fiber direction $\mathbf{f}_0$, the sheet direction $\mathbf{s}_0$, and the normal direction $\mathbf{n}_0$. Two diagonal components of $\mathbf{C}$ on the material axes are

$$I_{4\mathrm{f}} = \mathbf{f}_0 \cdot \mathbf{C} \cdot \mathbf{f}_0, \quad I_{4\mathrm{s}} = \mathbf{s}_0 \cdot \mathbf{C} \cdot \mathbf{s}_0. \tag{2}$$

The pseudo-invariants $I_{4\mathrm{f}}$ and $I_{4\mathrm{s}}$ represent the square of the stretch for each fiber family. Due to the symmetry of $\mathbf{C}$, we consider three off-diagonal components of $\mathbf{C}$ on the material axes

$$I_{8\mathrm{fs}} = \mathbf{f}_0 \cdot \mathbf{C} \cdot \mathbf{s}_0, \quad I_{8\mathrm{fn}} = \mathbf{f}_0 \cdot \mathbf{C} \cdot \mathbf{n}_0, \quad I_{8\mathrm{sn}} = \mathbf{s}_0 \cdot \mathbf{C} \cdot \mathbf{n}_0. \tag{3}$$

The pseudo-invariants $I_{8\mathrm{fs}}$, $I_{8\mathrm{fn}}$, $I_{8\mathrm{sn}}$ describe the interactions between material axes.

We consider the extracellular matrix (ECM), the fiber families, and their interactions are exponentially stiffening. The strain energy density function of the myocardium is given by

$$\begin{aligned}\Psi &= \Psi_{\mathrm{m}}(\bar{I}_1) + \Psi_{\mathrm{f}}(I_{4\mathrm{f}}, I_{4\mathrm{s}}) + \Psi_{\mathrm{int}}(I_{8\mathrm{fs}}, I_{8\mathrm{fn}}, I_{8\mathrm{sn}}) + \Psi_{\mathrm{vol}}(J) \\ &= \frac{a}{2b}\left(\exp\left(b\left(I_1 - 3\right)\right) - 1\right) + \sum_{i=\mathrm{f,s}} \frac{a_i}{2b_i}\left(\exp\left(b_i\left(I_{4i} - 1\right)^2\right) - 1\right) \\ &\quad + \sum_{i=\mathrm{fs,fn,sn}} \frac{a_i}{2b_i}\left(\exp\left(b_i I_{8i}^2\right) - 1\right) + \kappa\left(J - 1 - \ln J\right).\end{aligned} \tag{4}$$

The first three terms Ψ_{m}, Ψ_{f} and Ψ_{int} correspond to the isotropic contribution, the fiber contribution, the interaction contribution. The last term Ψ_{vol} penalizes the violation of the incompressibility constraint. There are a total of twelve model parameters for each specimen (Table 1), and κ was set to 5000 Pa.

Table 1
Model parameters of Ψ for each specimen simulated.
Source: Taken from [21].

	a (Pa)	b	a_{f} (Pa)	b_{f}	a_{s} (Pa)	b_{s}	a_{fs} (Pa)	b_{fs}	a_{fn} (Pa)	b_{fn}	a_{sn} (Pa)	b_{sn}
S1	4.81	5.05	1175	0.276	1665	6.93	135.5	0.024	3440	2.01	3278	20.4
S2	0.100	0.346	2936	0.045	989.1	0.190	548.6	0.0020	384.0	1.47	6397	0.004
S3	6.07	4.90	2892	0.0300	178.2	0.0500	1553	0.78	4174	40.3	1791	44.6
S4	1.05	12.0	2964	3.10	496.0	0.0870	369.0	0.011	1712	68.8	547.7	1.17

2.3. Parameterization of boundary conditions

As part of our motivation for the development of the NNFE model, we consider application of the many-query problem. In the context of the present problem, we are solving over multiple boundary conditions defined by the parametric PDE associated with the myocardium hyperelasticity problem. In order to define and perform the necessary computations, we discretized the reference domain of the cuboidal body Ω_0 using a finite element mesh with d_n nodes. This allowed the displacement field $\mathbf{u}(\mathbf{X}) : \Omega_0 \mapsto \mathbb{R}^3$ to be conventionally represented by the associated finite element basis functions

$$u_i(\mathbf{X}) = \sum_{A=1}^{d_n} U_{Ai} N_A(\mathbf{X}), \tag{5}$$

where U_{Ai} is a component of $\mathbf{U} \in \mathbb{R}^{d_n \times 3}$ that denotes the coefficient at the Ath node in the ith direction ($i = 1, 2, 3$) on a mesh of d_n nodes, and $N_A(\mathbf{X})$ is a component of $\mathbf{N}(\mathbf{X})$ that denotes the finite element basis function at the Ath node. Next, we treat the parameterized boundary conditions as the two classic types: Neumann and Dirichlet. For notational simplicity, we define $\mathbf{M} \in \mathbb{R}^{d_m}$ to represent both types.

2.3.1. Parameterization of Dirichlet boundary conditions

Consider the Dirichlet boundary conditions on $(\partial\Omega_0)_u$ are parameterized by $\mathbf{M} \in \mathbb{R}^{d_m}$, using

$$\mathbf{u}(\mathbf{X}) = \mathbf{u}_0(\mathbf{X}, \mathbf{M}) \text{ on } (\partial\Omega_0)_u. \tag{6}$$

Each instance of $\mathbf{M}$ corresponds to a prescribed displacement field $\mathbf{u}_0$ on $(\partial\Omega_0)_u$. At the discrete mesh level, this implies the nodal displacement $\mathbf{U}_A$ (the Ath row of $\mathbf{U}$) at the Ath node with coordinates $\mathbf{X}_A$ on $(\partial\Omega_0)_u$ satisfies

$$\mathbf{U}_A = \mathbf{u}_0(\mathbf{X}_A, \mathbf{M}). \tag{7}$$

2.3.2. Parameterization of Neumann boundary conditions

Next, we consider Neumann boundary conditions on $(\partial\Omega_0)_t$ are parameterized by $\mathbf{M} \in \mathbb{R}^{d_m}$, using

$$\mathbf{P}^{\mathrm{T}}(\mathbf{X})\mathbf{n}_0(\mathbf{X}) = \mathbf{p}(\mathbf{X}, \mathbf{M}) \text{ on } (\partial\Omega_0)_t, \tag{8}$$

where $\mathbf{P}$ is the first Piola–Kirchhoff stress tensor, $\mathbf{n}_0$ is the surface normal direction, and $\mathbf{p}$ is the traction vector field defined on $(\partial\Omega_0)_t$.

2.4. Neural network surrogate for the parametric PDE

We start the formulation of the NNFE model using

$$\mathbf{U} = f_{\mathrm{NN}}(\mathbf{M}) = f_{\mathrm{FCN}}(\mathbf{M}) + \mathbf{W}_{\mathrm{s}}\mathbf{M}, \tag{9}$$

where f_{FCN} is a fully connected neural network and $\mathbf{W}_s \in \mathbb{R}^{d_n \times 3 \times d_m}$ is the weight matrix for the skip connection [56]. Skip connections are extra connections between nodes in different layers of a neural network that skip one or more layers. The introduction of skip (or residual) connections is used to directly back-propagate the gradients linearly from the output to the inputs, which improves the convergence of the training. The fully connected neural network

with L hidden layers is thus a sequence of composite functions of element-wise nonlinear functions ϕ and affine transformations $\mathbf{A}_l$ $(l = 0, \ldots, L)$, given by

$$f_{\text{FCN}} = \mathbf{A}_L \circ \phi \circ \mathbf{A}_{L-1} \cdots \circ \phi \circ \mathbf{A}_0, \tag{10}$$

where $\circ$ represent function composition, e.g., $f \circ g(x) = f(g(x))$ for functions f and g. The affine transformation $\mathbf{A}_l$ $(l = 0, \ldots, L)$ is defined as

$$\mathbf{A}_l(\mathbf{h}) = \mathbf{W}_l \mathbf{h} + \mathbf{b}_l, \tag{11}$$

where $\mathbf{h}$ are the inputs of the neural network for $l = 0$ or the hidden state for lth layer. Specifically, $\mathbf{A}_l : \mathbb{R}^{d_{l-1}} \mapsto \mathbb{R}^{d_l}$ $(l = 1, \ldots, L)$ is determined by its weight $\mathbf{W}_l \in \mathbb{R}^{d_l \times d_{l-1}}$ and bias $b_l \in \mathbb{R}^{d_l}$, $\mathbf{A}_L : \mathbb{R}^{d_L} \mapsto \mathbb{R}^{3 \times d_n}$ is determined by its weight $\mathbf{W}_L \in \mathbb{R}^{3 \times d_n \times d_L}$ and bias $b_L \in \mathbb{R}^{3 \times d_n}$, d_l $(l = 0, \ldots, L)$ is the number of neurons at each layer. To simplify notation, the collection of all parameters of the NNFE model is denoted as θ. We also note that for notational simplicity we denote the function Ψ through its dependence on $\mathbf{u}$ as (Eq. (4)) as $\Psi(\mathbf{u})$.

In order to train the resulting f_{NN}, we generate N_{train} samples of $\mathbf{M}$ using a Quasi-Monte Carlo method, low-discrepancy Halton sequence [57,58] to cover the training space more evenly. Given each sample $\mathbf{M}^{(i)}$ $(i = 1, \ldots, N_{\text{train}})$, the neural network will produce a corresponding $\mathbf{U}^{(i)}$. Next, we assign the Dirichlet boundary condition to $\mathbf{U}^{(i)}$ and construct the FE function $\hat{\mathbf{u}}^{(i)}$. We then minimize the average potential energy $\mathcal{H}$ using

$$\min_{\theta} \qquad \mathcal{H} \overset{\text{def}}{=} \frac{1}{N_{\text{train}}} \sum_{i=1}^{N_{\text{train}}} \Pi(\hat{\mathbf{u}}^{(i)}; \mathbf{M}^{(i)}), \tag{12}$$

where we evaluate the potential energy via

$$\Pi(\hat{\mathbf{u}}^{(i)}; \mathbf{M}^{(i)}) = \int_{\Omega_0} \Psi(\hat{\mathbf{u}}^{(i)}) \mathrm{d}\Omega - \int_{\partial(\Omega_0)_t} \mathbf{p}(\mathbf{M}^{(i)}) \cdot \hat{\mathbf{u}}^{(i)} \mathrm{d}(\partial\Omega) \tag{13}$$

$$\hat{\mathbf{u}}^{(i)} = \mathbf{N}\mathbf{U}^{(i)}, \tag{14}$$

$$\mathbf{U}^{(i)} = f_{\text{NN}}(\mathbf{M}^{(i)}; \theta), \tag{15}$$

$$\mathbf{U}_A^{(i)} = \mathbf{u}_0(\mathbf{X}_A; \mathbf{M}). \tag{16}$$

We note that our approach is fundamentally different from the Deep Ritz approach [40]. In particular, our approach seamlessly bridges the neural network surrogate model with a model of the domain defined by finite elements. This allows the Dirichlet boundary to be discretized with a conforming mesh and interpolation basis functions so that it can be imposed strongly by constraint $\mathbf{U}^{(i)}$ to satisfy $\mathbf{U}_A^{(i)} = \mathbf{u}_0(\mathbf{X}_A; \mathbf{M}^{(i)})$. In turn, the Neumann boundary conditions can be naturally satisfied when the potential energy is minimized. The resulting computational algorithm for evaluation of the potential energy is illustrated in Fig. 1. In the computational algorithm, the nodes represent operators and the edges represent data or variables. Each operator is applied on the variables that flow into the node to generate outputs. The implementation consists of two parts, the FE part and the NN part. An example for the correspondence between the NN outputs and the FE nodal values is shown in Fig. 2. These two parts are implemented in a single machine learning pipeline to have a seamless integration, as discussed in the next section.

2.5. Differentiable implementation of the NNFE model

In light of the ability to utilize automatic differentiation we took a functional approach to implement differentiable finite elements that can be seamlessly integrated with the NNFE in a machine learning framework. The automatic differentiation library we used in this work is `Jax` [59]. In the context of differentiable implementation, first-order or high-order derivatives can be obtained by applying differential operators to the associated functions. Automating the construction of the computational graph for evaluating the derivatives can be achieved by applying automatic differentiation. It is important to properly construct the computational graph for the function that needs to be differentiated. To this end, this section focuses on the technical details of the differentiable implementation of finite elements.

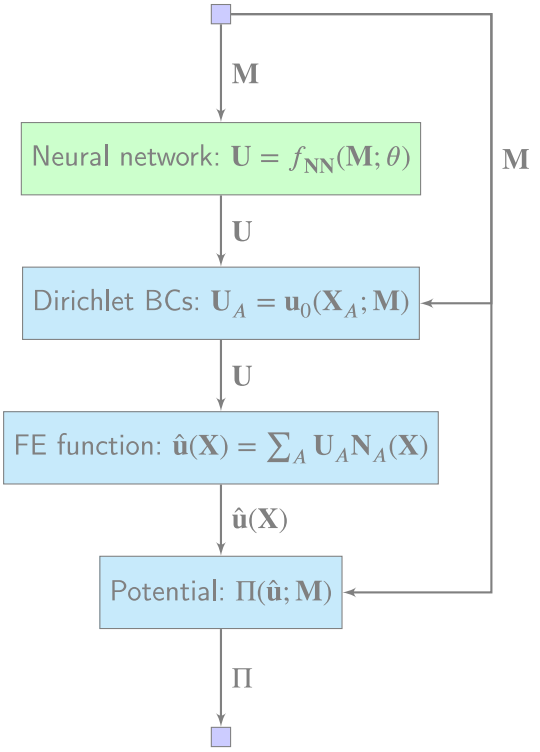
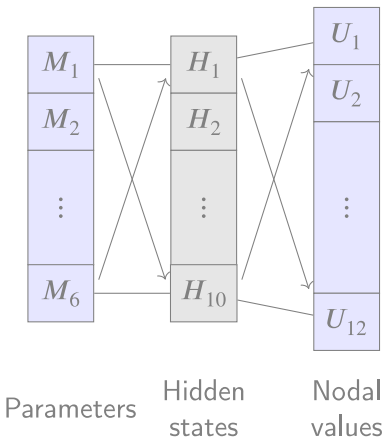
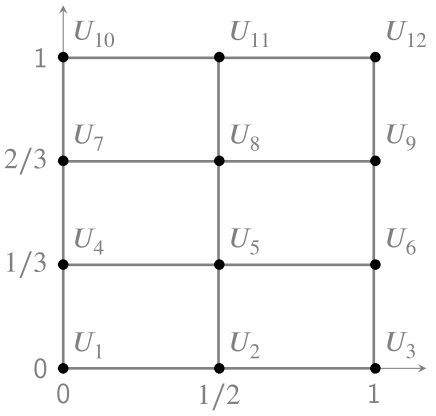


Fig. 1. The computational algorithm for evaluation of the energy function using neural network surrogate model (green) and finite element discretization (blue) for hyperelastic problem with varying boundary conditions. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) A network surrogate model with one layer and ten neurons that maps **M** to **U**.



(b) A FE function with nodal values **U** mapped from the neural network surrogate model and Q_1 Lagrange elements.

Fig. 2. Schematic illustration of a neural network surrogate model and its corresponding FE discretization.

2.5.1. Differential operators

Without loss of generality, we consider a scalar FE-represented function $v(\mathbf{X})$, which can also be similarly applied for a vector FE function, such as the displacement field $\mathbf{u}(\mathbf{X})$. We denote Ω_e as the reference element, and $\boldsymbol{\xi} \in \Omega_e$ as a point in the reference element. With a given coordinate map $\mathbf{X}(\boldsymbol{\xi})$, the evaluation of v at a certain Quadrature point $\boldsymbol{\xi}$ can be performed by

$$v(\boldsymbol{\xi}) = \sum_{\alpha} V_\alpha N_\alpha(\boldsymbol{\xi}), \tag{17}$$

where α is the local degree-of-freedom, N_α are the basis functions of which the support includes $\boldsymbol{\xi}$. From the functional perspective, the FE function $v(\boldsymbol{\xi})$ is not aware of the physical point $\mathbf{X}$. The differential operators that are commonly used in the PDEs, such as $\partial/\partial\mathbf{X}$, are the differentiation with respect to the physical coordinates. Since the inverse of the coordinate map $\boldsymbol{\xi}(\mathbf{X})$ is undetermined in the context of finite elements, we cannot construct a computational graph corresponding to $v(\boldsymbol{\xi}(\mathbf{X}))$ so that v is aware of $\mathbf{X}$.

To address this issue, we use the inverse of the Jacobian of the coordinate map

$$\mathbf{J}^{-1} = \left(\frac{\partial \mathbf{X}}{\partial \boldsymbol{\xi}}\right)^{-1} \tag{18}$$

as in the classical finite element methods. Then we can apply the linear transformation determined by $\mathbf{J}^{-T}$ locally using

$$\frac{\partial}{\partial \mathbf{X}} = \mathbf{J}^{-\mathrm{T}} \frac{\partial}{\partial \boldsymbol{\xi}} \tag{19}$$

to evaluate $\partial/\partial\mathbf{X}$.

2.5.2. Application of the boundary conditions

Dirichlet boundary conditions. There are multiple approaches to apply the Dirichlet boundary conditions in the traditional finite element method (for example, modifying the stiffness matrix and the force vector). In the context of differentiable implementation, the Dirichlet boundary conditions can be imposed explicitly by fixing the value of the corresponding components associated with the Dirichlet boundary conditions. By fixing the values, the partial derivatives of the objective function with respect to the corresponding components will be zero. Thus, the back-propagation algorithm will propagate zero gradient through the fixed components. The efficacy of this simple scheme will be demonstrated in the numerical examples with Dirichlet boundary conditions.

Neumann boundary conditions. The Neumann boundary conditions were imposed weakly as an external potential in the objective function of the training algorithm. In classical finite element theory [60] the stationary condition of principle of potential energy leads to the weak form of the governing equations or the variational equations. The Neumann boundary conditions are naturally imposed when the potential energy is stationary. In the training algorithm, we minimize the potential energy on the sampled input parameters to weakly impose the Neumann boundary conditions in an average sense.

2.5.3. Software implementation

`Jax` [59] is the automatic differentiation backend in the present study. It includes composable function transformations, such as just-in-time compilation, automatic differentiation, automatic vectorization, parallel programming of multiple accelerators. Its capability to interleave these transformations enabled the implementation of the differentiable finite elements intuitively. The just-in-time compiler provides efficiency optimization for the resulting computational graph. We also note that the use of automatic differentiation eliminated the need for tedious and error prone manual derivation of the mathematical expressions of derivatives. The automatic vectorization also allowed us focus on the implementation at each quadrature point. The communication primitives provided a convenient way for distributed learning using the single-program multiple-data (SPMD) paradigm. One drawback related to back-propagation is that it is essentially the adjoint method, so that the intermediate results of the computational graph need to be cached in order to propagate back the gradient information using the Jacobian transpose. This leads to an increased memory requirement for complex problems with a deep computational graph. In this study we stored all intermediate results for back-propagation. In the future work we plan to use a divide-and-conquer approach to reduce the complexity of the memory requirement [61]. The resulting computational pipeline was implemented to run on multiple GPUs leveraging the computational resources of Frontera at the Texas Advanced Computing Center.

2.6. Training scheme

The averaged energy potential for a number of realizations of the parameters $\mathcal{H}$ is the objective function for training the neural network representation of the parameter-to-state map. In the discretized setting, we aggregate the potentials for a set of realizations of $\mathbf{M}^{(i)}$ $(i = 1, \ldots, N_{\text{tr}})$ within a prescribed range or box $[\mathbf{M}^{\text{LB}}, \mathbf{M}^{\text{UB}}]$ and find the optimal parameters for the neural network that minimizes the aggregated potential. Sampling $\mathbf{M}^{(i)}$ can be done by stochastic methods such as Monte Carlo methods, or pseudo-stochastic methods such as quasi-Monte Carlo methods, or deterministic methods such as quadrature schemes using weighted sum. The quasi-Monte Carlo method has a faster rate of convergence in comparison with Monte Carlo methods when the variation of the functions is limited and low-discrepancy sequence is used. Moreover, quadrature schemes are intractable for high-dimensional problems. In this work, we thus used a Halton sequence to generate samples for $\mathbf{M}$. To facilitate the training process, we gradually expanded the training range $[\mathbf{M}^{\text{lb}}, \mathbf{M}^{\text{ub}}]$ to avoid mesh distortions until $\mathbf{M}^{\text{ub}} = \mathbf{M}^{\text{LB}}$ and $\mathbf{M}^{\text{ub}} = \mathbf{M}^{\text{UB}}$. For a given training range $[\mathbf{M}^{\text{lb}}, \mathbf{M}^{\text{ub}}]$, we sample the training dataset $\mathcal{D}_{\text{tr}} = \{\mathbf{M}^{(i)}\}_{i=1}^{N_{\text{tr}}}$ of realizations of $\mathbf{M}$ and adjust the parameters of the neural network θ by using inexact Newton-Conjugate Gradient (Newton-CG) method with Armijo line search [62]. This is a matrix-free algorithm in the sense that it only needs the Hessian action without constructing the full Hessian matrix.

Initialize a random initial guess θ_0;
for $k = 1$ **to** K **do**
  $\mathbf{M}^{\text{lb}} \leftarrow k\mathbf{M}^{\text{LB}}/K$, $\mathbf{M}^{\text{ub}} \leftarrow k\mathbf{M}^{\text{UB}}/K$, $\theta_k \leftarrow \theta_{k-1}$;
  sampling $\mathcal{D}_{\text{tr}} = \{\mathbf{M}^{(i)}\}_{i=1}^{N_{\text{tr}}}$ within $[\mathbf{M}^{\text{lb}}, \mathbf{M}^{\text{ub}}]$;
  for $j = 1$ **to** J **do**
    $p_k = \text{newton\_cg}(\theta_k)$;
    $\alpha_k = \text{linear\_search}(\mathcal{H}(\theta_k + \alpha p_k))$;
    $\theta_k = \theta_k + \alpha_k p_k$;
    if $||\nabla_\theta \mathcal{H}(\theta_k)||/||\nabla_\theta \mathcal{H}(\theta_0)|| < \epsilon_g$ **then**
      return ;
    end
  end
end

Algorithm 1: Stepping scheme for training the neural network surrogate model. In the k-th iteration ($k = 1, \ldots, K$), scale the training upper bound training range $[\mathbf{M}^{\text{lb}}, \mathbf{M}^{\text{ub}}]$ until it reaches the limit $[\mathbf{M}^{\text{LB}}, \mathbf{M}^{\text{UB}}]$. Sample the training dataset $\mathcal{D}_{\text{tr}}$ with N_{tr} samples. Construct the search direction p_k using Newton-CG method, using Armijo line search to find the step size α_k for the objective function $\mathcal{H}$. Finally, update the neural network parameters θ_k until the J-th iteration or the relative gradient is below a prescribed threshold ϵ_g.

The initialization of the parameters θ has a significant impact to the convergence of the training. To avoid ill-conditioning, the weight matrices $\mathbf{W}_i$ $(l = 1, \ldots, L-1)$ are initialized as orthonormal. To avoid mesh distortions at the initial step, we restrict the neural network functions to satisfy that $f_{NN} \to 0$ when $\mathbf{U} \to \mathbf{0}$. This asymptotic condition can be achieved by initializing biases $\mathbf{b}_i$ $(l = 1, \ldots, L)$ to be zeros, and the weight matrix $\mathbf{W}_L$ in the last layer to be zeros and using continuous activation function ϕ that satisfies condition $\phi(x = 0) = 0$. We choose the tangent hyperbolic function due to its boundedness and $\tanh(x = 0) = 0$.

2.7. Transfer learning

A final trained NNFE for a specific specimen (or more generally an individual) is typically only predictable for that case only. However, although the responses vary with the corresponding individual data, they tend to share similar behaviors. In the present application, the mechanical behavior of passive myocardium will vary but within a definable range. A natural extension of the present methods is to fine-tune the weights and biases with a different set of data, a form a transfer learning, which as the potential to significantly reduce training time. A transfer learning strategy is presented to leverage a trained neural network to reduce the net training time of subsequently trained specimens or individuals.

Without loss of generality, the varying individual data considered is the set of material parameters and the fiber structures. We denote such data as $\mathbf{D}$ for notation simplification, the αth realization of $\mathbf{D}$ as $\mathbf{D}_\alpha$. We assume a pre-trained NN $f_{\text{NN}}(\mathbf{M};\theta_\alpha)$ has been optimized. Its parameters θ_α are indexed by α as well to highlight its relationship with $\mathbf{D}_\alpha$, and θ_α is the optimizer in the sense

$$\theta_\alpha = \arg\min_\theta \frac{1}{N_{\text{tr}}} \sum_{i=1}^{N_{\text{tr}}} \Pi(\theta; \mathbf{M}^{(i)}, \mathbf{D}_\alpha), \tag{20}$$

where we unfold the dependence of Π on θ through $\hat{\mathbf{u}}$ to highlight the optimization variable, i.e. the neural network parameters θ. For another individual, we denote the corresponding realization as $\mathbf{D}_\beta$. To avoid finding the optimizer θ_β from a random initial guess, we can transfer our knowledge of the learned θ_α to θ_β that dictates similar responses of f_{NN}. The transfer learning in this context is thus to fine-tune θ_α to adapt to $\mathbf{D}_\beta$ by Algorithm 2. We keep all the hyperparameters, the neural network architecture and the sampled training dataset the same as the pre-trained model while only the neural network parameters are fine-tuned with a different combination of fiber architecture and material parameters.

Initialize $\theta_0 \leftarrow \theta_\alpha$;
for $k = 1$ **to** K **do**
  $\mathbf{M}^{\text{lb}} \leftarrow k\mathbf{M}^{\text{LB}}/K$, $\mathbf{M}^{\text{ub}} \leftarrow k\mathbf{M}^{\text{UB}}/K$, $\theta_k \leftarrow \theta_{k-1}$;
  sampling $\mathcal{D}_{\text{tr}} = \{\mathbf{M}^{(i)}\}_{i=1}^{N_{\text{tr}}}$ within $[\mathbf{M}^{\text{lb}}, \mathbf{M}^{\text{ub}}]$;
  $\theta_k = \arg\min_\theta \frac{1}{N} \sum_{i=1}^{N} \Pi(\theta; \mathbf{M}^{(i)}, \mathbf{D}_\beta)$
end
$\theta_\beta \leftarrow \theta_K$

Algorithm 2: Transfer learning scheme for finding the optimizer θ_β on $\mathbf{D}_\beta$ using θ_α that is trained on $\mathbf{D}_\alpha$ as an initial guess. We fold the details of the optimization algorithm to highlight the important aspect of transfer learning.

2.8. Numerical simulations

In summary, we considered a cuboidal tissue specimen of dimensions 1 cm × 1 cm × 1 cm, emulating the triaxial experiments. The domain is discretized by tri-linear Lagrange elements. The element-wise second order Gaussian quadrature is used for numerical integration of the element potential energy, and the first order Gaussian quadrature for volumetric contribution of the strain density energy function. To demonstrate the performance of the neural networks, we examined an error metric on the validation dataset $\mathcal{D}_{\text{val}} = \{\mathbf{M}^{(i)}\}_{i=1}^{N_{\text{val}}}$. The concrete examples of $\mathcal{D}_{\text{val}}$ are presented in the rest of this section for Neumann and Dirichlet boundary conditions. $\mathcal{D}_{\text{val}}$ is a set of different boundary conditions for which we will examine the discrepancy between the neural network predictions and the FE solutions. With a given realization of $\mathbf{M}$, the neural network prediction of the displacement nodal values is denoted as $\mathbf{U}_{\text{NN}}^{(i)}$ and the FE solution $\mathbf{U}_{\text{FE}}^{(i)}$. The L_∞ norm of the nodal errors $\mathbf{U}_{\text{NN}}^{(i)} - \mathbf{U}_{\text{FE}}^{(i)}$ is the maximum value of $|\mathbf{U}_{\text{NN}}^{(i)} - \mathbf{U}_{\text{FE}}^{(i)}|$. The average L_∞ error e_{L_∞} on $\mathcal{D}_{\text{val}}$ is defined as

$$e_{L_\infty} = \frac{1}{N_{\text{val}}} \sum_{\mathbf{M}^{(i)} \in \mathcal{D}_{\text{val}}} \|\mathbf{U}_{\text{NN}}^{(i)} - \mathbf{U}_{\text{FE}}^{(i)}\|_\infty, \tag{21}$$

which is the average of the maximum displacement errors for all boundary conditions in $\mathcal{D}_{\text{val}}$.

We note that our "training" data is simply the prescribed displacements (in units of cm) or tractions (in units of Pa). Normalization of training data is important in applications wherein actual data is used. In contrast, in our approach we did not need to consider input parameters with different physical units or wide range of values in the same surrogate modeling problem. The two separate problems we considered are parameterized Dirichlet boundary conditions and parameterized Neumann conditions. Thus, we did not need to explicitly normalize the inputs and outputs of the neural network.

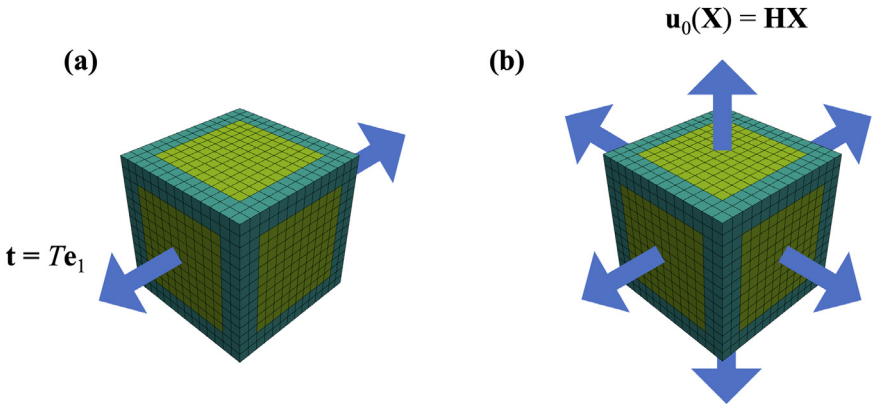


Fig. 3. Parameterization of boundary conditions applied to the yellow areas for (a) single loading path using $\mathbf{t} = T\mathbf{e}_1$, $T \in \mathbb{R}$, and (b) full physiological range using $\mathbf{u}_0(\mathbf{X}) = \mathbf{H}\mathbf{X}$, $\mathbf{H} \in \mathbb{R}^{3\times3}$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

2.8.1. Single loading path parameterized by a Neumann boundary condition

The traction field applied to the facets that are normal to one of the coordinate basis $\mathbf{e}_1$ is parameterized by a scalar $T \in \mathbb{R}$ using $\mathbf{T} = T\mathbf{e}_1$, $T \in [0, 1500]$ Pa (Fig. 3-a). In this concrete example, T serves as the input parameter M for the surrogate model. The training dataset includes 12 uniformly spaced points in $[0, 1500]$. The fiber structure is homogeneous and the material axes are aligned with the Cartesian basis as $\mathbf{f}_0 = \mathbf{e}_1$, $\mathbf{s}_0 = \mathbf{e}_2$ and $\mathbf{n}_0 = \mathbf{e}_3$. Since this becomes a pure Neumann problem, we need to restrict the rigid body motion. Due to the symmetry of the problem, we can apply symmetric boundary conditions to three mid-planes that are across the center point and normal to Cartesian bases $\mathbf{e}_i$ $(i = 1, 2, 3)$. The symmetric boundary conditions restrict the nodes on these mid-planes to move within the planes only to eliminate the rigid body rotation and translation. For validation purposes, $\mathcal{D}_{\text{val}}$ includes 10 uniformly spaced points within the training range [0, 1500 Pa]. We start with a mesh of $14 \times 14 \times 14$ elements to demonstrate the expressiveness of the neural network. The fibers are orientated in the $\mathbf{e}_1$ direction uniformly. The NNFEs were trained on a NVIDIA(R) GeForce RTX 2080 Ti GPU. Then, we further demonstrate the speed-up of the neural network forward pass compared with the finite element solutions with two refined meshes with $28 \times 28 \times 28$ elements and $56 \times 56 \times 56$ elements. To compare the run times on the same hardware, we use a serial single-core program on a Intel(R) Core(TM) i9-9920X on a System 76 Thelio Major computer.

2.8.2. Full physiological deformations parameterized by Dirichlet boundary conditions

The Dirichlet boundary conditions are parameterized using affine deformation modes parameterized by $\mathbf{H} \in \mathbb{R}^{3\times3}$ using

$$\mathbf{u}_0 = \mathbf{H} \cdot \mathbf{X}, \tag{22}$$

where $\mathbf{X}$ is the reference configuration. $\mathbf{H}$ is physically the displacement gradient tensor applied as a Dirichlet boundary condition and serves as $\mathbf{M}$ for the surrogate model training (Fig. 3-b). Without loss of generality, we remove the rigid body rotation by restricting $\mathbf{H}$ to be upper diagonal as

$$\mathbf{H} = \begin{bmatrix} H_1 & H_4 & H_5 \\ & H_2 & H_6 \\ & & H_3 \end{bmatrix}. \tag{23}$$

The deformation modes of the corresponding components of $\mathbf{H}$ are illustrated in Fig. 4. The off-diagonal terms represent the magnitudes of shear deformation modes. The diagonal terms represent the magnitudes of the extension deformation modes, and $\mathbf{H}$ needs to respect the incompressibility constraint of the myocardium to avoid unrealistic deformation modes and resulting divergence issues. To generate physically feasible training data, the incompressibility constraint is imposed by restricting $\mathbf{H} + \mathbf{I}$ to be unimodular where $\mathbf{I}$ is the identity matrix. We

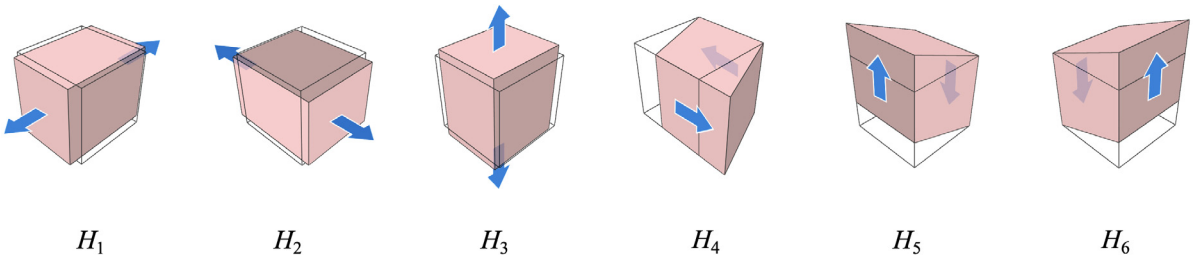


Fig. 4. Component-wise deformation modes of the upper diagonal matrix $\mathbf{H}$. The diagonal components H_i $(i = 1, 2, 3)$ represent the deformation magnitude of the extension modes. The diagonal components H_i $(i = 4, 5, 6)$ represent the deformation magnitude of the simple shear modes.
Source: Adapted from [21].

Table 2
Hyperparameters for training neural network surrogate model and its transfer learning.

Neural network architecture	
Number of neurons	10
Number of hidden layer	1
Activation function	Tangent hyperbolic function
Algorithm 1	
Maximum number of CG iterations	300
Maximum number of Armijo line search iterations	30
Initial step size for Armijo line search	1.0
Number of iteration K	1–12

parameterize $\mathbf{H}$ using

$$\mathbf{H} = \begin{bmatrix} 1/t_1 - 1 & t_3 & t_4 \\ & 1/t_2 - 1 & t_5 \\ & & t_1 t_2 - 1 \end{bmatrix}, \tag{24}$$

where $t_1, \ldots, t_5 \in \mathbb{R}$. Thus, $\mathbf{H}$ automatically satisfies the incompressibility constraint $\det(\mathbf{H} + \mathbf{I}) = 1$. We restrict the diagonal components of $\mathbf{H}$ within $[-0.2, 0.2]$ and the off-diagonal components within $[-0.3, 0.3]$ to cover the physiological deformation range. To demonstrate the learnability of the NNSM, we considered four combinations of fiber architectures and sets of material parameters obtained from a numerical experimental approach [21] (Fig. 5). The validation is performed using a optimal set of six loading protocols including pure shear deformations and simple shear deformations: (1) $H_1 \in [0, 0.2]$ cm, $H_2 \in [-0.167, 0]$ cm; (2) $H_1 \in [0, 0.2]$ cm, $H_3 \in [-0.167, 0]$ cm; (3) $H_2 \in [0, 0.2]$ cm, $H_3 \in [-0.167, 0]$ cm; (4) $H_4 \in [0, 0.3]$ cm; (5) $H_5 \in [0, 0.3]$ cm; (6) $H_6 \in [0, 0.3]$ cm (if not mentioned the other components of $\mathbf{H}$ are zeros). Each protocol has 10 uniformly spaced points within the training range.

2.8.3. Transfer learning for different combination of material parameters and fiber architectures

We leveraged the transfer learning techniques to speed up a consequence training process by initialization using a trained neural network. We start with a trained neural network for specimen 1 (S1). The trained neural network then served as a initial guess for learning the NNSMs for the other three different combinations (S2–S4) of material parameters and fiber architectures in Fig. 5. To demonstrate the usage of the transfer learning, we compared the accuracy and the training run time for random initialization and initialization using transferred parameters.

3. Results

The hyperparameters used are summarized in Table 2. As described in Section 2.6, the shallow neural network is activated by tangent hyperbolic function which satisfies the condition $\tanh(x = 0) = 0$ and the weights and biases in the last layer are zeroed to avoid mesh distortion at the beginning of training. The training using the inexact

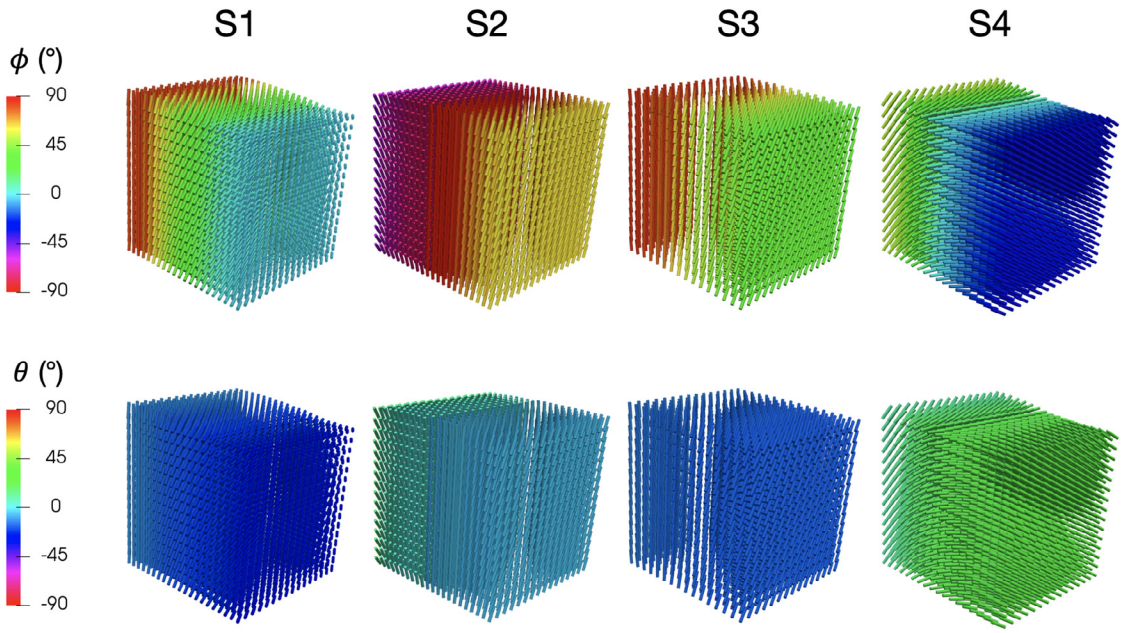


Fig. 5. The fiber directions are represented with respect to the circumferential–longitudinal planes of the left ventricle. The in plane angles ϕ and the out of plane angles θ for four myocardium specimens computed from the diffusion tensor imaging are represented as colored arrows. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)
Source: Adapted from [21].

Newton-CG optimizer with line search is relatively robust with the hyperparameters listed in Table 2. To avoid pathological behavior at the beginning of training and train the model gradually, K in Algorithm 1 varies from 1 to 12 to refine the stepping scheme and make the training stable, without the knowledge of the validation error.

We first examined the proposed solution for parameterized Neumann boundary conditions. The trained neural network surrogate model with only 10 hidden neurons associated with a mesh of $14 \times 14 \times 14$ elements demonstrated well alignment with the corresponding finite element solution (Fig. 6). The run time for a single prediction of the neural network surrogate model increases minimally with increasing number of elements, while the finite element assembly and solution for a single step increases super-linearly with the number of elements (Fig. 7). The neural network surrogate model can give predictions in parallel while the finite element solver needs a stepping scheme to incrementally obtain the solutions for the fully loaded state which would multiply the cost with the number of steps.

Next, we examined the proposed approach for parameterized Dirichlet boundary conditions. With 10 hidden neurons and $14 \times 14 \times 14$ elements and 140 training data points in the full physiological range, the trained neural network surrogate model has an average L_∞ error of 0.1201 mm on the validation dataset (Fig. 8). We then varied the number of samples for $\mathbf{M}$ or $\mathbf{H}$ in this concrete example generated using Halton sequence to study its effects to the accuracy of the neural network. The average L_∞ validation error quickly reduces to 0.1682 mm for 20 $\mathbf{H}$ samples and then gradually reduces to 0.1201 mm for 140 $\mathbf{H}$ samples (Fig. 9). The average L_∞ validation errors for each specimen, each one with its unique set of material parameters and fiber architectures, are listed in Table 3. Note that all specimens demonstrated similar errors. Using the same neural network architecture and training algorithm, the neural network surrogate model is expressive for the responses of all these four specimens, which demonstrates the efficacy of the proposed approach for the complex mechanical behavior of myocardium.

Lastly, we applied the transfer learning techniques to reuse a trained neural network surrogate model for a different combination of material parameters and fiber architectures for S2–S4. We examined two metrics: the average L_∞ error e_{L_∞} to quantify the accuracy on the validation dataset and the training time T_{tr} to quantify the computational expense for training. The ones using transfer learning are denoted by a superscript $(\cdot)^{\text{transfer}}$ to differentiate from the ones using random initialization. Due to the similarity between the responses of S1 and those of S2–S4, the training time using transfer learning was significantly reduced while the accuracy retained (Table 4).

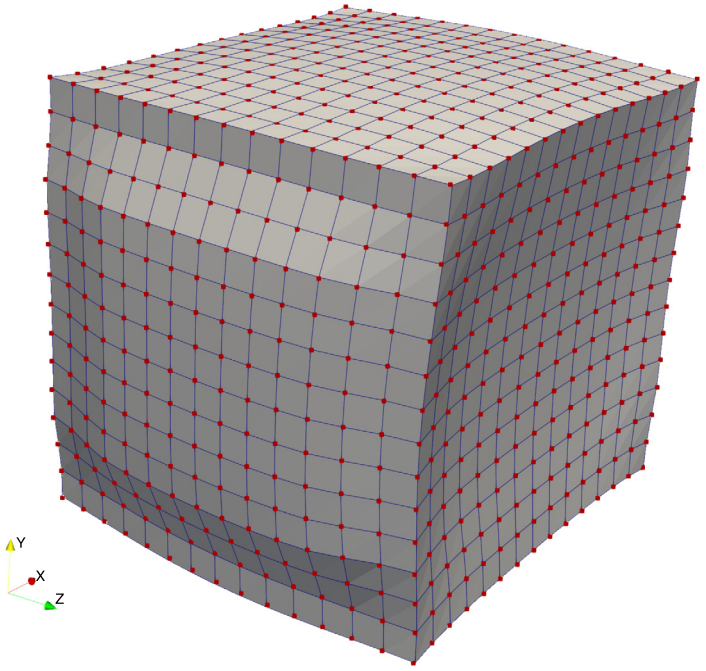


Fig. 6. Validation of the NN prediction (blue wire frames) with the FE solution (red points) for parameterized Neumann boundary conditions at 1500 Pa. The average L_∞ error is 0.04501 mm for the cube with 1 cm on each side.

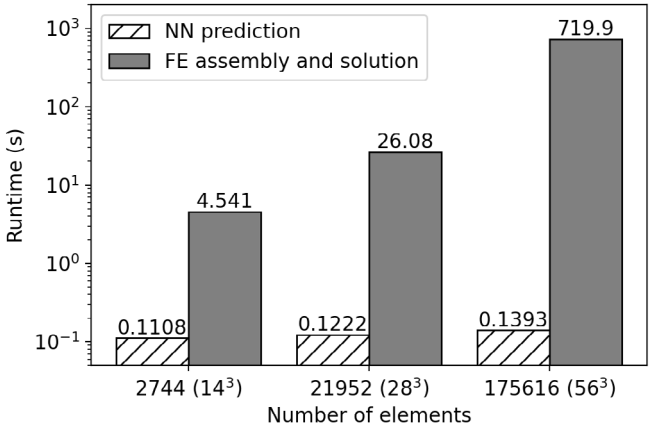


Fig. 7. Observed simulations times of the NNSM predictions compared with the equivalent FE solutions for three levels of mesh size. The NNSM provided a significantly decreased computational time in all cases, with up to 5168 fold decrease with the finest mesh. Note too that as the mesh sized increased from 2744 to 175 615 elements, the NNSM computational time only increased 1.257 times from 0.1108 s to 0.1393 s while the FE increased 158.5 times from 4.541 s to 719.9 s.

Table 3
The average L_∞ error of neural network predictions using the corresponding FE solutions as ground truth on the validation datasets.

	S1	S2	S3	S4
Average L_∞ error	0.1201 mm	0.1171 mm	0.1639 mm	0.2012 mm

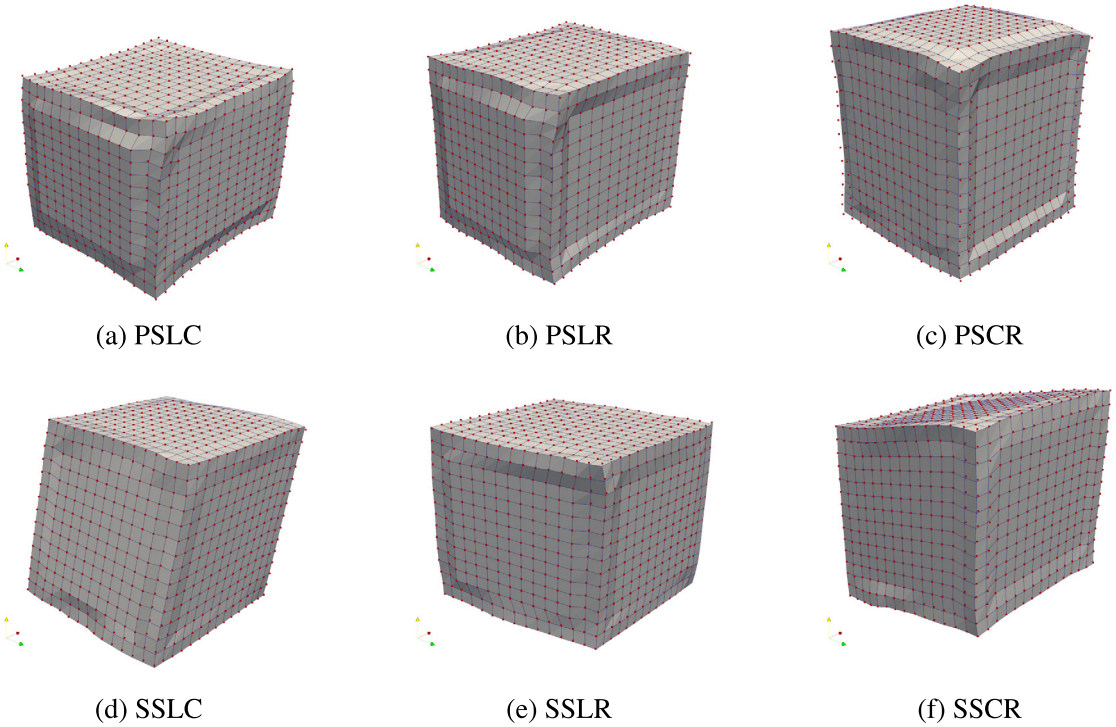


Fig. 8. Comparison between FE solutions (red points) and NN predictions (blue wire frames). The average L_∞ error is 0.1201 mm. The plotted six loading protocols are same as the original model in [21] in terms of the deformation. Abbreviations: pure shear (PS), simple shear (SS), longitudinal (L), circumferential (C), radial (R).

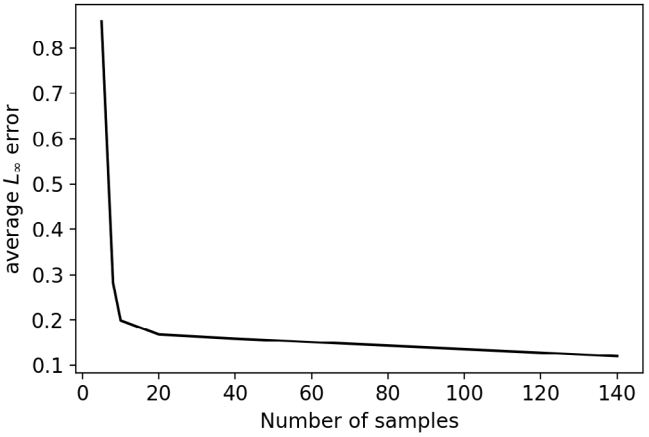


Fig. 9. The number of samples for the input parameter **H** that parameterized the Dirichlet boundary conditions vs. the average L_∞ validation error of the displacement nodal values.

4. Discussion

4.1. Approach and major findings

In this study we developed a neural network finite element (NNFE) model approach to represent the fully three dimensional behavior of hyperelastic behavior of passive ventricular myocardium. Our method utilized a finite element discretization of the domain to compute the local potential energy for a given displacement field,

Table 4
The validation errors e_∞ and the training run time T_{tr} for transfer learning using the trained NN parameters from S1 compared with the ones with random initialization. The ones using transfer learning is indicated by a superscript $(\cdot)^{\text{transfer}}$.

	S2	S3	S4
$e_\infty^{\text{transfer}}$	0.1005 mm	0.1639 mm	0.1819 mm
e_∞	0.1171 mm	0.1639 mm	0.2012 mm
$T_{\text{tr}}^{\text{transfer}}$	6 min	19 min	37 min
T_{tr}	35 min	58 min	93 min

with the solution—the FE nodal vector—represented by a neural network. To exploit automatic differentiation the gradient information was back-propagated from the potential energy to the neural network, with the potential energy computed using the numerical integration for each finite element. Note that as in the classical finite element theory, the spatial gradients are evaluated by chain rule on the reference element. Dirichlet boundary conditions were directly imposed by fixing the corresponding components of the finite element vector. Inspired by the underlying physical principle, the optimization problem for training NNFE (Eqs. (12)–(16)) is based on the minimum potential energy principle for hyperelasticity. The objective function is the average potential energy on the sampled boundary conditions in the training dataset. For given boundary conditions, the NNSM predicts a trial solution and the corresponding potential energy is evaluated by the FE part. These techniques enabled us to seamlessly use gradient-based optimization algorithms to find the optimal neural network parameters that minimize the potential energy for a range of input parameters. An advanced optimization method with inexact Newton-CG and globalization with Amijo line search was developed and implemented for our NNFE framework. In our numerical tests we utilized actual myocardial cuboidal specimen geometry, boundary conditions, fiber structure, and an orthotropic constitutive model to carefully evaluate this method for cardiac tissue simulations as a directed application. This was a much more rigorous test compared to other studies, which tended to focus on isotropic (e.g., neo Hookean) material models with homogeneous internal structures.

The trained NNFE not only faithfully represented the complete 3D mechanical response very well (Fig. 8), but also exhibited extremely fast run times, with up to 5168-fold decrease in computation time as compared to a conventional FE solution using the same mesh and material model (Fig. 7). One of the major practical advantages of the present approach is its scalability. In the present study, as the mesh size was increased from 2744 to 175,615 elements (a 64 fold increase) the NNFE computational time only increased from 110.8 ms to 139.3 ms. This is in contrast with the conventional FE approach which increased from 4.541 to 719.9 s (Fig. 7). This result suggests that with the use of a NNFE the *run times* can be significantly reduced compared with the traditional large-deformation based finite element solution methods. This is especially important for many-query problems, where hundreds to thousands of simulations may be undertaken. The trade-off is to train the NNFE offline within a range of anticipated physiological responses. However, training time would only have to be performed once before any number of application uses. Moreover, since the NNFE is an analytical function its computational performance will be amplified when the corresponding problem becomes more complex and more computationally expensive. We need to emphasize that our goal in the present study was not to compare a single deformation path simulation between the FE and NNFE methods, including all training and pre-processing steps. Rather, our goal is that, once trained over a range of all anticipated deformation states, the NNFE method can reproduce any deformation state within the trained range accurately and quickly. Thus, our focus is on the end point applications of the heart, wherein a fully trained NNFE model is exclusively used (with no further training required). The transfer learning methods further allow us to reduce training time in multi-specimen scenarios.

4.2. Use of transfer learning to decrease training time

Transfer learning is a technique for transferring knowledge to the target domain from a different source domain(s). Transfer learning has been applied to text/image classification, medical imaging, etc. [63]. In the present application, we applied transfer learning to utilize pre-trained NNFE from one test specimen as an initial guess for a different specimen, which has a different combination of material parameters and fiber structures. The training time can be

significantly reduced by using transfer learning techniques when the original responses and the targeting responses are similar. The transfer learning approach fine-tuned the predictions of the NNFE. In the context of cardiac modeling, the material parameters and the fiber structures cannot be incorporated into the input parameters of the NNFE due to the insufficient training data.

4.3. Broader motivations for improved cardiac simulations

The ultimate application goal of the present approach is surgical planning and clinical diagnosis in clinically relevant time frames. The nature of these applications is time-sensitive and they usually involve many outcome evaluations (potentially thousands). Traditional finite element methods that have been extensively used are quite well established for cardiac modeling [64,65]. While the cardiac models become more sophisticated to achieve high-fidelity solutions, the computational time associated with the finite element simulations becomes prohibitive for such time-sensitive clinical applications. To tackle the challenges for efficient online predictions, we developed a NNFE approach. Surrogate modeling in general can shift the computational cost from solving FE equations to training surrogate models. By front-loading the computational cost, significant performance gains can be achieved for online predictions using surrogate models.

4.4. Additional considerations and limitations

In this work, we considered the finite element solutions as the ground truth for accuracy evaluations. Using the same mesh and finite element function space, the neural network predictions in general reproduced the corresponding finite element solutions very closely. It should be stated that the NNFE is not a more accurate or robust method in comparison with traditional FE approaches, but rather represents a means to achieve nearly instantaneous solutions once trained. The source of errors for the NNFE includes approximation errors, optimization errors, and sampling errors [46]. With the advent of automatic differentiation techniques, the neural network training can be guided by back-propagation of the gradients of potential energy in the present approach. To this end, the differentiable implementation of finite elements that we have developed is crucial. We further note that traditional surrogate modeling approaches are usually data-driven, which require evaluations of the simulation results in data fitting, adding additional training time. As we train on the PDE itself, this decreases training time.

The current study remains in the realm of elasticity using the potential-based formulation. This can limit its applicability for general problems such as modeling active behavior of ventricular myocardium. The neural network tightly coupled with the finite element discretization since the output of the neural network corresponds to the components of the finite element nodal values. However, the neural network parameters for the last layer can be interpolated if using a different mesh for the same domain. The back-propagation algorithm requires check-pointing intermediate results of the computational graph from the input parameters to the objective function value. This results in the memory requirement increasing with the number of operations in the computational graph. Thus, the complexity of the computation is bounded by the memory resources. This issue can be alleviated by divide-and-conquer strategies [61].

In addition to the above theoretical and numerical considerations, there are a number of practical issues encountered when dealing with biological structures such a myocardium. As noted above, we “train” the NNFE using a known passive myocardial constitutive model [21]. It may seem better to train the NNFE on the experimental data directly. However, these approaches are only possible when large amounts of mechanical responses can be obtained experimentally. In general, practical considerations greatly limit the amount of mechanical response data that can be collected, due to the limited time explanted myocardium can be kept viable. More importantly, we have demonstrated with the appropriate choice of constitutive model, robust parameter estimation can be achieved using a minimal number of selected loading paths that optimize the D-optimality criteria for a generalized 3D deformation state [19]. In addition to basic force–displacement information, constitutive modeling of any biological structures also requires detailed maps of the internal microstructure (e.g., muscle fiber orientation), which is usually only available in a limited manner (e.g., only in the referential configuration). Along these lines, meso-structural constitutive models have been extensively used and have very good predictive capability, and can be developed with minimal experimental data requirements [66,67]. Thus, our approach exploits use of extant robust constitutive models that optimize parameter estimation, allowing focus on optimizing the computational implementation. Moreover, our approach will intrinsically incorporate all necessary convexity and related features, since it is trained on a mathematically correct constitutive formulation.

4.5. Future directions

Physics-guided NNFE, finite elements, and cardiac physiology-guided training regimes are the integrated parts of the present approach. The essential component of cardiac modeling we are addressing is simulation of the hyperelastic behavior of myocardium, so that we can employ a potential formulation for model training. Although we consider a specific type of hyperelastic material model we developed before, it does not restrict the applicability of the present approach for other types of hyperelastic materials. As an example, the present approach can incorporate local fiber structures to inform the anisotropy of any hyperelastic material. While the present approach assumes the underlying physics problem of the PDEs has a potential formulation, it does not restrict its extensibility for other types of PDEs. For example, the L_2 norm of the PDE residuals can be used in the objective function as discussed in [39,45] when there is no potential formulation for the PDEs. As we consider hyperelastic based problems, there are rich contents in the hyperelasticity theory and the optimization theory for further studies. One possible direction is to consider the trainability of an analysis tractable neural network architecture. The convergence analysis for the simplified elastic problem might provide insights on the underlying mechanism or directions to improve the neural network architecture for a specific physics process.

More complex cardiac simulations will need further development. Specifically, the active contraction is an essential mechanism for driving the motion of the heart. The present formulation needs to be extended to incorporate the associated active stress. In the organ level, the patient-specific cardiac models involve much more complex geometry directly imported from medical imaging data. The complex geometry is usually discretized in an unstructured mesh which needs further technical development. The full cardiac models involve multi-physics in nature. The active contraction of myocardium is electrophysiologically-driven. The pressures applied in the chambers of the heart are induced by the fluid dynamics. Ultimately, the coupling of the electromechanical and fluid dynamics constitutes the underlying mechanisms. In addition, incorporating the potential pathophysiological changes of the myocardium in the cardiac models is also important for long-term predictions of myocardium remodeling. Such future directions will likely include the fundamental studies on the NNFEs. In this work, we use a simple neural network architecture heuristically. More rational basis on design of the architecture for specific physical problems is necessary. Since the neural network is tightly coupled with the finite elements, it might be possible to exploit the corresponding mathematical structure to guide the architecture design. As mentioned in Section 4.3, the approximation errors, optimization errors, and sampling errors affect the accuracy of the trained neural network model.

4.6. Conclusion

In this work, we developed a NNFE approach to learn the responses of myocardium within a range of physiologically guided boundary conditions. The NNFE is integrated with the finite element discretization to streamline the existing finite element models. The use of automatic differentiation enabled the back-propagation of the gradient information which facilitates the implementation. This enables us to develop an advanced second order inexact Newton-CG optimization method instead of using the standard at most first order stochastic gradient descent approaches. The potential-based formulation is based on the physical principle of the elasticity problem without relying on the finite element solutions to train the NNFE. Both Neumann and Dirichlet boundary conditions were considered to demonstrate the efficacy of the present approach. The efficiency and accuracy of online predictions of the surrogate model were examined using the corresponding finite element solutions as reference. By training the neural network offline, significant speed up gain can be achieved for online predictions, which is further amplified for a finer mesh. We also demonstrated that usage of transfer learning techniques can further reduce the training time. We believe this work is a first step towards our ultimate application goal of surgical planning and clinical diagnosis in clinically relevant time frames.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Michael Sacks reports financial support was provided by National Institutes of Health.

Acknowledgments

This work was made possible by the National Institutes of Health, USA Grants R01-HL073021 and R01-HL142504.

References

[1] J. Corral-Acero, F. Margara, M. Marciniak, C. Rodero, F. Loncaric, Y. Feng, A. Gilbert, J.F. Fernandes, H.A. Bukhari, A. Wajdan, M.V. Martinez, M.S. Santos, M. Shamohammdi, H. Luo, P. Westphal, P. Leeson, P. DiAchille, V. Gurev, M. Mayr, L. Geris, P. Pathmanathan, T. Morrison, R. Cornelussen, F. Prinzen, T. Delhaas, A. Doltra, M. Sitges, E.J. Vigmond, E. Zacur, V. Grau, B. Rodriguez, E.W. Remme, S. Niederer, P. Mortier, K. McLeod, M. Potse, E. Pueyo, A. Bueno-Orovio, P. Lamata, The 'Digital Twin' to enable the vision of precision cardiology, Eur. Heart J. (2020).

[2] N.P. Smith, A.D. McCulloch, D.J. Paterson, What can modelling provide to cardiac physiology? J. Physiol. 590 (Pt 18) (2012) 4401.

[3] R. Chabiniok, V.Y. Wang, M. Hadjicharalambous, L. Asner, J. Lee, M. Sermesant, E. Kuhl, A.A. Young, P. Moireau, M.P. Nash, et al., Multiphysics and multiscale modelling, data–model fusion and integration of organ physiology in the clinic: ventricular cardiac mechanics, Interface Focus 6 (2) (2016) 20150083.

[4] A. Krishnamurthy, C.T. Villongco, J. Chuang, L.R. Frank, V. Nigam, E. Belezzuoli, P. Stark, D.E. Krummen, S. Narayan, J.H. Omens, A.D. McCulloch, R.C. Kerckhoffs, Patient-specific models of cardiac biomechanics, J. Comput. Phys. 244 (2013) 4–21.

[5] R. Avazmohammadi, M. Hill, M. Simon, M. Sacks, Transmural remodeling of right ventricular myocardium in response to pulmonary arterial hypertension, APL Bioeng. 1 (1) (2017) 016105.

[6] H. Schmid, W. Wang, P.J. Hunter, M.P. Nash, A finite element study of invariant-based orthotropic constitutive equations in the context of myocardial material parameter estimation, Comput. Methods Biomech. Biomed. Eng. 12 (6) (2009) 691–699.

[7] R. Avazmohammadi, E.A. Mendiola, J.S. Soares, D.S. Li, Z. Chen, S. Merchant, E.W. Hsu, P. Vanderslice, R.A.F. Dixon, M.S. Sacks, A computational cardiac model for the adaptation to pulmonary arterial hypertension in the rat, Ann. Biomed. Eng. 47 (2019) 138–153.

[8] R. Avazmohammadi, D.S. Li, T. Leahy, E. Shih, J.S. Soares, J.H. Gorman, R.C. Gorman, M.S. Sacks, An integrated inverse model-experimental approach to determine soft tissue three-dimensional constitutive parameters: application to post-infarcted myocardium, Biomech. Model. Mechanobiol. (2017).

[9] D.H. Lin, F.C. Yin, A multiaxial constitutive law for mammalian left ventricular myocardium in steady-state barium contracture or tetanus, J. Biomech. Eng. 120 (4) (1998) 504–517.

[10] M. Sacks, C. Chuong, A constitutive relation for passive right-ventricular free wall myocardium, J. Biomech. 26 (11) (1993) 1341–1345.

[11] J.D. Humphrey, R.K. Strumpf, F.C. Yin, Determination of a constitutive relation for passive myocardium: I. A new functional form, J. Biomech. Eng. 112 (3) (1990) 333–339.

[12] J.D. Humphrey, R.K. Strumpf, F.C. Yin, Determination of a constitutive relation for passive myocardium: II. Parameter estimation, J. Biomech. Eng. 112 (3) (1990) 340–346.

[13] A. Horowitz, Y. Lanir, F.C. Yin, M. Perl, I. Sheinman, R.K. Strumpf, Structural three-dimensional constitutive law for the passive myocardium, J. Biomech. Eng. 110 (3) (1988) 200–207.

[14] S.E. Moskowitz, Constitutive stress-strain relations for the myocardium in diastole, J. Biomech. 18 (1985) 177–187.

[15] l. Demer, r. Strumpf, f. Yin, Biaxial constitutive relations of passive myocardium, Biophys. J. 41 (1983) 245.

[16] G.A. Holzapfel, R. Ogden, Constitutive modelling of passive myocardium: A structurally based framework for material characterization, Philos. Trans. Ser. A Math. Phys. Eng. Sci. 367 (1902) (2009) 3445–3475.

[17] J. Criscione, Constitutive framework optimized for myocardium and other high-strain, laminar materials with one fiber family, J. Mech. Phys. Solids 50 (8) (2002) 1681–1702.

[18] R. Avazmohammadi, M.R. Hill, M.A. Simon, W. Zhang, M.S. Sacks, A novel constitutive model for passive right ventricular myocardium: evidence for myofiber-collagen fiber mechanical coupling, Biomech. Model. Mechanobiol. 16 (2017) 561–581.

[19] R. Avazmohammadi, D. Li, T. Leahy, E. Shih, J. Soares, J. Gorman, R. Gorman, M. Sacks, An integrated inverse model-experimental approach to determine soft tissue three-dimensional constitutive parameters: Application to post-infarcted myocardium, Biomech. Model. Mechanobiol. (2017) 1–23.

[20] R. Avazmohammadi, J.S. Soares, D.S. Li, S.S. Raut, R.C. Gorman, M.S. Sacks, A contemporary look at biomechanical models of myocardium, Annu. Rev. Biomed. Eng. 21 (2019) 417–442.

[21] D.S. Li, R. Avazmohammadi, S.S. Merchant, T. Kawamura, E.W. Hsu, J.H. Gorman III, R.C. Gorman, M.S. Sacks, Insights into the passive mechanical behavior of left ventricular myocardium using a robust constitutive model based on full 3D kinematics, J. Mech. Behav. Biomed. Mater. 103 (2020) 103508.

[22] E.R. Pfeiffer, J.R. Tangney, J.H. Omens, A.D. McCulloch, Biomechanics of cardiac electromechanical coupling and mechanoelectric feedback, J. Biomech. Eng. 136 (2) (2014) 021007.

[23] T.S. Eriksson, A.J. Prassl, G. Plank, G.A. Holzapfel, Modeling the dispersion in electromechanically coupled myocardium, Int. J. Numer. Method Biomed. Eng. 29 (11) (2013) 1267–1284.

[24] N.A. Trayanova, J.J. Rice, Cardiac electromechanical models: from cell to organ, Front. Physiol. 2 (2011) 43.

[25] R.C.P. Kerckhoffs, S.N. Healy, T.P. Usyk, A.D. McCulloch, Computational methods for cardiac electromechanics, Proc. IEEE 94 (4) (2006) 769–783.

[26] P. Hunter, M. Nash, G. Sands, Computational electromechanics of the heart, Comput. Biol. Heart 12 (1997) 347–407.

[27] A. Minajeva, M. Kulke, J.M. Fernandez, W.A. Linke, Unfolding of titin domains explains the viscoelastic behavior of skeletal myofibrils, Biophys. J. 80 (3) (2001) 1442–1451.

[28] H. Granzier, M. Kellermayer, M. Helmes, K. Trombitas, Titin elasticity and mechanism of passive force development in rat cardiac myocytes probed by thin-filament extraction, Biophys. J. 73 (4) (1997) 2043–2053.
[29] S.A. Thompson, C.R. Copeland, D.H. Reich, L. Tung, Mechanical coupling between myofibroblasts and cardiomyocytes slows electric conduction in fibrotic cell monolayers, Circulation 123 (19) (2011) 2083–2093.
[30] F. Sheikh, A. Raskin, P.H. Chu, A. Domenighetti, S. Lange, M. Zheng, X.Q. Liang, T. Zhang, T. Yajima, Y.S. Gu, N.D. Dalton, G.W. Dorn, A.D. McCulloch, J.H. Omens, J. Heller-Brown, K.L. Peterson, J. Chen, A novel stretch effector within the cardiomyocyte sarcomere is critical to sense biomechanical stress responses leading to pathological hypertrophy, Circulation 118 (18) (2008) S394.
[31] 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.
[32] T. Bui-Thanh, K. Willcox, O. Ghattas, Model reduction for large-scale systems with high-dimensional parametric input space, SIAM J. Sci. Comput. 30 (6) (2008) 3270–3288.
[33] C.G. Bucher, U. Bourgund, A fast and efficient response surface approach for structural reliability problems, Struct. Saf. 7 (1) (1990) 57–66.
[34] T.W. Simpson, T.M. Mauery, J.J. Korte, F. Mistree, Kriging models for global approximation in simulation-based multidisciplinary design optimization, AIAA J. 39 (12) (2001) 2233–2241.
[35] I. Bilionis, N. Zabaras, B.A. Konomi, G. Lin, Multi-output separable Gaussian process: Towards an efficient, fully Bayesian paradigm for uncertainty quantification, J. Comput. Phys. 241 (2013) 212–239.
[36] D. Xiu, G.E. Karniadakis, The Wiener–Askey polynomial chaos for stochastic differential equations, SIAM J. Sci. Comput. 24 (2) (2002) 619–644.
[37] I. Babuška, F. Nobile, R. Tempone, A stochastic collocation method for elliptic partial differential equations with random input data, SIAM Rev. 52 (2) (2010) 317–355.
[38] F. Scarselli, A.C. Tsoi, Universal approximation using feedforward neural networks: A survey of some existing methods, and some new results, Neural Netw. 11 (1) (1998) 15–37.
[39] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. 378 (2019) 686–707.
[40] E. Weinan, B. Yu, The deep Ritz method: a deep learning-based numerical algorithm for solving variational problems, Commun. Math. Stat. 6 (1) (2018) 1–12.
[41] V.M. Nguyen-Thanh, X. Zhuang, T. Rabczuk, A deep energy method for finite deformation hyperelasticity, Eur. J. Mech. A Solids 80 (2020) 103874.
[42] J. He, L. Li, J. Xu, C. Zheng, ReLU deep neural networks and linear finite elements, 2018, arXiv preprint arXiv:1807.03973.
[43] P. Hajela, L. Berke, Neural networks in structural analysis and design: An overview, Comput. Syst. Eng. 3 (1) (1992) 525–538, High-Performance Computing for Flight Vehicles.
[44] R.K. Tripathy, I. Bilionis, Deep UQ: Learning deep neural network surrogate models for high dimensional uncertainty quantification, J. Comput. Phys. 375 (2018) 565–588.
[45] Y. Zhu, N. Zabaras, P.-S. Koutsourelakis, P. Perdikaris, Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data, J. Comput. Phys. 394 (2019) 56–81.
[46] T. O'Leary-Roseberry, U. Villa, P. Chen, O. Ghattas, Derivative-informed projected neural networks for high-dimensional parametric maps governed by PDEs, 2020.
[47] S. Sheriffdeen, J.C. Ragusa, J.E. Morel, M.L. Adams, T. Bui-Thanh, Accelerating PDE-constrained inverse solutions with deep learning and reduced order models, 2019.
[48] H. Goh, S. Sheriffdeen, J. Wittmer, T. Bui-Thanh, Solving Bayesian inverse problems via variational autoencoders, 2021.
[49] H.V. Nguyen, T. Bui-Thanh, Model-constrained deep learning approaches for inverse problems, 2021.
[50] L. Sun, H. Gao, S. Pan, J.-X. Wang, Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data, Comput. Methods Appl. Mech. Engrg. 361 (2020) 112732.
[51] Y. Zhu, N. Zabaras, Bayesian deep convolutional encoder–decoder networks for surrogate modeling and uncertainty quantification, J. Comput. Phys. 366 (2018) 415–447.
[52] H. Gao, L. Sun, J.-X. Wang, PhyGeoNet: Physics-informed geometry-adaptive convolutional neural networks for solving parameterized steady-state PDEs on irregular domain, J. Comput. Phys. 428 (2021) 110079.
[53] D.S. Li, R. Avazmohammadi, C.B. Rodell, E.W. Hsu, J.A. Burdick, J.H. Gorman III, R.C. Gorman, M.S. Sacks, How hydrogel inclusions modulate the local mechanical response in early and fully formed post-infarcted myocardium, Acta Biomater. 114 (2020) 296–306.
[54] S. G., A. Schriefl, M. Andra, M. Sacherer, C. Viertler, H. Wolinski, G. Holzapfel, Biomechanical properties and microstructure of human ventricular myocardium, Acta Biomater. (2015).
[55] O. Gültekin, H. Dal, G.A. Holzapfel, On the quasi-incompressible finite element analysis of anisotropic hyperelastic materials, Comput. Mech. 63 (3) (2019) 443–453.
[56] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[57] W.J. Morokoff, R.E. Caflisch, Quasi-Monte Carlo integration, J. Comput. Phys. 122 (2) (1995) 218–230.
[58] J.V. Dillon, I. Langmore, D. Tran, E. Brevdo, S. Vasudevan, D. Moore, B. Patton, A. Alemi, M. Hoffman, R.A. Saurous, Tensorflow distributions, 2017, arXiv preprint arXiv:1711.10604.
[59] J. Bradbury, R. Frostig, P. Hawkins, M.J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, Q. Zhang, JAX: composable transformations of Python+NumPy programs, 2018.

[60] T. Hughes, The Finite Element Method: Linear Static and Dynamic Finite Element Analysis, Dover Publications, 2000.
[61] A. Griewank, A. Walther, Algorithm 799: revolve: an implementation of checkpointing for the reverse or adjoint mode of computational differentiation, ACM Trans. Math. Softw. 26 (1) (2000) 19–45.
[62] J. Nocedal, S. Wright, Numerical Optimization, Springer Science & Business Media, 2006.
[63] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, Q. He, A comprehensive survey on transfer learning, Proc. IEEE 109 (1) (2020) 43–76.
[64] J.S. Soares, D.S. Li, E. Lai, J.H. Gorman, R.C. Gorman, M.S. Sacks, Modeling of myocardium compressibility and its impact in computational simulations of the healthy and infarcted heart., in: Functional Imaging and Modeling of the Heart : ... International Workshop, FIMH ..., Proceedings. FIMH 10263, 2017, pp. 493–501.
[65] H. Liu, J.S. Soares, J. Walmsley, D.S. Li, S. Raut, R. Avazmohammadi, P. Iaizzo, M. Palmer, J.H. Gorman, R.C. Gorman, M.S. Sacks, The impact of myocardial compressibility on organ-level simulations of the normal and infarcted heart, Sci. Rep. 11 (1) (2021) 1–15.
[66] M.S. Sacks, Incorporation of experimentally-derived fiber orientation into a structural constitutive model for planar collagenous tissues, J. Biomech. Eng. 125 (2) (2003) 280–287.
[67] W. Zhang, S. Ayoub, J. Liao, M.S. Sacks, A meso-scale layer-specific structural constitutive model of the mitral heart valve leaflets, Acta Biomater. 32 (2016) 238–255.