
USF Tampa Graduate Theses and Dissertations

USF Graduate Theses and Dissertations

June 2021

A Constitutive-Based Deep Learning Model for the Identification of Active Contraction Parameters of the Left Ventricular Myocardium

Igor Augusto Paschoalotte Nobrega
University of South Florida

Follow this and additional works at: <https://digitalcommons.usf.edu/etd>

 Part of the [Biomechanics Commons](#), and the [Computer Sciences Commons](#)

Scholar Commons Citation

Nobrega, Igor Augusto Paschoalotte, "A Constitutive-Based Deep Learning Model for the Identification of Active Contraction Parameters of the Left Ventricular Myocardium" (2021). *USF Tampa Graduate Theses and Dissertations*.

<https://digitalcommons.usf.edu/etd/9202>

This Thesis is brought to you for free and open access by the USF Graduate Theses and Dissertations at Digital Commons @ University of South Florida. It has been accepted for inclusion in USF Tampa Graduate Theses and Dissertations by an authorized administrator of Digital Commons @ University of South Florida. For more information, please contact digitalcommons@usf.edu.

A Constitutive-Based Deep Learning Model for the Identification of
Active Contraction Parameters of the Left Ventricular Myocardium

by

Igor Augusto Paschoalotte Nobrega

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Mechanical Engineering
Department of Mechanical Engineering
College of Engineering
University of South Florida

Major Professor: Wenbin Mao, Ph.D.
Nathan Gallant, Ph.D.
Shaun Canavan, Ph.D.
Yu Sun, Ph.D.

Date of Approval:
June 23, 2021

Keywords: Constitutive modeling, Hyperelastic material, Machine learning, Cardiac mechanics, Clinical metrics

Copyright © 2021, Igor Augusto Paschoalotte Nobrega

Dedication

I would like to convey my heartfelt gratitude to my family and friends, who have been vital in inspiring and encouraging me to advance my educational endeavors and complete my master's degree.

Acknowledgments

I want to express my gratitude to Dr. Wenbin Mao, my supervisor, not only for providing this opportunity, but also for the encouragement that has enabled me to progress in my engineering career.

Table of Contents

List of Tables	iii
List of Figures	iv
Abstract.....	viii
Chapter 1: Introduction	1
Chapter 2: Theory	5
2.1 Anatomy.....	5
2.1.1 Cardiovascular System.....	5
2.1.2 Left Ventricular Myocardium	7
2.1.2.1 Morphology and Structure	7
2.1.2.2 Mechanical Behavior and Active Contraction.....	10
2.1.2.3 Clinical Metrics of the Ventricular	12
2.1.2.3.1 Wall Thickening	13
2.1.2.3.2 Left Ventricular Torsion	13
2.1.2.3.3 Longitudinal Shortening	14
2.1.2.3.4 Radial Shortening	15
2.2 Cardiac Cycle.....	15
2.2.1 Features of the Ventricular Cycle	17
2.2.1.1 Stroke Volume	18
2.2.1.2 Ejection Fraction.....	18
2.2.1.3 Cardiac Output.....	19
2.2.1.4 Stroke Work.....	19
2.2.1.5 Coupling Ratio.....	19
2.2.1.6 Arterial Elastance.....	19
2.3 Constitutive Model of the Myocardium.....	20
2.3.1 Background.....	20
2.3.2 Constitutive Passive Model.....	22
2.3.3 Constitutive Active Model.....	25
2.4 Neural Networks and Deep Machine Learning.....	26
2.4.1 Special Neural Network Architectures	31
2.4.1.1 Recurrent Networks	31
2.4.1.2 Autoencoder Networks	33
Chapter 3: Constitutive Model of Myocardium	36
3.1 Passive Model	36
3.1.1 Identification of Material Parameters	37
3.1.2 Validation.....	37
3.2 Active Model	39
3.2.1 Validation.....	39
3.3 Geometrical Model	40

3.3.1 Fiber Field.....	43
3.3.2 Mesh Independence	46
3.3.3 Boundary Conditions	49
Chapter 4: Constitutive-based Deep Learning Model.....	51
4.1 Data Acquisition and Manipulation	51
4.1.1 General Myocardium Features Under Specified Pressure-Volume Values	51
4.1.2 Data Acquisition for PV Loop	54
4.2 PVLOOPED.....	55
4.2.1 Model Architecture	57
4.2.2 Training and Validating.....	59
4.2.3 Testing	62
4.3 CMM4	63
4.3.1 Model Architecture	63
4.3.2 Training, Validating and Testing	65
4.4 GM4	67
4.4.1 Model Architecture	67
4.4.2 Training, Validating and Testing	67
4.5 GWM9	70
4.5.1 Custom Layers	72
4.5.2 Model Architecture	74
4.5.3 Training and Validating	76
Chapter 5: Summary of the Model and its Effectiveness.....	80
References.....	86
Appendix A: Copyright Permissions	90

List of Tables

Table 2.1 Invariants and stresses when applying shear along a given plane in a specified direction using local fiber coordinated system according to the derivations proposed by Holzapfel and Ogden.....	23
Table 3.1 Material parameters used within the passive constitutive model of the myocardium	37
Table 3.2 Results of mesh independence study	47
Table 4.1 Summary of dataset containing clinical metrics of the myocardium when exposed to linear loading ranging from 0 to 16kPa, gamma varying from 0 to 0.25, and fiber orientations ranging from 40 to 80 degrees in the epicardium direction and from -40 to -80 degrees in the endocardium direction	52
Table 4.2 Evaluation of PVLOOPED model using testing dataset.....	61
Table 4.3 Comparison of true and predicted values of gamma using GM4 model in random testing values	70
Table 4.4 Evaluation of GWM9 model during training and validating when using best parameters identified by the hyperparameter study	79
Table 5.1 Evaluation of study	84

List of Figures

Figure 2.1 Anatomic illustration of the heart and blood flow.....	6
Figure 2.2 Representation of fiber distribution along the ventricular myocardium	8
Figure 2.3 Fiber direction in a myocardium	10
Figure 2.4 Experimental data for amount of shear.....	11
Figure 2.5 Estimated biaxial loading responses in the E_{ff} and E_{ss} directions.....	12
Figure 2.6 Sample vector used to compute left ventricular torsion	14
Figure 2.7 Pressure and volume relationships in a cardiac cycle.....	16
Figure 2.8 Diagram of major pressure-volume loop relationships.....	18
Figure 2.9 Classification of Machine Learning models	28
Figure 2.10 Illustration of a Perceptron	29
Figure 2.11 Common activation functions and their derivatives	29
Figure 2.12 The schematics of a LSTM cell	33
Figure 2.13 General architecture of an Autoencoder model	35
Figure 3.1 Comparison of analytical and simulated values for shear tests in different fiber orientations.....	38
Figure 3.2 Comparison of analytical and simulated values of biaxial loading in E_{ff} and E_{ss} directions.....	38
Figure 3.3 Comparison of analytical (dots) and FE (lines) results of the active contraction model for shear tests along different combinations with γ value of 0.05 (a) and 0.11 (b)	39
Figure 3.4 Modeling dimensions for idealized ellipsoid with a truncated base and a thick wall of the left ventricular myocardium.....	40
Figure 3.5 Mesh representation of the idealized myocardium geometry composed of only hexahedral elements (a and b) and with pentahedral elements around the apex (c and d)	41

Figure 3.6 Features of the all-hexahedral mesh used in this study (a) with 6 cross-sectional layers and pure hexahedral elements in the apex region (c)	42
Figure 3.7 Meshing representation of a realistic model of the left ventricular myocardium.....	43
Figure 3.8 Fiber distribution of 65.697 and -69.612 degrees in the endocardium and epicardium directions respectively.....	44
Figure 3.9 Fiber distribution of 59.810 and -71.693 degrees in the endocardium and epicardium directions respectively.....	45
Figure 3.10 Comparison of the distribution of z component of f_0 using all-hexahedral mesh (a) and with pentahedral elements at the apex (b) using fiber directions of 65.697 and -69.612 degrees in the endocardium and epicardium directions respectively.....	46
Figure 3.11 Representative differences in mesh densities used in the mesh study (left) and a sample tetrahedral mesh (right)	47
Figure 3.12 Average of stress values greater than 1% of maximum value for corresponding mesh configurations (kPa).....	48
Figure 3.13 Maximum total displacement node-wise for corresponding mesh configurations (kPa)	49
Figure 3.14 Boundary conditions used in all simulations performed with idealized myocardium.....	50
Figure 4.1 Sample data augmentation of volumetric fraction.....	53
Figure 4.2 Sample distribution of gamma values along different ranges of endocardium and epicardium fiber directions	53
Figure 4.3 Sample PV loop data source (a) used for point-wise data extraction (b) and later augmented.....	54
Figure 4.4 Random generated PV loops based on the proposed generation algorithm with (a) default setting, (b) volume bounded to (None, 1.0) and (c) initial and maximum volumes are set to 1.0.....	55
Figure 4.5 Sample PVLOOPED input with highlighted features	56
Figure 4.6 Schematics of (a) training and (b) final model for the PVLOOPED autoencoder model.....	57
Figure 4.7 PVLOOPED implemented model structure with respective layer connection information.....	59
Figure 4.8 Training and validating (left) loss and (right) MSE of the PVLOOPED model	60

Figure 4.9 Comparison between PV loops taken from testing dataset and generated by PVLOOPEd	62
Figure 4.10 Generating PV loops using cardiovascular data with PVLOOPEd model	63
Figure 4.11 CMM4 implemented model structure with respective layer connectgions information	64
Figure 4.12 Training and validating (left) loss and (right) MSE of the CMM4 model	65
Figure 4.13 Comparison of true and predicted values in testing set of the CMM4 model	66
Figure 4.14 GM4 implemented model structure with respective layer connections information	68
Figure 4.15 MSE values found during hyperparameter tunning of the GM4 model	69
Figure 4.16 Training and validating (left) loss and (right) MSE of the GM4 model.....	70
Figure 4.17 Schematics of a Gated Residual Layer with details of its main building block	72
Figure 4.18 Schematics of a ‘Warm-start’ Recurrent Neural Network layer wrapper.....	73
Figure 4.19 General schematic of the GWM9 model	74
Figure 4.20 GWM9 implemented model structure with respective layer connections information	75
Figure 4.21 Training schematics of the GWM9 model illustrating the usage of CMM4 and GM4 models	77
Figure 4.22 Validation and training losses of gama waveform (top) and fiber orientation (bottom).....	77
Figure 5.1 General schematic of PVLOOPEd and GWM9 model usage	80
Figure 5.2 Procedure for evaluating the constitutive model parameters estimated by our deep learning model using the results obtained by FEA simulations	81
Figure 5.3 Sample comparison (1) of results from constitutive model FEA simulation using predicted gamma waveform from GWM9 model and ‘true’ value of a PV loop generated using PVLOOPEd model.....	82
Figure 5.4 Sample comparison (2) of results from constitutive model FEA simulation using predicted gamma waveform from GWM9 model and ‘true’ value of a PV loop generated using PVLOOPEd model.....	82
Figure 5.5 Sample comparison (3) of results from constitutive model FEA simulation using predicted gamma waveform from GWM9 model and ‘true’ value of a PV loop generated using PVLOOPEd model.....	83

Figure 5.6 Sample comparison (4) of results from constitutive model FEA simulation using predicted gamma waveform from GWM9 model and ‘true’ value of a PV loop generated using PVLOOPED model..... 83

Abstract

Modern breakthroughs in biomedical engineering, computer science, and data mining have created new opportunities for detecting important mechanical properties of soft tissues that can be employed to identify possible signs of diseases or physiological difficulties. However, the scarcity of different mechanical properties obtained through noninvasive testing emphasizes the importance of incorporating authentic biological data into computer models capable of replicating the behavior of soft tissues.

The field of continuum theory of large deformation hyperactivity permits the formulation of highly descriptive mathematical research and computational models capable of perfectly describing the minute mechanical characteristics of soft materials. By including features about the tissue's morphology into its internal constitution, constitutive models effectively associate applied loading to the material's mechanical function, allowing for accurate reinterpretation and analysis of tissue behavior.

The advancement of sophisticated analytics techniques, such as machine learning and high-performance computational science, has sparked interest in data-driven computational modeling to extract fast and valid observations of complicated systems. Additionally, machine learning approaches are proving beneficial in a range of biological applications.

In this study, we present a physics-based deep learning approach for predicting material parameters for the active constitutive model based exclusively on a few clinical parameters. This model is also capable of predicting the fiber orientation of epicardium and endocardium of the myocardium wall which are used in an algorithm for computing the fiber distribution through a given geometry. The data used to train the deep learning model was gathered by finite element simulations developed with state-of-the-art passive and active constitutive modeling of the myocardium. Moreover, a variational autoencoder

is presented and utilized to produce realistic PV loops based on set of cardiovascular metrics for testing our final model.

The results demonstrate that the deep learning model can estimate active material parameters and fiber orientations with high accuracy for any specified conditions. Moreover, the constitutive model was able to replicate material characteristics based on experimental data presented in the literature and follow expected behavior under diverse pre-set cardiac cycles. Further research is suggested to optimize the proposed deep learning model and its applicability to patient-specific scenarios to account for passive properties and non-idealized fiber distribution by extending its training dataset with data obtained by clinical procedures.

Chapter 1: Introduction

Heart disease remains the major cause of mortality worldwide. According to the American Heart Association (AHA), the frequency of cardiovascular diseases (CVD) surpasses 48% among young people (aged 20 to 30) and worsens with aging [1]. This statistic exceeds the number of lives lost to various forms of cancer and other respiratory disorders, which account for roughly one in every three fatalities in the United States [1]. To feasibly avert an unmanageable escalation in such numbers, it is crucial to understand various elements of the human heart, as well as other biological tissues. Developments in biomedical engineering, computer science, and data mining have opened new avenues for identifying critical mechanical characteristics of soft tissues and their effect on their physiological activities.

The elastic characteristics of living tissue is used to identify potential indicators of illness or physiological complications. Complex structures underlying soft tissues are inextricably linked to the water content within their structural configurations, which comprise most of their incompressibility and hyperplastic nature [2]. Additionally, the existence of certain pathologies can modify the water content of tissues fluids, impacting the microstructure activity of such fluids [2]. These microstructures are critical predictors of the tissue's mechanical responses, whereas material characteristics and internal microstructure are strongly associated [3]. Thus, the mechanical properties of the material are utilized to identify areas of inflammation, edema, hypertrophy, and fibrosis produced by the presence of disease processes [4].

However, assessing mechanical properties is a difficult task owing to the distinctive biological composition inherent in the preponderance of these biological tissues, such as anisotropy and incompressibility, as well as the diverse physiological processes, mechanical stresses, and boundary conditions. Furthermore, differing materials exhibit divergent behavioral patterns in active and passive

environments, which not only subjects them to residual stresses with undetermined values, but also considerably increases their mechanical response variance [5].

The Young's modulus and shear modulus of elasticity are the fundamental mechanical principles that govern a material's mechanical behavior. These are frequently determined using uniaxial tensile and compression tests, which determine the stress-strain relationship [2] [3]. Furthermore, given the difficulty of conducting intricate tests on weakly soft materials, most researchers resort on simple tension, planar biaxial, and inflation ascertain the material's mechanical characteristics [5]. Likewise, the tensile test is the most viable technique for physically small samples due to its simplicity and adaptability.

Unfortunately, solely mechanical characteristics along the tensile axis are recorded, drastically reducing the amount of data gathered necessary for a comprehensive assessment. To overcome this limitation, tensile stress can be employed in a variety of directions, for example, along the fibers or across the fibers of the tissue [5]. However, these experiments are often intrusive and not applied to patient-specific scenarios. The paucity of diverse mechanical characteristics derived through noninvasive testing highlights the vital need of combining genuine biological data into computer models geared at recreating the material's behavior.

The domain of continuum theory of large deformation hyperactivity enables the founding of highly descriptive mathematical studies and computational models worthy of precisely characterizing the microscopic mechanical properties underlying soft materials. Interestingly, over multiple decades, the formulation of continuum-based constitutive relationships has been a focus of active research, allowing for the investigation of a variety of clinical methodologies entailing soft biological tissues [3]. By incorporating details about the morphology of the tissue into its internal constitution, structural continuum constitutive models can associate imposed compressive forces with the material's mechanical function, allowing for reliable reinterpretation and investigation of tissue behavior [3].

Moreover, the left ventricle is the heart's main chamber and is responsible for pumping blood throughout the body, entailing a higher level of pressure than all other chambers. Subsequently, the

ventricular myocardium, as explained by Holzapfel and Ogden in [6], is the functional tissue of the heart's wall. Consequently, modeling this particular region of the heart is critical for comprehending its whole fundamental mechanical properties.

Nevertheless, despite various models that have been devised to simulate both passive and active contraction of myocardial tissues, most of them are undesirable from a stability standpoint, considering their strain energy functions are not guaranteed to reach convergence [7]. Further, running constitutive simulations normally demand significant processing resources and time, limiting their use in clinical applications; specially when certain material parameters are not patient-specific and might not reflect real behavior if several simulations are not run to find the best-fitting parameters.

The innovation of advanced analytics, including machine learning techniques and high-performance computational science, aroused interest in data-driven computational modeling as a means of finding rapid and reliable observations of complex processes [8]. Through repeated training on high-quality experimental data, informational models are able to decipher hidden patterns between input and output [4]. In addition, due to the immediacy inherent in this approach, complex systems are not predicated entirely on a predefined set of equations, thereby reducing the incidence of superfluous constraints. Further, machine learning techniques are edifying a variety of biomedical applications. For example, it has been successfully used to interpret electrocardiograms, diagnose breast cancer and melanomas, optimize hip implant geometries, and treat cardiovascular diseases [5].

Taking this into account, it is feasible to optimize patient-specific parameters by utilizing constitutive-based machine learning models. We present a schematic strategy for training a deep learning model to learn a multivariate function that maps clinical metrics to the constitutive parameters regulating the active contraction of the left ventricular myocardium. More importantly, our model learns from a set of constitutive based results.

Our deep learning model is composed of two sets of data: (1) statistics specific to the myocardium, such as wall thickening, left ventricular torsion, longitudinal shortening, and radial

shortening; and (2) a specified cardiac cycle represented by a pressure-volume loop. By retrieving pertinent knowledge from the input data, our model uncovers the best-fitting pair of fiber orientations (in both epicardium and endocardium directions) and quantifies an active contraction-specific parametric waveform that can be used in conjunction with the constitutive model to attain desired mechanical behavior.

Moreover, our model derives its knowledge from a set of constitutive-based data attained from the Finite Element Analysis (FEA) with customized model implementation based on the work of Holzapfel and Ogden [6] and Pezzuto, Ambrosi and Quarteroni [9]. Additionally, we present a modified variational autoencoder capable of generating realistic PV loops in response to a set of specified cardiovascular features.

Chapter 2: Theory

2.1 Anatomy

2.1.1 Cardiovascular System

The Cardiovascular system has the significant function of delivering a steady supply of the oxygen, water, glucose, vitamins, and other required nutrients to each cell of the human body in order to support life. Furthermore, this system is also responsible for carrying and delivering hormones to target cells, transporting waste products, created by metabolism processes of each cell, and regulating the body's temperature [10]. Throughout this complex system, more than 2,000 gallons of blood stream for over 12,000 miles through the specialized blood vessels every day [11].

The cardiovascular system is composed of the heart and the connecting vasculature, from aorta to arterioles to capillaries to veins [10]. Arteries are blood vessels with thick walls responsible for carrying oxygen-rich blood from the left side of the heart into all regions of the body. The blood flowing within them normally moves at high velocities, which yields high pressure on the artery walls and causes the artery to expand during heart contraction [11]. In contrast, veins are comprised of thin-walled tubes, and are focused on returning the blood to the heart, after all processes within a cell are completed. As a result, blood normally flows at slow velocities within veins and, therefore, does not exert high pressures on its walls [11].

The heart is a muscular organ divided into four chambers: left and right atrium and left and right ventricles, which are illustrated in Figure 2.1. The cellular wall that holds these chambers are mainly composed by striated muscle tissue layered in three structures: a thick layer of myocardium located in between the endocardium (inner structure) and the epicardium (outer structure) [12]. The walls of the atrium chambers are thinner, as it only handles incoming low-pressured deoxygenated blood flow. Each

atrium is connected to its respective ventricles by atrioventricular valves (AV valves), which operates based on the pressure difference between the atrium and the ventricle: when the pressure in the first exceeds the second, the valves are open and the ventricles are filled [10]. Moreover, papillary muscles and chordae tendineae compose the connection between AV valves and ventricles, which, in order to prevent blood from returning to the atrium, contracts along the ventricle contraction [10]. The four main valves are named tricuspid, pulmonary, mitral, and aortic. The first valve connects the right atrium to the right ventricle, while the second protects the outlet from the right ventricle to the pulmonary artery. The mitral valve is responsible for controlling the blood flow between the left atrium and the left ventricle, while the aortic valve prevents counter blood flow from the aorta.

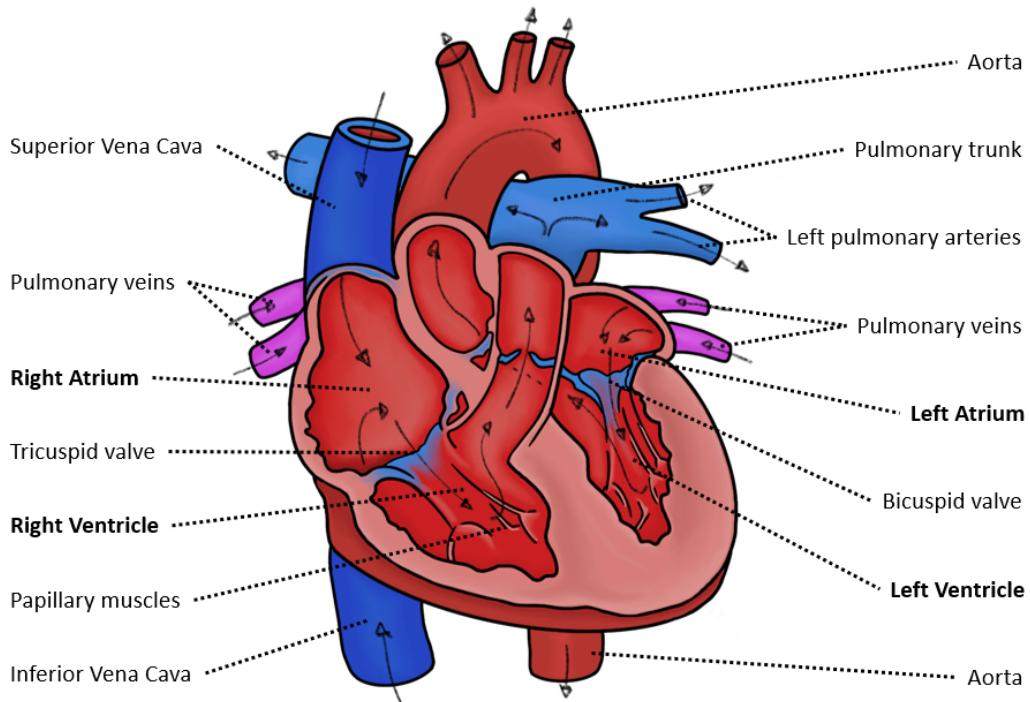


Figure 2.1 Anatomic illustration of the heart and blood flow.

In a regular cycle, deoxygenated blood flows into the right atrium through the inferior vena cavae and moves passively into the right ventricle through the tricuspid valve following a simple pressure gradient. In turn, this blood is pumped through the pulmonary valve and through the pulmonary artery to

the lungs. In this low-pressure pulmonary circulation, blood is exposed to air in the alveoli causing the release of carbon dioxide and the bind of oxygen. The now oxygenated blood continues through the pulmonary veins and into the left atrium through pulmonary veins. Similar pressure gradient behavior occurs between the left atrium and left ventricle, allowing for oxygenated blood to fill the left ventricle through the mitral valve. Through a spiral-like contraction, the left ventricle generates enough force for the blood to be pumped through the aortic valve and to the aorta artery and the entire systemic circulation [10].

2.1.2 Left Ventricular Myocardium

As previously discussed, the left ventricular myocardium is responsible for pumping blood throughout the entire systemic circulation, leading it to be the focus of many studies related to cardiovascular diseases. Moreover, the structure of the heart, particularly the anisotropic cardiac microstructure, remains a continuous discussion [13] [6]. One methodology portrays the heart as a single muscle winding in a helical fashion, whereas the other, used in this study, characterizes the heart as a continuum composed of laminar sheets [6].

2.1.2.1 Morphology and Structure

The left ventricle is the main chamber of the heart and is dedicated to pumping blood for the entire body, demanding the support of greater pressure than the right ventricle. As a result of the increased pressure, the left ventricle's wall thickness is greater than that of the right ventricle. Moreover, the left ventricle's wall thickness and curvature differ spatially; it is thickest at the base and equator and thinnest at the apex [6]. Additionally, the wall thickness and curvature of the heart differ temporally throughout the cardiac cycle as a result of the material's contraction and sheet orientations. Furthermore, the left ventricular wall can be thought of as a continuous network of layered myocardial fibers with a smooth transmural difference in their orientations. Also, it is equitably well modelled as a thick-walled revolution ellipsoid with a truncated base [6].

The heart wall is composed of three distinct layers: an inner layer known as the endocardium, a middle layer, denoted as myocardium, and an outer layer, called the epicardium. Figure 2.2a underlines the composition of the ventricular wall. The endocardium covers the interior of the four chambers. It is a serous membrane with a thickness of about 100 μm and is composed primarily of epimysial collagen, elastin, and a layer of endothelial cells that functions as an interfacial layer between the wall and the blood. The outermost layer, the epicardium, is also a membrane of about 100 μm in thickness, composed primarily of epimysial collagen and some elastin [6].

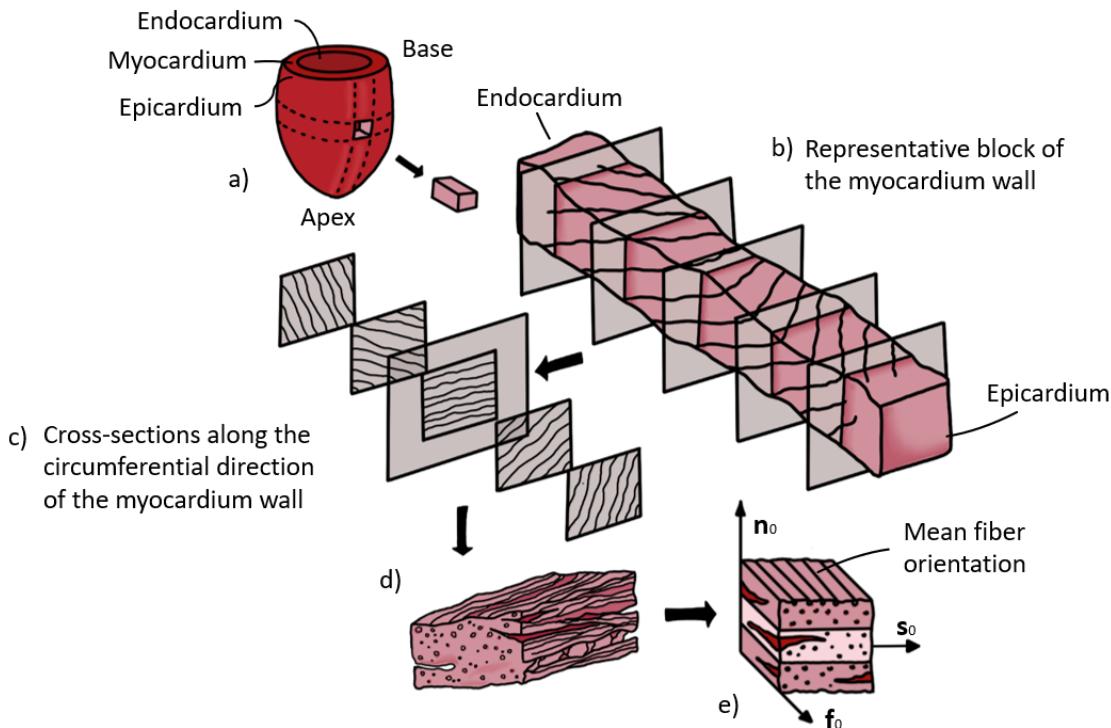


Figure 2.2 Representation of fiber distribution along the ventricular myocardium (adapted from [6])

As described in [6], the ventricular myocardium is the functional tissue of the heart's wall, with emphasis on the left ventricle, which is responsible for creating the pressure head for blood circulation throughout the body [10]. The majority (about seventy percent) of the left ventricle's wall structure is composed of parallel myocytes partitioned into branching layers by extensive cleavage planes [14], often

described by layered sheets. A study performed in [15], observed a repeated pattern of myocyte aggregation in sections parallel to their long axis. The remainder is composed of different interstitial elements, with collagen occupying just 2–5% of the interstitial volume. The collagen is organized in a spatial network that establishes lateral connections between neighboring muscle fibers, with attachments near the sarcomere's z-line [6]. Notably, as these proteins are primarily organized along the muscle cell's longitudinal axis, allowing for low resistance conduction between two or more cells [10]. Moreover, ventricular myocytes are long and thin muscle cells [13] with only one nucleus [10]. Their exact measurements vary according to age and cardiac region but are usually between 50 μm and 150 μm in length and 10 μm and 20 μm in diameter [13]. In addition, as myocytes are mostly extended cells, they have a primary orientation defined by the cell's long axis, and their ellipsoidal myocardial nuclei are aligned with this axis [13].

Furthermore, it is crucial to understand the transmural shift of the myocardial tissue structure since this complex architecture ensures the resilience of the heart to bending and twisting during the cardiac cycle. As Figure 2.2b and 2.2c illustrate, the transmural variation of layer orientations is linearly dispersed along longitudinal-circumferential sections. The prevailing muscle fiber orientation rotates from +50° to +70° (sub-epicardial region) to approximately 0° (mid-wall region) to -50° to -70° (sub-endocardial region) with respect to the circumferential direction of the left ventricle [6]. Additionally, the authors of [15] conducted an experiment that demonstrated the epicardial and the endocardial fibers of the ventricular wall are orientated with the angles of their long axes almost at 90° to one another. It was also shown that consecutive endocardial fibers transverse in an anticlockwise pattern from the apex to the base, whereas preceding epicardial fibers rise in a clockwise direction [15]. It is important to notice that, even though this is frequently assumed in the literary work, the layers are not necessarily parallel with vessel walls, as seen in Figure 2.2b [6]. In fact, only a subset of cells are partitioned along their long axis with alignment parallel to the epicardium across prolonged lengths [15]. To better visualize how the fiber orientation is obtained, Figure 2.3 highlights the main axis of an idealized myocardium geometry.

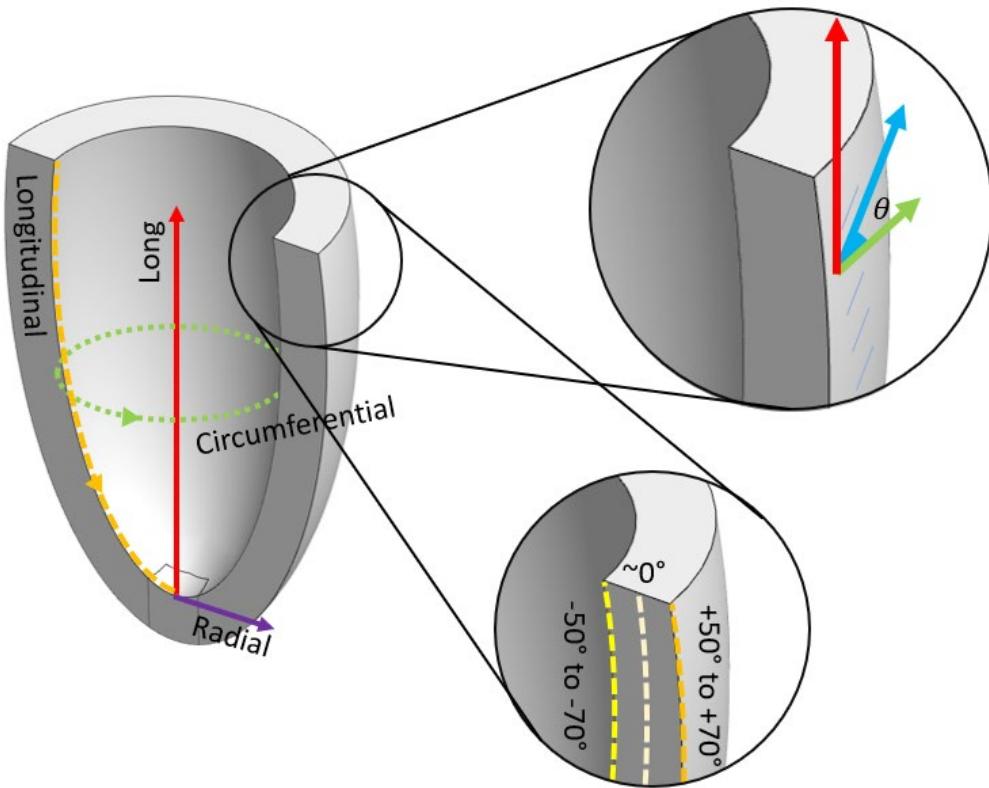


Figure 2.3 Fiber direction in a myocardium

The mathematical representation of these layered structures is defined by a right-handed orthonormal set of basis vectors and a correlated orthogonal curvilinear system of coordinates. The local fixed set of (unit) basis vectors is made up of fiber axis f_0 that corresponds to the orientation of muscle fiber, a sheet axis s_0 that is defined as perpendicular to the direction of the fiber direction (also known as the direction of cross-fiber sheet), and a sheet-normal axis n_0 that is orthogonal to the two others [6].

2.1.2.2 Mechanical Behavior and Active Contraction

Shear deformation, or the relative sliding of myocardial layers, is believed to be critical for the heart's mechanical operation [14]. In fact, as discussed by the authors of [16] and mentioned in [14], there is evidence that this process contributes to the thickening of the subendocardial wall during systole, thus facilitating left ventricular ejection. Similarly, gradual wall thinning that arises within passive ventricular filling is aligned with myocardial layer reorientation or shearing [14]. As a result, the tissue shows a

strongly nonlinear, regionally, and temporally, dependent behavior with reduced hysteresis as well as a directionally dependent softening as the strain rises [6]. Provided that the myocardium is an orthotropic material represented at any juncture by three mutually orthogonal planes, its nonlinear mechanical activity results in distinct material responses for each plane [6].

This behavior is readily identifiable based on the experimental data gathered in [14] and shown in Figure 2.4 adopted from [6], who also explored it. As it can be seen, ventricular myocardium is the least sensitive to simple shear in the **fn** and **sn** planes, respectively, for shear in the **f** and **s** fields. On the other hand, the material is intrinsically subject to shear deformations that cause the myocyte (**f**) axis to extend in the **fs** and **fn** planes. Moreover, the planes **f** and **s** exhibit unequal shear responses along their respective axes, while the **n** plane does not manifest such behavior.

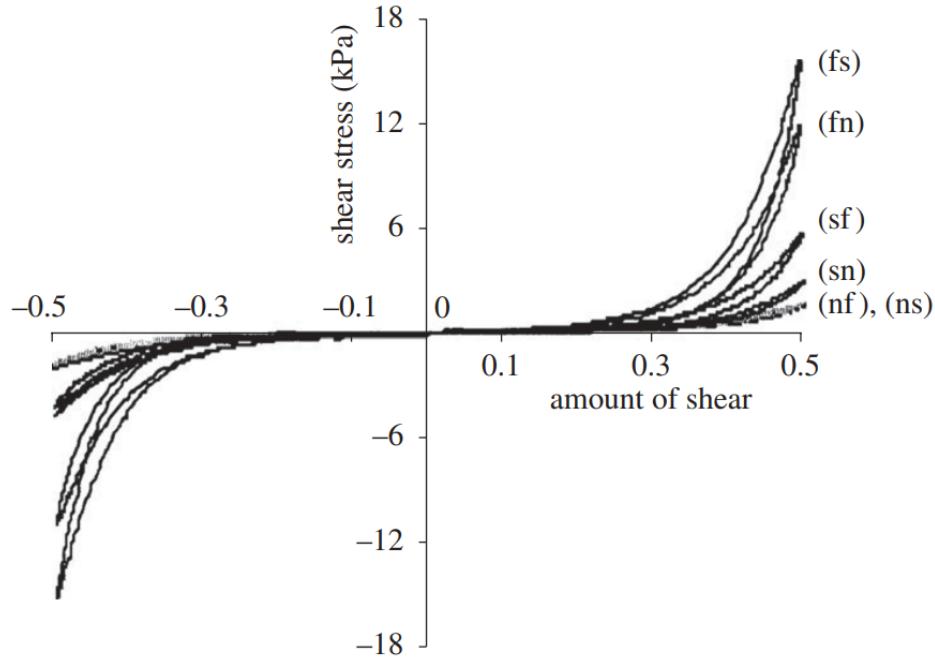


Figure 2.4 Experimental data for amount of shear [6]

Furthermore, the authors of [6] retrieve sample biaxial test plots from the data gathered and published in [17]. It illustrates three distinct loading regimens for biaxial loading in the **fs** plane of a canine left ventricular myocardium and it is shown in Figure 2.5 (extracted from [6]). As explained by the

authors, Figure 2.5a exhibits the Cauchy stress S_{ff} in the fiber direction as a function of the engineering strain strain E_{ff} in the same direction; Figure 2.5b displays equivalent curves in the sheet direction (S_{ss} against E_{ss}). As with the shear response, the biaxial data reveal a significant degree of nonlinearity and anisotropy [6].

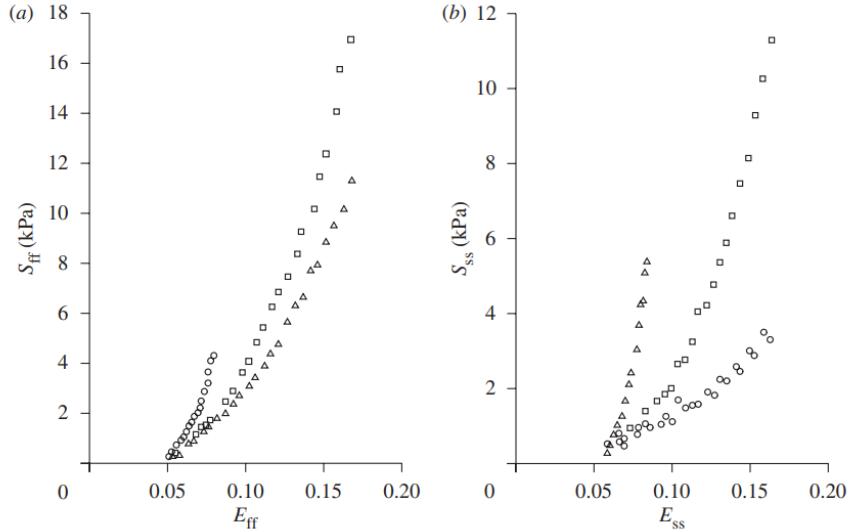


Figure 2.5 Estimated biaxial loading responses in the E_{ff} and E_{ss} directions [6]

Furthermore, contrary to other hyperelastic inert materials, a fundamental and inherent property of living tissues is their ability to effectively deform in the absence of external loads [18]. An electrically stimulated myocyte sustained at a constant length produces an active response that encodes the forces produced at the microscale, as described by an isometric test discussed by the authors of [9]. Several ionic channels in the myocardium have been shown to be activated by cell stretch and volume shifts in experimental tests performed by [19] and [20]. Additionally, cardiac muscle fibers contract in response to active mechanisms, which are primarily triggered by the release of intracellular calcium at the microscopic stage [18].

2.1.2.3 Clinical Metrics of the Ventricular Deformation

The behavior of the left ventricle's deformation is commonly described by clinical metrics defined by a series of geometrical parameters. These physical values are explained in this section and are

later used as comparison values for model validity. It is also worth noting that the given metrics are normally calculated using only two reference points (generally, data acquired at end-diastole and end-systole); however, in order to extend the capability of the proposed neural network model and capture relevant training data purely from simulation runs, each definition was extended to work at each timestep throughout the data acquisition process (see data acquisition section for further details).

2.1.2.3.1 Wall Thickening

It is assumed that the thickening of the ventricular wall during systole plays a significant role in the overall contractile effect. Generally, clinical metrics use the end-diastolic and end-systolic wall thickness as reference, which allows to quantify the ratio of wall thicknesses. In addition, the work described in [21] reveals normal ranges for wall thickness fraction of 52%.

$$W_{fraction} = \frac{W_{end-diastole} - W_{end-systole}}{W_{end-diastole}} \quad (2.1)$$

Moreover, as later discussed in our results, in order to minimize numerical instability, we measure wall thicknesses fraction at two distinct locations: at the apex and at the base. At the apex, we take into consideration **n** number of nodes on the endocardium and epicardium that corresponds to the same longitudinal axis as reference points. Similarly, we use **m** number of nodes located at the circumference of the endocardium and epicardium base as references to compute the wall thicknesses from the base. In both cases we compute the wall thicknesses fraction individually and take the overall average.

2.1.2.3.2 Left Ventricular Torsion

Left ventricular torsion is defined as the helical twisting effect due to the layered fiber orientation. It occurs along the longitudinal axis and is described using the right-handed rule (positive values are counterclockwise). The amount of torsion is associated with ventricular contraction is known to be critical

to the overall function of the left ventricle and it is typically measured based on the absolute difference between the end-systole and end-diastole.

To compute the ventricular torsion, we used, as reference, the coordinates of the apex node and, from it, randomly assigned vectors to n nodes located at the epicardium surface. At each timestep, as the geometry contracts and rotates, we compared the step angle shift with their respective reference configuration. Figure 2.6 illustrates this process, where a reference vector V at initial state at time t is compared with its next configuration at time $t+1$ by the angle θ ; as the two vectors do not significantly shift in x and y at their endpoints, the angle between them closely represents the amount of torsion T .

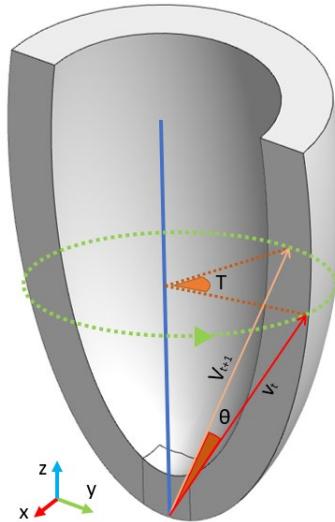


Figure 2.6 Sample vector used to compute left ventricular torsion [6]

The total torsion was taken as the average of the computed values from all n reference vectors and it is described by equation 2.2.

$$T_{rel} = \frac{\sum \left| \frac{\Delta T}{T_i} \right|}{n} \quad (2.2)$$

2.1.2.3.3 Longitudinal Shortening

Longitudinal shortening is commonly defined as the change in perpendicular distance between the apex (lowest region) to the central base (topmost region), denoted as ΔL . In this study, we use the

relative ventricular longitudinal shortening, which is defined as the relative difference between change in L with respect to its initial value:

$$LS_{rel} = \left| \frac{\Delta L}{L_i} \right| \quad (2.3)$$

2.1.2.3.4 Radial Shortening

Another relevant clinical metric that is used to characterize the left ventricle's contraction is its radial shortening, which is calculated based on the difference between aperture radiiuses, ΔR of base region and the inner radius R_i . The inner radius is calculated as the distance from the center basal node to the endocardium surface, R_i . We make use of the relative radial shortening, which compares the change in radius with its original value R_i :

$$RS_{rel} = \left| \frac{\Delta R}{R_i} \right| \quad (2.4)$$

2.2 Cardiac Cycle

The assessment of cardiac physiology lays the groundwork for determining shifts in the myocardium's anatomic and physiological characteristics [22]. Among the methods for measuring cardiac physiology, there are many options that may be chosen, such as echocardiography, cardiac MRI (Magnetic Resonance Imaging) pressure-volume (PV) catheters. Moreover, the cardiac cycle can be described through pressure-volume loop tests, which are generated by plotting pressure and volume waveforms against one another. These analyses provide detailed, load-independent guidance on the cardiac activity and the heart's natural and diseased states. Moreover, PV loops can be used to assess a variety of physiologically important hemodynamic parameters, including stroke volume, cardiac output, ejection fraction, and myocardial contractility. A sample PV loop, along with its respective pressure and volume waveforms, is illustrated in Figure 2.7.

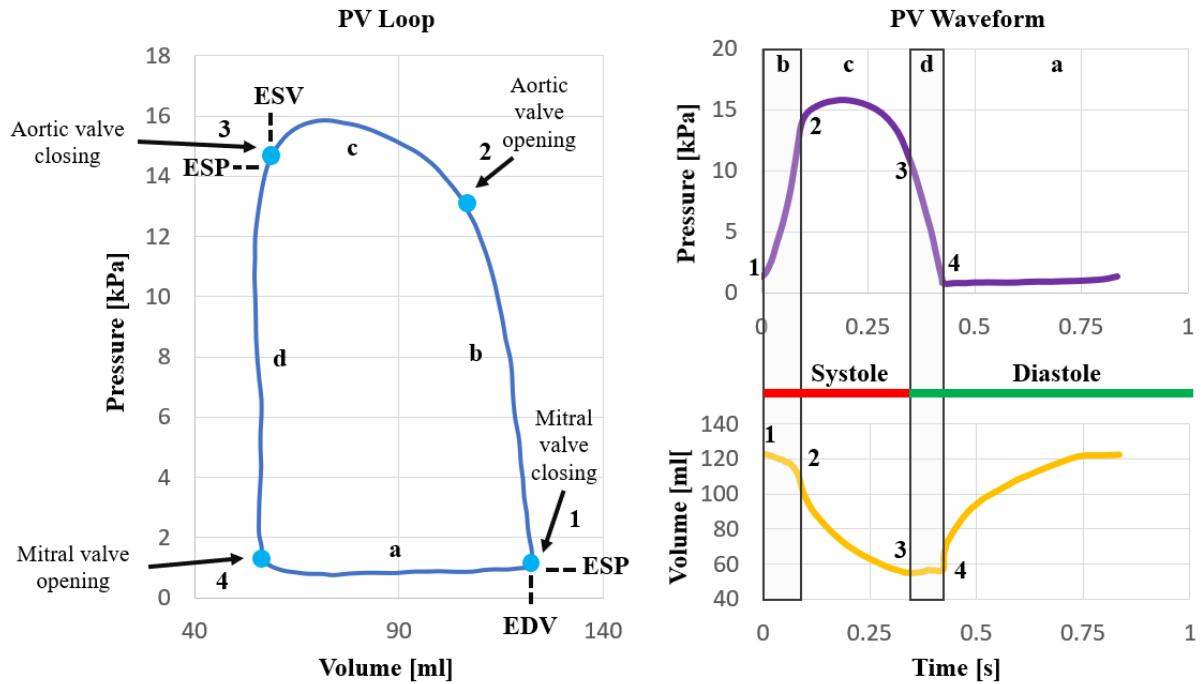


Figure 2.7 Pressure and volume relationships in a cardiac cycle

The cardiac cycle is divided into two main phases: systole and diastole and it is illustrated by the red and green bars on top of Figure 2.7. Systole is the time interval between the muscle's fully relaxed state and the moment of maximum mechanical activation. Diastole is the time interval during which the muscle returns to its resting state from the end-systolic (mostly activated) state [23]. Moreover, different mechanical events occur during the cardiac cycle as the pressure in the ventricular chamber increases, causing blood to flow into and out of the ventricle. Hence, we can describe the cardiac cycle by monitoring variations in the ventricle's pressures and volumes.

Additionally, the cardiac cycle can be categorized into four phases: filling of the ventricles (region *a* in figure 2.7), isovolumetric contraction (region *b* in figure 2.7), ejection (region *c* in figure 2.7), and isovolumetric relaxation (region *d* in figure 2.7).

During the initial phase of the ventricular filling, the tricuspid and mitral valves stay open, while the aortic and pulmonary valves remain closed, allowing for blood to flow passively from the veins into the ventricles. The ventricles fill in approximately 0.45 seconds, and the process is completed by the atria

contracting during the final 0.12 seconds [24]. The amount of blood remaining in the left ventricle at the completion of ventricular filling is referred to as the end-diastolic volume (EDV), which in an adult person is about 120 mL, while the analogous pressure, referred as the end-diastolic pressure (EDP), is approximately 4–7 mmHg [24].

The next phase, systole, is divided into two principal stages. First, there is a short isovolumetric contraction, which lasts approximately 0.05 seconds; it is then followed by a longer ejection phase that lasts approximately 0.30 seconds [24]. As the isovolumetric contraction occurs, the left ventricular pressure rises above the atrial pressure, closing the mitral valve and producing the first cardiac sound. When left ventricular pressure reaches aortic pressure at the end of isovolumetric contraction, the aortic valve opens [24].

At the ejection phase, blood is expelled from the ventricle, as the ventricular pressure exceeds the pulmonary artery pressure, the aortic and pulmonary valves open. When the aortic valve clenches, the cardiac cycle's ejection process ends, triggering the heart's second sound [24]. In addition, this entire process lasts for approximately 0.30 seconds. The amount of blood remaining in the left ventricle at the conclusion of the ejection process, referred to as the end-systolic volume (ESV), is approximately 50 mL [24].

The last phase involves the isovolumetric relaxation of the heart. Throughout this phase, the aortic and pulmonary valve are closed, impeding any fluid to flow through them. This process continues until the intraventricular pressure drops below that of the atria, at which point the mitral and tricuspid valves reopen [24]. The isovolumetric relaxation takes approximately 0.08 seconds.

2.2.1 Features of the Ventricular Cycle

PV loops can be used to evaluate a number of physiologically significant hemodynamic parameters (as shown in Figure 2.8) and provide insight into cardiac function and the normal and pathological states of the heart. A few of these metrics are used to both train and evaluate a deep learning model to generate patient-like cardiac cycles and to validate our constitutive model.

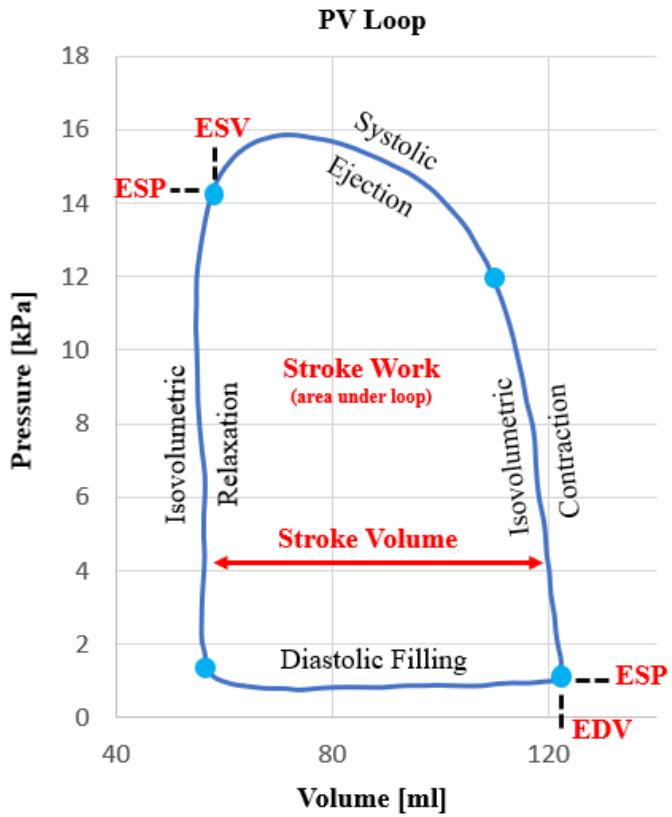


Figure 2.8 Diagram of major pressure-volume loop relationships

2.2.1.1 Stroke Volume

The volume of blood ejected by a ventricle during a single contraction is called the stroke volume. It is calculated based on the difference between the end diastolic (EDV) and end systolic volumes (ESV).

$$SV = EDV - ESV \quad (2.5)$$

2.2.1.2 Ejection Fraction

The ejection fraction is calculated as the ratio of the amount of blood expelled from the ventricle per beat (stroke volume) to the volume of blood remaining in the ventricle at the end of diastole.

$$EF(\%) = \frac{SV}{EDV} \cdot 100 \quad (2.6)$$

According to the European Association of Cardiovascular Imaging (EACVI) consensus, ventricles in good health usually have an ejection fraction greater than 52% for males and greater than 54% for females [25].

2.2.1.3 Cardiac Output

Cardiac output is characterized as the volume of blood pumped by the ventricle per unit time and, therefore, it is a function of stroke volume and heart rate (HR).

$$CO = SV \cdot HR \quad (2.7)$$

2.2.1.4 Stroke Work

The ventricular stroke work is characterized as the work done by the ventricle in the process of expelling the stroke volume into the aorta or pulmonary artery and it is approximated as the area covered by the PV loop.

2.2.1.5 Coupling Ratio

The relationship between the left ventricle and the arterial system is widely recognized as a critical factor affecting overall cardiovascular efficiency [26]. The coupling ratio indicates the efficiency with which power is transferred from the ventricle to the peripheral vasculature.

$$CR = \frac{SV}{ESV} \quad (2.8)$$

2.2.1.6 Arterial Elastance

Arterial Elastance is a parameter used to describe the arterial load and its effect on the ventricle. It is calculated as the ratio of the end-systolic pressure to the volume of the stroke.

$$Ea = \frac{ESP}{ESV} \quad (2.9)$$

2.3 Constitutive Model of the Myocardium

A thorough understanding of physiological and pathological load carrying processes is derived from constitutive equations, which encapsulate critical knowledge about the composition and function of the described biological tissue that is seldom represented in solely phenomenological models [3]. By combining conservation laws and kinematics relationships, constitutive models generate stress-strain relationships from which governing equations describing the materials' reaction to various mechanical loading conditions can be derived [5]. A summary of the principal component of the kinematics for constitutive modelling is briefly summarized in the following sub-section. For detailed information, it is suggested to follow the word described in [6] and [27].

2.3.1 Background

The fundamental deformation vector for describing the local kinematics is the deformation gradient \mathbf{F} , and it is defined by equation 2.10, in which \mathbf{X} denotes the original positional vector and \mathbf{x} refers to the same vector displaced after an infinitesimal time $t > 0$.

$$\mathbf{F}(X, t) = \frac{d\mathbf{x}}{d\mathbf{X}} \quad (2.10)$$

From standard notation and convention, we express the Jacobian determinant for the deformation gradient as:

$$J = \det \mathbf{F} > 0 \quad (2.11)$$

and, for an incompressible material, the following constrain is applied:

$$J = \det \mathbf{F} \equiv 1 \quad (2.12)$$

In relationship to \mathbf{F} , the right and left Cauchy-Green tensors are respectively defined by:

$$\mathbf{C} = \mathbf{F}^T \mathbf{F} \text{ and } \mathbf{B} = \mathbf{F} \mathbf{F}^T \quad (2.13)$$

The Green-Lagrange strain tensor is defined by:

$$\mathbf{E} = \frac{1}{2}(\mathbf{C} - \mathbf{I}) \quad (2.14)$$

where \mathbf{I} is the identity tensor. Finally, the principal isotropic invariants of \mathbf{C} , and, consequently of \mathbf{B} , are defined by

$$\mathbf{I}_1 = \text{tr } \mathbf{C}, \mathbf{I}_2 = \frac{1}{2}[\mathbf{I}_1^2 - \text{tr}(\mathbf{C}^2)] \text{ and } \mathbf{I}_3 = \det \mathbf{C} \quad (2.15)$$

For anisotropic materials with preferred directions from its reference configuration, denoted as \mathbf{a}_0 , two additional transversely isotropic invariants are defined by:

$$\mathbf{I}_4 = \mathbf{a}_0 \cdot (\mathbf{C}\mathbf{a}_0) \text{ and } \mathbf{I}_5 = \mathbf{a}_0 \cdot (\mathbf{C}^2\mathbf{a}_0) \quad (2.16)$$

For materials with two preferred directions, with the second being denoted by \mathbf{b}_0 , the following variants must be defined:

$$\mathbf{I}_6 = \mathbf{b}_0 \cdot (\mathbf{C}\mathbf{b}_0) \text{ and } \mathbf{I}_5 = \mathbf{b}_0 \cdot (\mathbf{C}^2\mathbf{b}_0) \quad (2.17)$$

These are associated with the coupling invariant, denoted by:

$$\mathbf{I}_8 = \mathbf{a}_0 \cdot (\mathbf{C}\mathbf{b}_0) = \mathbf{b}_0 \cdot (\mathbf{C}\mathbf{a}_0) \quad (2.18)$$

Moreover, strain energy density functions ψ define a relationship between the amount of energy required to deform a volume unit of a solid and the strain applied. As mentioned in [6], strain energy functions are directly related to the deformation gradient \mathbf{F} through \mathbf{C} , ensuring objectivity. For an elastic material the Cauchy stress tensor σ is defined by the following equation:

$$J\sigma = \mathbf{F} \frac{\partial \psi}{\partial \mathbf{F}} = \mathbf{F} \frac{\partial \psi}{\partial \mathbf{E}} \mathbf{F}^T \quad (2.19)$$

This formulation can be slightly altered to accommodate compressible materials, where ψ is a function of \mathbf{F} and \mathbf{E} by implementing the Lagrange multiplier ρ [6], given by:

$$\boldsymbol{\sigma} = \mathbf{F} \frac{\partial \psi}{\partial \mathbf{F}} - \rho \mathbf{I} = \mathbf{F} \frac{\partial \psi}{\partial \mathbf{E}} \mathbf{F}^T - \rho \mathbf{I} \quad (2.20)$$

Lastly, as explained by [6], considering a strain energy function composed by a list of invariants the two previous equations are extended to:

$$J\boldsymbol{\sigma} = \mathbf{F} \sum_{i=1}^N \psi_i \frac{\partial \mathbf{I}_i}{\partial \mathbf{F}} \quad \text{and} \quad \boldsymbol{\sigma} = \mathbf{F} \sum_{i=1, i \neq 3}^N \psi_i \frac{\partial \mathbf{I}_i}{\partial \mathbf{F}} - \rho \mathbf{I} \quad (2.21)$$

where \mathbf{I}_i are the invariants, ψ_i are the derivatives of strain energy function with respect to each \mathbf{I}_i .

2.3.2 Constitutive Passive Model

The constitutive model for the left ventricular myocardium used in this study is based on the analysis of Holzapfel and Ogden introduced in [6]. It is worth noting that the literature identifies the model as one of the most precise representation of the material's mechanical responses. Whereas the formulation of the material's defining equation is diligently explained in [6], we will outline main principles from their analysis in this section for consistency.

Bearing in mind the orthotropic property of the ventricular myocardium, the material's three mutually orthogonal axes can be defined by the following vectors when considering simple shear γ under these fiber orientations:

$$\mathbf{f}_0 = [1 \ 0 \ 0]^T, \ \mathbf{s}_0 = [0 \ 1 \ 0]^T \text{ and } \mathbf{n}_0 = [0 \ 0 \ 1]^T \quad (2.22)$$

With the fiber, sheet (cross-fiber), and sheet-normal (normal) directions defined, we now assess the invariant I_4 aligned with each of these orientations:

$$\mathbf{I}_{4f} = \mathbf{f}_0 \cdot (\mathbf{C}\mathbf{f}_0), \mathbf{I}_{4s} = \mathbf{s}_0 \cdot (\mathbf{C}\mathbf{s}_0), \text{ and } \mathbf{I}_{4n} = \mathbf{n}_0 \cdot (\mathbf{C}\mathbf{n}_0) \quad (2.23)$$

Moreover, there are invariants of coupling aligned with the pairs of direction defined by:

$$\mathbf{I}_{8fs} = \mathbf{I}_{8sf} = \mathbf{f}_0 \cdot (\mathbf{C}\mathbf{s}_0), \mathbf{I}_{8fn} = \mathbf{I}_{8nf} = \mathbf{f}_0 \cdot (\mathbf{C}\mathbf{n}_0) \text{ and } \mathbf{I}_{8sn} = \mathbf{I}_{8ns} = \mathbf{s}_0 \cdot (\mathbf{C}\mathbf{n}_0) \quad (2.24)$$

In addition, let

$$\mathbf{B} = \mathbf{F}\mathbf{F}^T, \mathbf{f} = \mathbf{F}\mathbf{f}_0 \text{ and } \mathbf{s} = \mathbf{F}\mathbf{s}_0 \quad (2.25)$$

Following the reasoning described in [6], the results of shear being applied in the fs plane in both s0 and s0 direction, as well as the shear applied along the sn plane in the s0 and n0 directions and the shear applied in the fn plane in the f0 and n0 directions are summarized in table 2.1.

Table 2.1 Invariants and stresses when applying shear along a given plane in a specified direction using local fiber coordinated system according to the derivations proposed by Holzapfel and Ogden [6].

Plane	Direction	Invariants	Shear stress
fs	\mathbf{f}_0	$I_{4s} = 1 + \gamma^2$ $I_{4f} = I_{4n} = 1$ $I_{8fs} = \gamma, I_{8fn} = I_{8sn} = 0$	$\sigma_{12} = 2\gamma(\psi_1 + \psi_{4s})$ $\sigma_{13} = \sigma_{23} = 0$
	\mathbf{s}_0	$I_{4f} = 1 + \gamma^2$ $I_{4s} = I_{4n} = 1$ $I_{8fs} = \gamma, I_{8fn} = I_{8sn} = 0$	$\sigma_{12} = 2\gamma(\psi_1 + \psi_{4f})$ $\sigma_{13} = \sigma_{23} = 0$
	\mathbf{s}_0	$I_{4n} = 1 + \gamma^2$ $I_{4f} = I_{4s} = 1$ $I_{8sn} = \gamma, I_{8fs} = I_{8fn} = 0$	$\sigma_{23} = 2\gamma\psi_1$ $\sigma_{12} = \sigma_{13} = 0$
sn	\mathbf{s}_0	$I_{4s} = 1 + \gamma^2$ $I_{4f} = I_{4n} = 1$ $I_{8sn} = \gamma, I_{8fs} = I_{8fn} = 0$	$\sigma_{23} = 2\gamma(\psi_1 + \psi_{4s})$ $\sigma_{12} = \sigma_{13} = 0$
	\mathbf{n}_0	$I_{4s} = 1 + \gamma^2$ $I_{4f} = I_{4n} = 1$ $I_{8sn} = \gamma, I_{8fs} = I_{8fn} = 0$	$\sigma_{23} = 2\gamma(\psi_1 + \psi_{4f})$ $\sigma_{12} = \sigma_{13} = 0$
	\mathbf{f}_0	$I_{4n} = 1 + \gamma^2$ $I_{4f} = I_{4s} = 1$ $I_{8fn} = \gamma, I_{8fs} = I_{8sn} = 0$	$\sigma_{13} = 2\gamma\psi_1$ $\sigma_{12} = \sigma_{23} = 0$
fn	\mathbf{n}_0	$I_{4f} = 1 + \gamma^2$ $I_{4s} = I_{4n} = 1$ $I_{8fn} = \gamma, I_{8fs} = I_{8sn} = 0$	$\sigma_{13} = 2\gamma(\psi_1 + \psi_{4f})$ $\sigma_{12} = \sigma_{23} = 0$

According to data shown in [14], the stiffness of the shear response is large when the fiber direction is extended, small when the normal direction is extended and often intermediate when the sheet direction is extended [6]. Additionally, the results show that there are distinctions between (fs) and (fn), as well as between (sf) and (sn), but are unable to differentiate among responses to (nf) and (ns). One

strategy to enhance the modeling for these variations is to use one or more of the coupling invariants in the strain-energy equation, which lead to equation 2.26 when selecting the invariants I_1 , I_2 , I_3 , I_{4f} , I_{4s} , I_{8fs} and I_{8fn} .

$$\mathbf{J}\boldsymbol{\sigma} = 2\psi_1\mathbf{B} + 2\psi_2(\mathbf{I}_1\mathbf{B} - \mathbf{B}^2) + 2I_3\psi_3\mathbf{I} + 2\psi_{4f}\mathbf{f} \otimes \mathbf{f} + 2\psi_{4s}\mathbf{s} \otimes \mathbf{s} + \psi_{8fs}(\mathbf{f} \otimes \mathbf{s} + \mathbf{s} \otimes \mathbf{f}) + \psi_{8fn}(\mathbf{f} \otimes \mathbf{n} + \mathbf{n} \otimes \mathbf{f}). \quad (2.26)$$

However, Ψ ought to be unaffected by the sign sensibility of the invariants I_{8fs} and I_{8fn} , which occur when one of the vectors \mathbf{f}_0 , \mathbf{s}_0 , or \mathbf{n}_0 is reversed. This leads to defining $\widehat{\Psi}(\dots, I_{8fs}^2, \dots) = \Psi(\dots, I_{8fs}, \dots)$ that results in $\psi_{8fs} = 2 \partial \widehat{\Psi} / \partial (I_{8fs}^2) I_{8fs}$, which vanishes in the reference state as long as Ψ behaves properly [6]. Recognizing these requirements and the assumption that the reference configuration is stress-free, equation 2.23 reduces to:

$$2(\psi_1 + 2\psi_2 + \psi_3)\mathbf{I} + 2\psi_{4f}\mathbf{f}_0 \otimes \mathbf{f}_0 + 2\psi_{4s}\mathbf{s}_0 \otimes \mathbf{s}_0 = 0 \quad (2.27)$$

As a result, for an incompressible material, this equation can be substituted by

$$\boldsymbol{\sigma} = 2\psi_1\mathbf{B} + 2\psi_2(\mathbf{I}_1\mathbf{B} - \mathbf{B}^2) - \rho\mathbf{I} + 2\psi_{4f}\mathbf{f} \otimes \mathbf{f} + 2\psi_{4s}\mathbf{s} \otimes \mathbf{s} + \psi_{8fs}(\mathbf{f} \otimes \mathbf{s} + \mathbf{s} \otimes \mathbf{f}) + \psi_{8fn}(\mathbf{f} \otimes \mathbf{n} + \mathbf{n} \otimes \mathbf{f}) \quad (2.28)$$

Further, the meanings of the invariants are used to ascertain which invariants should be included in a specific model. The invariant I_1 , otherwise identified as the isotropic term, is included considering it is assumed to be compatible with the underlying non-collagenous and non-muscular matrix, including fluid properties [6]. Moreover, it is reasonable to use an exponential function of I_{4f} to represent the stiffening behavior in the muscle fibre direction, as shown by observational results. Similar approach is appropriate for I_{4s} since it is associated with the deformation along the sheet direction transverse to the muscle fibers [6]. It is worth mentioning that when the related directions are under tension, these terms greatly contribute to the accumulated energy. However, their involvement is negligible when the fibers are compressed, as the fibers do not support compression. As a result, these terms are used in the energy

equation only if $I_{4sf} > 1$ or $I_{4s} > 1$, as applicable [6]. The preceding factors result in the following strain-energy function:

$$\begin{aligned}\psi = & \frac{a}{2b} \exp[b(I_1 - 3)] \\ & + \sum_{i=f,s} \frac{a_i}{2b_i} \{\exp[b_i(I_{4i} - 1)^2] - 1\} + \frac{a_{fs}}{2b_{fs}} [\exp(b_{fs}I^2_{8fs}) - 1]\end{aligned}\quad (2.29)$$

in which a , b , a_f , a_s , b_f , b_s , a_{fs} and b_{fs} are positive material constants with b values being unitless while a parameters are in units of stress.

2.3.3 Constitutive Active Model

The second part of our constitutive formulation focuses on representing the active contraction that occurs due to some electrical physiologies of the myocardium. The selected constitutive model for the active contraction is based on the work proposed by Pezzuto, Ambrosi and Quarteroni in [9]. While the cardiac muscle's passive activity is interpreted as an orthotropic hyperelastic material, active contraction is described using a multiplicative decomposition of the deformation gradient. The main advantage of this “active strain” method is that it maintains the ellipticity of the stress tensor implicitly while imposing just one additional parameter into the model, making it a simple and effective methodology. In this section, we will briefly present the core concepts necessary to comprehend the logic behind the active constitutive model; for further knowledge on the model's derivation, visit [9].

Consider the decomposition of the deformation gradient tensor in two principal spectrums: an elastic deformation \mathbf{F}_e and an active distortion \mathbf{F}_a , resulting in:

$$\mathbf{F} = \mathbf{F}_e \mathbf{F}_a \quad (2.30)$$

Consequently, the strain energy density is purely defined as a function of \mathbf{F}_e , while the active deformation \mathbf{F}_a is considered to be a result of fully dissipative processes [9]. Now, the purpose is to minimize the overall elastic energy in all feasible variants, as noted by:

$$\min_{\mathbf{u} \in V} \left\{ \int_{\omega_0} \Psi(\mathbf{F}\mathbf{F}_a^T) (\det F_a) dV - \langle \mathbf{f}^{ext}, \mathbf{u} \rangle \right\} \quad (2.31)$$

where ω_0 denotes the reference placement of a body \mathcal{B} modeled as a continuum in \mathbb{R} , such that from reference configuration \mathbf{x}_0 , $\omega_0 = \mathbf{x}_0(\mathcal{B})$. Assuming that a minimum exists, then the first variation of the functional in the previous equation must be null for every viable alternative [9]. Henceforth the first Piola-Kirchhoff stress is:

$$\mathbf{P} = (\det F_a) \frac{\partial \Psi(F_e)}{\partial F_e} F_a^{-T} \quad (2.32)$$

The Cauchy stress tensor can then be perceived as a stimulus that is entirely active:

$$\boldsymbol{\sigma} = (\det F_e)^{-1} \frac{\partial \Psi(F_e)}{\partial F_e} F_e^T = (\det F_a) \frac{\partial \Psi}{\partial F_e} \Big|_{F_e=F_a^{-1}} F_a^{-T} \quad (2.33)$$

As explained by the authors of [28] and mentioned in [9], a physiologically driven option for F_a is a transversely isotropic and locally isochoric tensor, as shown by the fact that myocytes shorten in the sarcomeres direction throughout contraction but do not profoundly alter their volume. Thus, the final form of the active contraction is as follows:

$$\mathbf{F}_a = (1 - \gamma) \mathbf{f} \otimes \mathbf{f} + \frac{1}{1 - \gamma} (\mathbf{I} - \mathbf{f} \otimes \mathbf{f}) \quad (2.34)$$

The coefficient γ is dictated by the internal dynamics specified by some internal parameter that describes the action potential and/or relevant ionic current [9].

2.4 Neural Networks and Deep Machine Learning

At its core, machine learning (ML) is the process in which a model is able to extract patterns and relationships purely from data. In contrast to traditional problem-solving tasks where a set of explicit rules describe how it should behave, machine learning models generally iterate through data and inherently encounters deterministic rules for the problem [29]. Moreover, the advancement of data mining

technologies and high-performance computing, prompted deep learning modeling as a the standard method of accurately and quickly infer predictions from complex processes [8].

One of the main characteristics that differentiates machine learning models is their capacity to enhance their output interactively as new experimental data becomes available, thus increasing the versatility and sustainability of the problem's characterization. Furthermore, machine learning techniques are enlightening diverse applications in the biomedical context. For instance, it was successfully applied in the interpretation of electrocardiograms, diagnosis of breast cancer or melanomas, optimized hip implant geometries and treatment of cardiovascular diseases [5].

Classifying machine learning approaches into broad groups depending on the level and form of supervision during training particularly valuable. The four main groups are: supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning. The first type of models is conditioned along know optimal solutions, commonly referred to as labels. Additionally, the two major supervised learning routines are classification, in which the model learns to cluster data into predefined categories, and regression, wherein the model forecasts a numerical outcome provided a series of inputs. Conversely, unsupervised learning identifies models that train without any labelled results. Typical unsupervised tasks are k-means clustering, isolation forest, and PCA [30]. Semi-supervised learning involves training the algorithm on both labeled and unlabeled data. The last classification is used for models that were trained differently: an agent (or model) observes and interacts within an environment, and it does so by means of a reward system that incentivizes the model depending on form of interaction and learning goal. Figure 2.9 shows various machine learning algorithms according to the form of supervision and architectures employed.

Another criterion used to characterize ML models is whether the algorithm can learn incrementally from a stream of incoming data. If the model can learn with new sets of data by feeding them sequentially, either individually or in small subsets (mini batches) it is characterized as online

learning. On the other hand, if the model can only be trained using all available data, it is described as offline learning [30].

In this study, we focus on the usage of online deep neural network models. More specifically, we design feed-forward models, whereby several layers, each with a large number of neurons, analyze and process data before passing it on to the next layer, resulting in a final layer that outputs a prediction. They were initially inspired by biological neural networks [31] and consist of two major components: a dynamic system of interconnected units (neurons) and an optimization algorithm that often follows a cost function that establishes the amount of error between the input and the desired output.

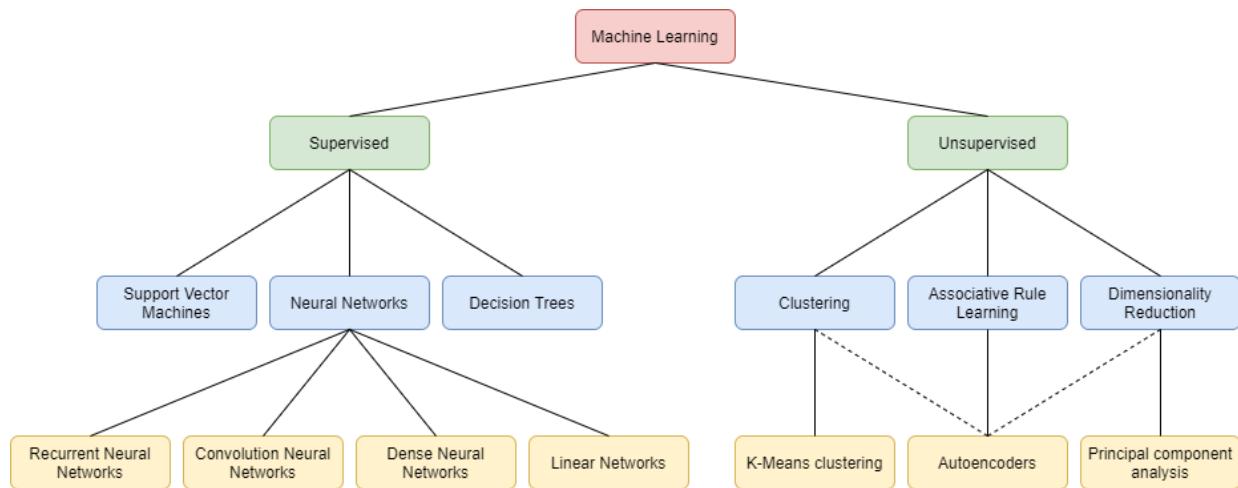


Figure 2.9 Classification of Machine Learning models

Furthermore, the feed-forward network is mainly composed of densely connected perceptrons in three fundamental types of layers: input layers, hidden layers, and output layers. The input layer is the leftmost layer of the network, and the neurons contained within it are referred to as input neurons. The output neurons are located in the rightmost or output layer. The middle layer is referred to as the hidden layer since the neurons in this layer do not act as inputs or outputs. Each connection multiplies the signal from the previous layer by a weight, adds it to a bias, and then passes it through an activation function

[32]. Backpropagation is used in feed-forward networks to iteratively change the network's weights before it reaches the desired output.

Rosenblatt's perceptron is among the most fundamental artificial neural network structures, first introduced in 1957 [30]. It operates by applying an activation function to the weighted sum of the various inputs. Let $\hat{y}(x)$ denote the output of a non-linear activation function $\varphi(x)$, where x represents the array of input features; also, let w represent the array of weights and b the bias term. Then, the output of a perceptron is easily visualized by figure 2.10 and can be expressed as:

$$\hat{y}(x) = \varphi(b + \sum_{i=1}^m x_i w_i) = \varphi(x^T w + b) \quad (2.35)$$

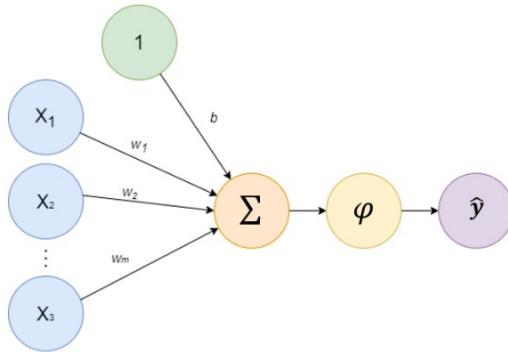


Figure 2.10 Illustration of a Perceptron

An activation function is a non-linear technique that a neuron employs to integrate non-linear properties into a network. Their use is critical for machine learning development as it retains numerical information across layers while allowing the neural network model to inherently develop nonlinear relationships between inputs and outputs [33]. In addition, it is imperative for an activation function to be continuous whose derivatives can be derived at any point. Figure 2.11 illustrates the most common activation functions and their derivatives.

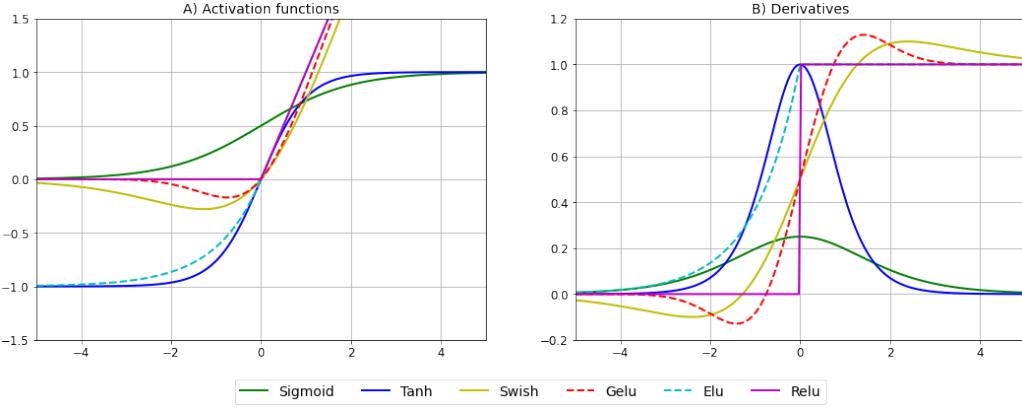


Figure 2.11 Common activation functions and their derivatives

To successfully train a neural network model, the backpropagation algorithm must be considered.

In general, it is analogous to the Gradient Descent algorithm, a generalized optimization algorithm capable of determining the optimum solution to a large variety of problems. The Gradient Descent algorithm is built on the concept of interactively manipulating parameters to minimize a cost function, whereby, in essence, determines the error function's local gradient with respect to parameter vector and proceeds in the course of the descending gradient [30]. Correspondingly, the backpropagation algorithm is capable of computing the gradient of the network's error for each component: connection weights and bias terms, in a single pass. As detailed explained in [33], consider the nested function $\mathcal{F}(\mathbf{x}, \mathbf{w}) = \varphi(\mathbf{v}(\mathbf{x}, \mathbf{w}))$, where $\mathbf{v} = \mathbf{x} \times \mathbf{w}$, its analytical contribution to the backpropagation is expressed by its derivative with respect to \mathbf{x} :

$$\frac{\partial \mathcal{F}}{\partial \mathbf{x}} = \frac{\partial \varphi}{\partial \mathbf{v}}(\mathbf{v}(\mathbf{x}, \mathbf{w})) \times \frac{\partial \mathbf{v}}{\partial \mathbf{x}}(\mathbf{x}, \mathbf{w}) \quad (2.36)$$

As \mathbf{x} represents a matrix, the second term of this equation defines the derivative with respect to each element x . Also, due to the cross product of \mathbf{v} , its output is a scalar number, resulting in:

$$\frac{\partial \mathbf{v}}{\partial \mathbf{x}} = \mathbf{w}^T \quad (2.37)$$

Therefore, the resulting vector from a single perceptron during backpropagation has the form:

$$\frac{\partial \mathcal{F}}{\partial x} = \frac{\partial \varphi}{\partial u} \boldsymbol{v} \times \boldsymbol{w}^T \quad (2.38)$$

To compute the gradient of the entire network, however, through the basis of chain rule, this method is effectively applied to all parameters using computational graph theory. For detailed explanation on how the backpropagation is computed and applied during training of a neural network, refer to [34].

Moreover, the loss of a neural network quantifies the expense of erroneous assumptions and is used during backpropagation to re-evaluate the weights of each neural network layer during the training process. The loss is quantified using a cost function (also known as objective function), which empirically measures the total loss over the entire dataset. A common cost function for models that output continuous real numbers is the Mean Square Error (MSE):

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_{true} - y_{predicted})^2 \quad (2.39)$$

2.4.1 Special Neural Network Architectures

2.4.1.1 Recurrent Networks

Recurrent neural network (RNN) models are used to analyze sequential data. In comparison to pure dense models, they operate on sequences of arbitrary lengths rather than fixed-size inputs. RNNs are capable of capturing long-term associations, preserving knowledge regarding order, and sharing parameters through sequences due to their architecture [34]. A few examples include the interpretation and forecasting of time series data, such as stock values, to determine when to purchase or sell, and to project vehicle trajectories and assist automated driving systems in avoiding accidents [30].

In principle, RNNs act in the same way to feedforward neural networks, with the addition of backward-pointing connections. At each time step t , a recurrent unit (RU), the simplest element of an RNN, accepts the inputs of x_t and its own output from the previous iteration y_{t-1} . Correspondingly, each RU has two sets of weights: one for the inputs w_x and one for the previous iteration step's output w_y . In

addition, the component of a neural network that maintains a state over time steps is referred to as a memory cell. Obeying the rationale in [30], the output of a single recurrent layer is defined by:

$$y_t = \varphi(\mathbf{w}_x^T \mathbf{x}_t + \mathbf{w}_y^T \mathbf{y}_{t-1} + b) \quad (2.40)$$

However, the simple RNN models have a significant disadvantage known as the vanishing gradient dilemma and unstable gradient problem. The later can be resolved by implementing good parameter initialization, faster optimizers, dropout, and layer normalization [30]. However, the problem with the vanishing gradients arises when the same weights are used to measure y_t at each timeframe during training, causing the error signal to get larger or smaller when it steps backwards in the backpropagation phase [30]. This suggests that the network loses temporal information over time, thus having trouble to memorize terms that are further down the sequence and, therefore, encouraging decisions to be made mainly in dependance of most recent data, contradicting its primary intent.

To overcome this problem, Long Short-Term Memory (LSTM) cells was proposed by the authors of [35] and improved over the year by different researchers. The cell's central principle is that the network should learn which data to retain overall and which to eliminate, as well as which data to retrieve from it [30]. A schematic representation of the LSTM cell is shown on figure 2.12. As can be seen, a long-term state stream, denoted by \mathbf{c}_t , filters out unnecessary items through a "forget" gate before inserting new memories collected through the "input" gate. Additionally, a duplicate of the long-term memory is forwarded through the tanh activation mechanism and filtered through the "output" gate, yielding the short-term state \mathbf{h}_t , which is identical to the cell's output, \mathbf{y}_t . Similarly, the other incoming data streams, \mathbf{h}_{t-1} and \mathbf{x}_t , are densely connected. The principal layer is the one that outputs \mathbf{g}_t , which analyzes the interactions between the current and previous states (short-term) and preserves pertinent data that passes through the input gate. The remaining layers serve as gate controls, with outputs varying between 0 and 1 [30].

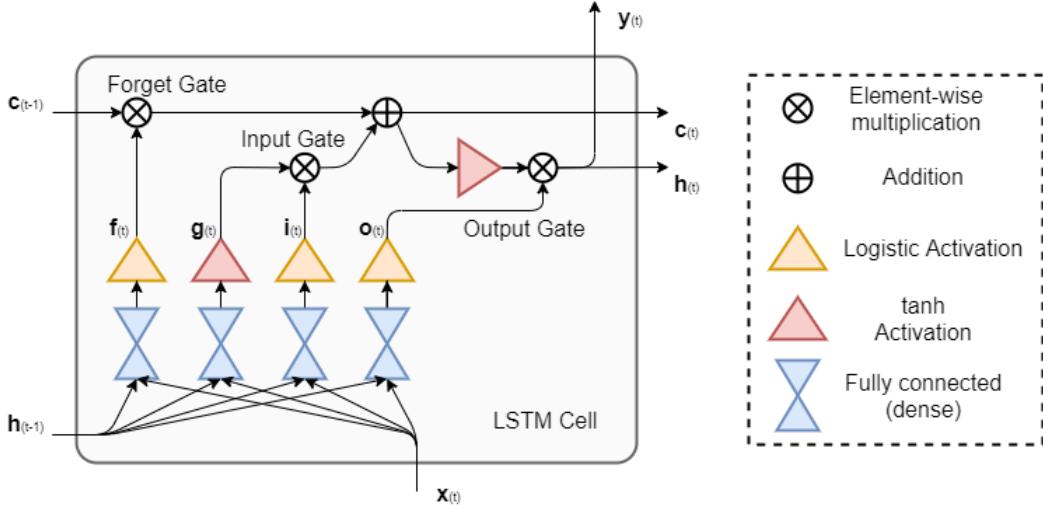


Figure 2.12 The schematics of a LSTM cell (adapted from [30])

As detailed explained in [30], we present the outline of the equations govern the interaction between inputs and outputs of the LSTM cell:

$$i_t = \varphi(\mathbf{w}_{xi}^T \mathbf{x}_t + \mathbf{w}_{hi}^T \mathbf{h}_{t-1} + b_i) \quad (2.41)$$

$$f_t = \varphi(\mathbf{w}_{xf}^T \mathbf{x}_t + \mathbf{w}_{hf}^T \mathbf{h}_{t-1} + b_f) \quad (2.42)$$

$$o_t = \varphi(\mathbf{w}_{xo}^T \mathbf{x}_t + \mathbf{w}_{ho}^T \mathbf{h}_{t-1} + b_o) \quad (2.43)$$

$$g_t = \tanh(\mathbf{w}_{xg}^T \mathbf{x}_t + \mathbf{w}_{hg}^T \mathbf{h}_{t-1} + b_g) \quad (2.44)$$

$$c_t = f_t \otimes c_{t-1} + i_t \otimes g_t \quad (2.45)$$

$$y_t = h_t = o_t \otimes \tanh(c_t) \quad (2.46)$$

2.4.1.2 Autoencoder Networks

Autoencoders are a subclass of self-supervised deep neural networks that are capable of learning condensed representations of data. Such representations are frequently referred to as latent representations or codings. Additionally, autoencoders are highly valuable for dimensionality reduction, since the acquired codings are often albeit less dimensions than that of the input data. In addition, autoencoder also act as feature detectors, and they can be used for unsupervised pretraining of deep neural networks [30]. Another advantageous property of autoencoders is that some of them are generative models, which means they can create new data at random that is highly comparable to the training data [30].

In short, an autoencoder learns to transfer their inputs to their outputs by adding various sets of constraints and other organizational processes. Subsequently, the latent reduction and noisy input preclude the network from quickly and easily mapping inputs to outputs, compelling it to learn effective data representation techniques [30].

As illustrated in Figure 2.13, a conventional autoencoder has almost the same composition as a network consisting entirely of fully connected layers (a multi-layer perceptron), only that the number of units in the output shall match the number of units in the inputs. As stated by the author of [30], the outputs are often referred to as reconstructions since the autoencoder attempts to reconstruct the inputs. At the same time, the cost function often includes a reconstruction loss that negatively affects the model whenever the reconstructions do not suit the inputs. Additionally, an autoencoder is considered to be undercomplete if its inner representation is less complex than the input data. As this emerges, the autoencoder is unable to simply replicate its inputs to the codings and therefore it ought to find an alternative to deliver a copy of its inputs, prompting it to learn the most meaningful features of the input data and disregard the less significant ones [30].

Two key methods for enhancing an autoencoder's capacity for learning meaningful features from the data and thus its efficiency in retrieving the source data without much noise are (a) applying gaussian noise to the input and adding a dropout layer immediately after the model obtained its inputs; and (b) constructing sparse autoencoders. The first approach, suggested by [36] and addressed in detail by [30], involves simply adding a dropout layer as the encoder's first layer, preceded by another layer that adds gaussian noise to the inputs. It is worth mentioning that these layers will work only throughout the training process. From the other side, the second approach entails augmenting the cost function with an extra parameter that regulates the number of active units in the coding layer [30]. Alternatively, a cheaper method is to use the sigmoid activation function in the coding layer, which confines the coding to ranging between 0 and 1, or the softmax activation function, which performs the same constraint except their sum is 1.

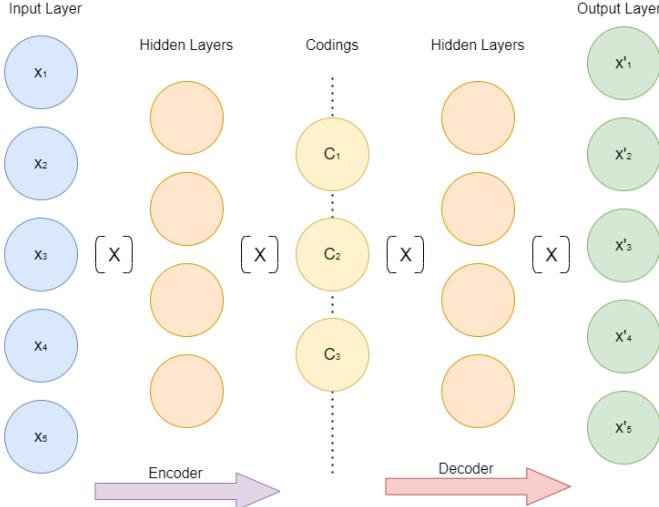


Figure 2.13 General architecture of an Autoencoder model

The variational autoencoder, originally suggested by [37], is a significant group of autoencoders that is adapted in this research. Their primary distinctions are they are probabilistic autoencoders, in which certain outputs are partly decided by chance even after training, and that they are generative autoencoders, which means that they may produce new instances that are quite representative of the original dataset [30].

In contrast to a conventional autoencoder, which produces codings for a series of inputs directly, variational autoencoders compute a mean coding μ and a standard deviation σ . The actual coding is then performed using a Gaussian distribution with a mean μ and a standard deviation σ , while the remainder of the decoding process remains unchanged [30]. A critical component of training a variational autoencoder is adding a latent loss to the model's overall loss, which forces the autoencoder to generate codings that seem to have been sampled from a standard gaussian distribution. The foundations of this latent loss are discussed in detail in [38]. The author of [30] addresses how a far more numerically robust and widely used variation is to make the encoder output $\gamma = \log(\sigma^2)$. The final form of the latent loss of a variational autoencoder is described as:

$$\mathcal{L} = -\frac{1}{2} \sum_{i=1}^n [1 + \gamma_i - \exp(\gamma_i) - \mu_i^2] \quad (2.47)$$

Chapter 3: Constitutive Model of Myocardium

Finite element method is used to solve nonlinear large deformation in the myocardium under pressure load. The constitutive equations described in 2.29 and 2.34 were implemented as a plugin material models [39] for the open source FE software, FEBio [40]. This section discusses the implementation of passive and active models, as well as the geometrical modelling along with the fiber implementation, mesh independence and boundary conditions.

3.1 Passive Model

The passive model of the left ventricular myocardium defined by the constitutive laws derived by Holzapfel et. Al. and noted on equation 2.29 were implemented as a material plugin within the FEBio software. The code was written in C++ and then compiled to a Dynamic Link Library (DLL) file that FEBio can import without any additional requirements.

To establish the material parameters and validate passive constitutive model, we used the methods presented in [6] and [7], in which we employed experimental data from six cyclic simple shear tests on cubic samples with an edge length of 1mm cut from the ventricular wall of explanted pig hearts to determine a , b , a_f , b_f , a_s , b_s , a_{fs} , and b_{fs} parameters. Moreover, the monotonous loading portion of the distinct cyclic shear stress-strain curves is utilized. It is worth noting that related methods were used in [41] and [42], as stated by the authors of [21]. Additionally, we contrasted the model's behavior under biaxial loading in three distinct conditions against representative stress-strain data obtained by [43] and described by the authors of [6].

3.1.1 Identification of Material Parameters

Taking into account that the authors of [7] assert their set of parameters accurately reflect their results, we used their material parameters, described in table 3.1, to facilitate our analysis, as the primary objective was not to derive our own collection of parameters for the passive model. Correspondingly, it is interesting to note that [6] and [44] used the same data set as of [7] to extract similar material parameters.

Table 3.1 Material parameters used within the passive constitutive model of the myocardium.

a (kPa)	b	a _f (kPa)	b _f	a _s (kPa)	b _s	a _{fs} (kPa)	b _{fs}
0.496	7.209	15.193	20.417	3.283	11.176	0.662	9.466

3.1.2 Validation

The validation of our implementation of Holzapfel and Ogden's passive constitutive model suggested in [6] consisted of evaluating FEBio simulation outcomes in two separate testing scenarios: simple shear and biaxial loading, with data provided in their paper.

Following the procedures explained in [6] we implemented the analytical solution summarized in 2.3.1 in MATLAB and performed six cyclic simple shear FE simulations with our constitutive model. Each test consisted on fixing one side of 1mm by 1mm cube while stretching the opposite side in fs, fn, sf, sn, nf and ns directions, respectively. The results were compared and are shown in figure 3.1. As it can be seen, stress-strain relationships computed through FEA agree well with their respective analytical solutions, enforcing the validation of our implementation.

Using a similar approach, we conducted FEA simulations for the biaxial test under different stretch ratios along the fiber (Eff) and sheet directions (Ess). The results were compared with experimental data showed in [6] and summarized in section 2.1 and are illustrated in figure 3.2. As it can be observed, our model presented similar behavior seen on experimental data for each stretching ration. The small divergence from results obtained by [6] and the experimental data is explained by the different set of parameters used during the simulations. In fact, the consisting stress-strain relationships obtained

through different material parameters enforces the validation and applicability of our model implementation.

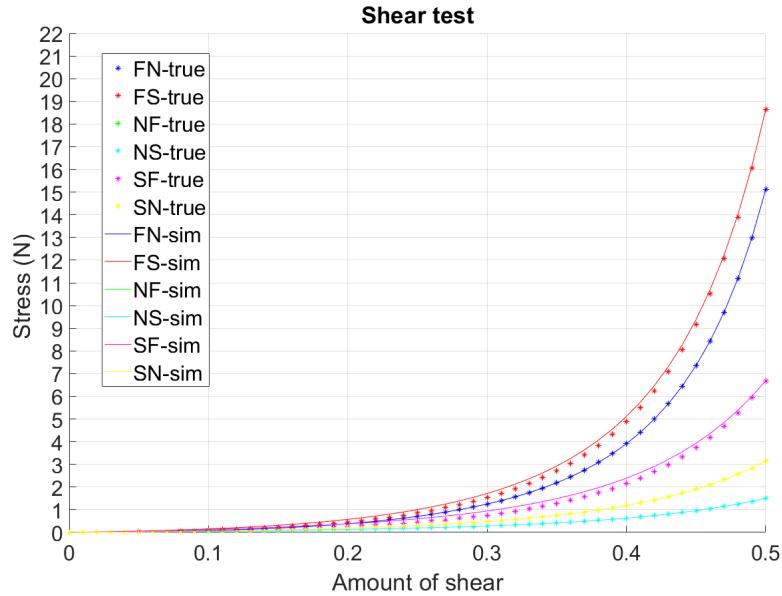


Figure 3.1 Comparison of analytical and simulated values for shear tests in different fiber orientations

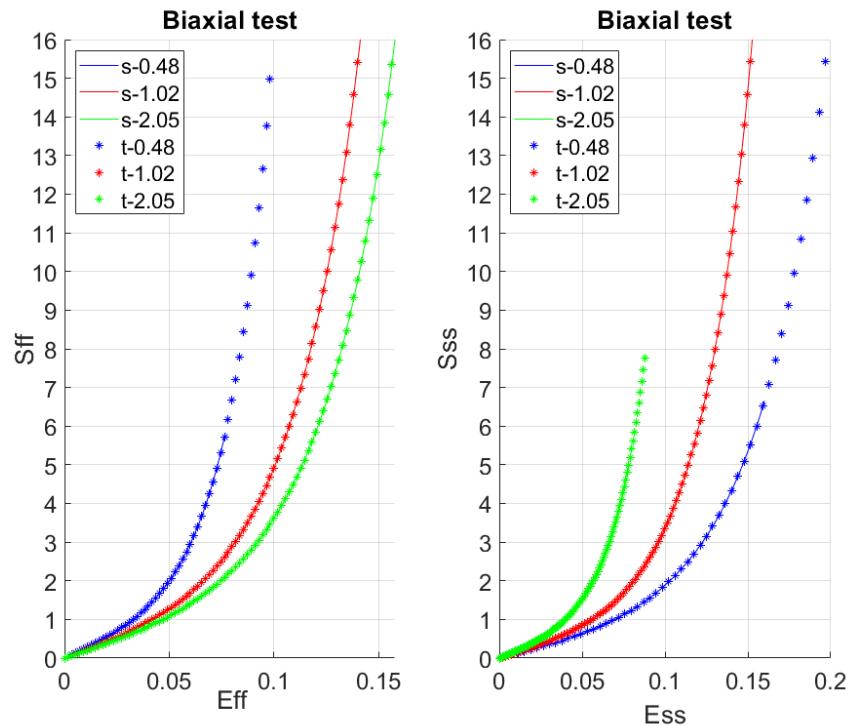


Figure 3.2 Comparison of analytical and simulated values of biaxial loading in Eff and Ess directions

3.2 Active Model

The active model of the left ventricular myocardium was implemented as a material plugin within the FEBio program using the constitutive laws deduced by Pezzuto et al. and shown in equation 2.34. The code was developed in C++ and then compiled to a Dynamic Link Library (DLL) file that FEBio can easily import.

3.2.1 Validation

To validate the active constitutive model implementation, we used a similar procedure to that used to verify the passive model: we extended the prior MATLAB script to incorporate the analytical solutions for the active formulation and conducted analogous shear tests for different values of γ . As with the last experiment, each test involved fixing one side of a 1mm by 1mm cube while extending another side in the fs, fn, sf, sn, nf, and ns directions, respectively. However, we varied the value of γ this time. Figure 3.3 illustrates the findings for two γ settings. As expected, when stretched along the fiber directions (fn and fs) with increased γ value, the material exhibits a stiffer contraction, closely matching the behavior of active contraction. Additionally, the findings suggest that the values acquired by FEA are highly aligned to their corresponding analytical solutions, indicating that the model's implementation is valid.

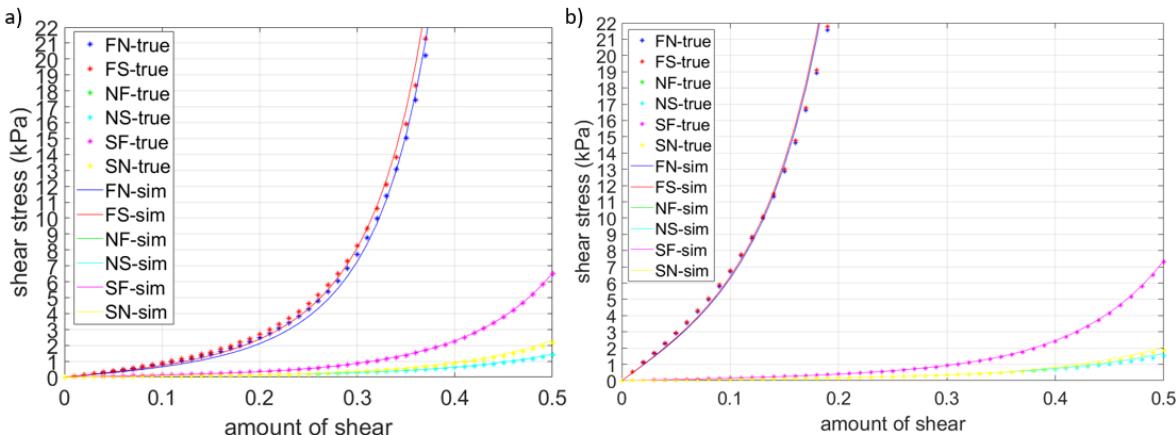


Figure 3.3 Comparison between the analytical (dots) and FE (lines) results of the active contraction model for shear tests along different combinations with γ value of 0.05 (a) and 0.11 (b).

3.3 Geometrical Model

Throughout the majority of the study, the geometry utilized was an idealized ellipsoid with a truncated base and a thick wall, which, as explained in [6], is an adequate and comprehensive representation of the left ventricular myocardium. Moreover, we modeled a normal-sized myocardium, which, according to the work of [45] has approximated 122ml volume. We based on the dimensions provided in [6] for a consistent wall thickness of 10mm. Additionally, as a reference we also based our idealized geometry on a related study done by [46]. The idealized geometry, along with its constructing dimensions, is illustrated in Figure 3.4.

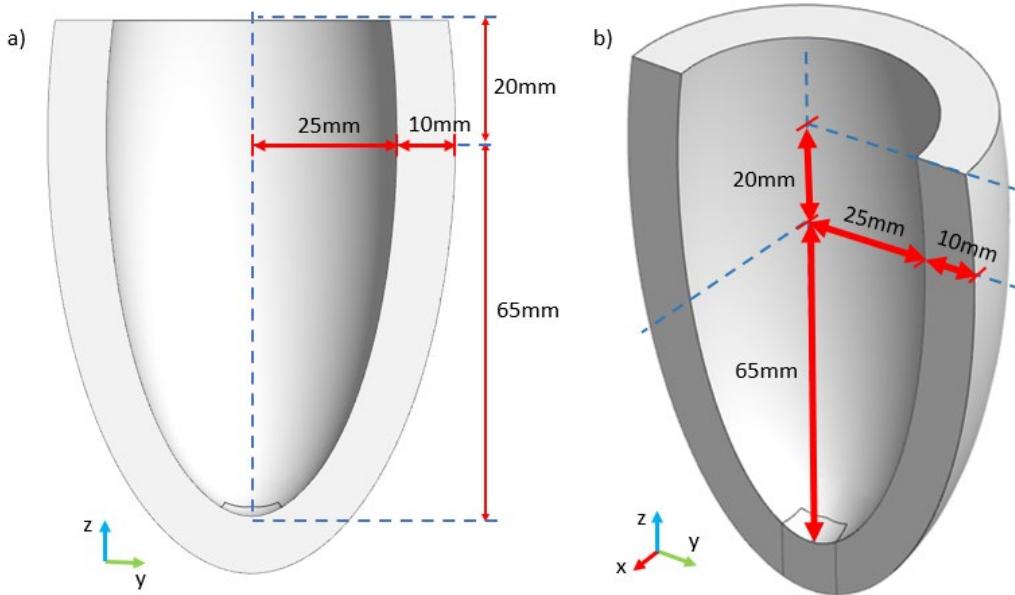


Figure 3.4 Modelling dimensions for idealized ellipsoid with a truncated base and a thick wall of the left ventricular myocardium

After the geometric dimensions were set, there were two main structural designs proposed to construct the model's mesh. The first is illustrated in Figure 3.5a and 3.5b and is composed of pure hexahedral elements of approximately constant size (with slightly exception on the apex, in which dimension vary with relation to the rest of the geometry). Moreover, as seen in figures 3.5b and 3.5d, the lower portion of the myocardium was designed with a wide hexahedral form, which causes all elements to

acquire square-like surfaces as they reach the apex, ensuring the entirety of hexahedral elements. The same is not true for the geometry suggested in this study, in which all components converge on a central point, caused by the accumulation of pentahedral elements surrounding the apex region, as seen in Figure 3.5c and 3.5d.

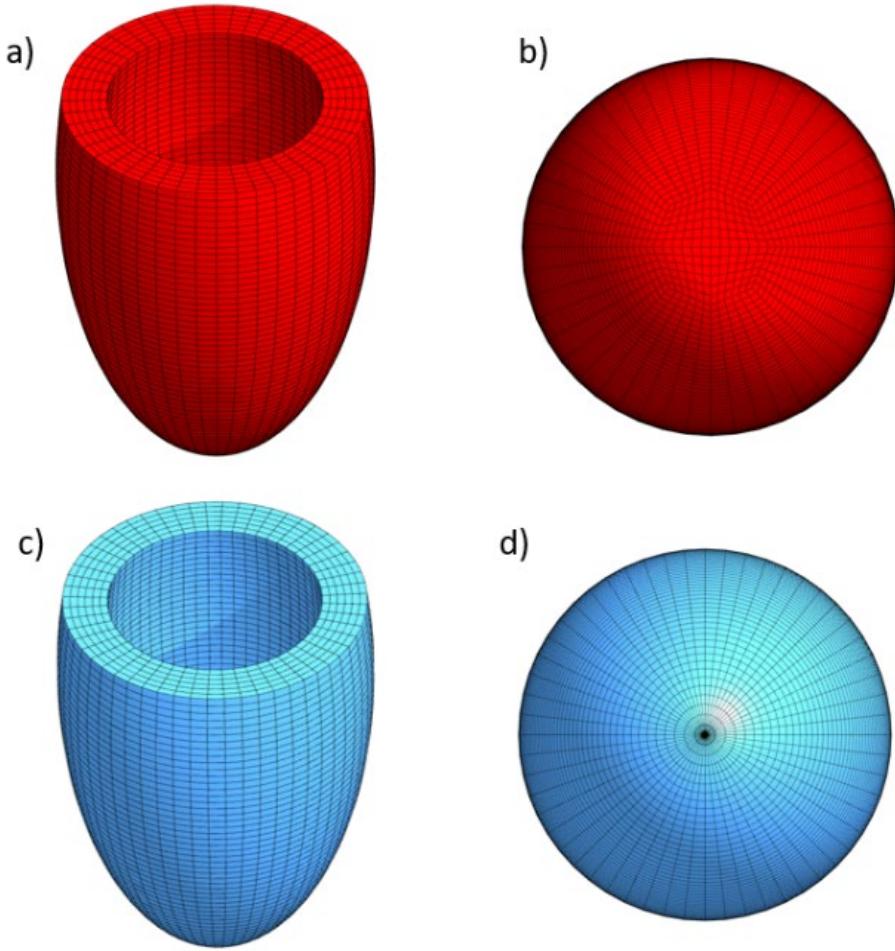


Figure 3.5 Mesh representation of the idealized myocardium geometry composed of only hexahedral elements (a and b) and with pentahedral elements around the apex (c and d).

Given the small differences between the two meshes, we first chose the second option as it offers a more uniform volumetric distribution throughout the model, thereby improving its mesh quality. This mesh structure was employed during the first trials and data collection. However, we detected large stress concentrations at the apex at later phases (during simulations with distinct PV loops and thus various

loading conditions), leading to simulation convergence errors. This problem was mitigated by changing to the all-hex mesh, which was subsequently utilized for the remainder of the study. With equal geometrical features and mesh independence, we lost no information upon changing meshes.

Moreover, Figure 3.6 highlights the features used in the final version of the idealized geometry. As it can be seen by the focus on Figure 3.6b, our mesh consisted of 6 layers across the thickness direction. Also, the apex consisted of pure hexahedral elements, as shown by figure 3.6c. Moreover, as later explained, the specified elements in the apex, are treated slightly differently from the other elements in order to improve model's convergence by limiting the significance

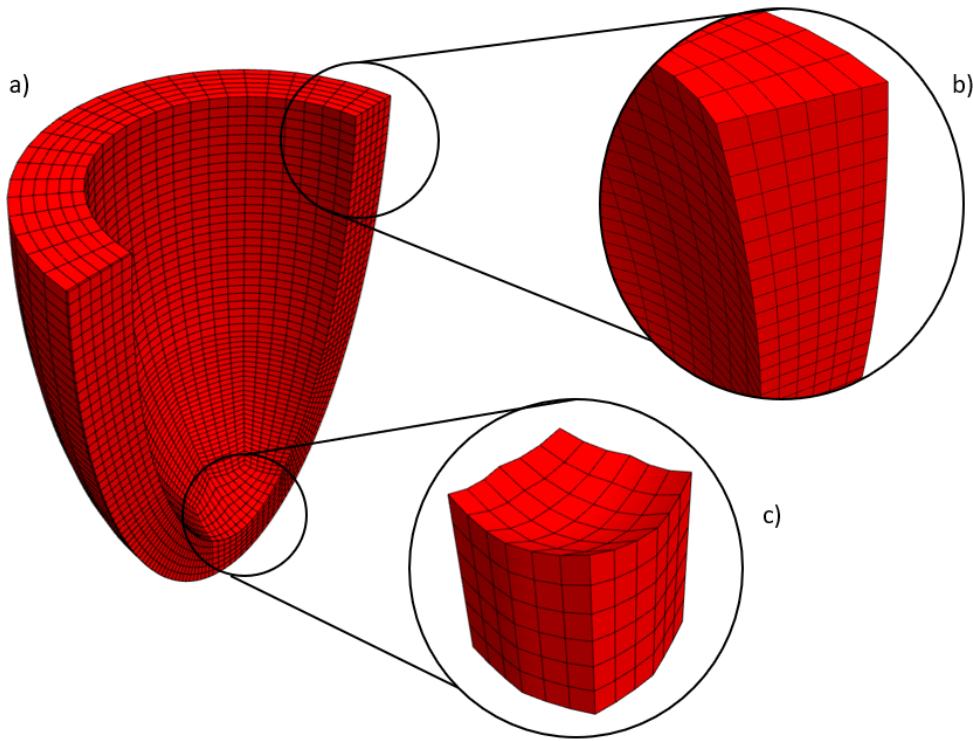


Figure 3.6 Features of the all-hexahedral mesh used in this study (a) with 6 cross-sectional layers (b) and pure hexahedral elements at the apex region (c).

Lastly, Figure 3.7 illustrates a patient-specific left ventricular myocardium. As it can be seen, the outlying geometry is similar to the idealized version. The main changes are related to the uneven thickness of the ventricular wall in both longitudinal and circumferential directions.

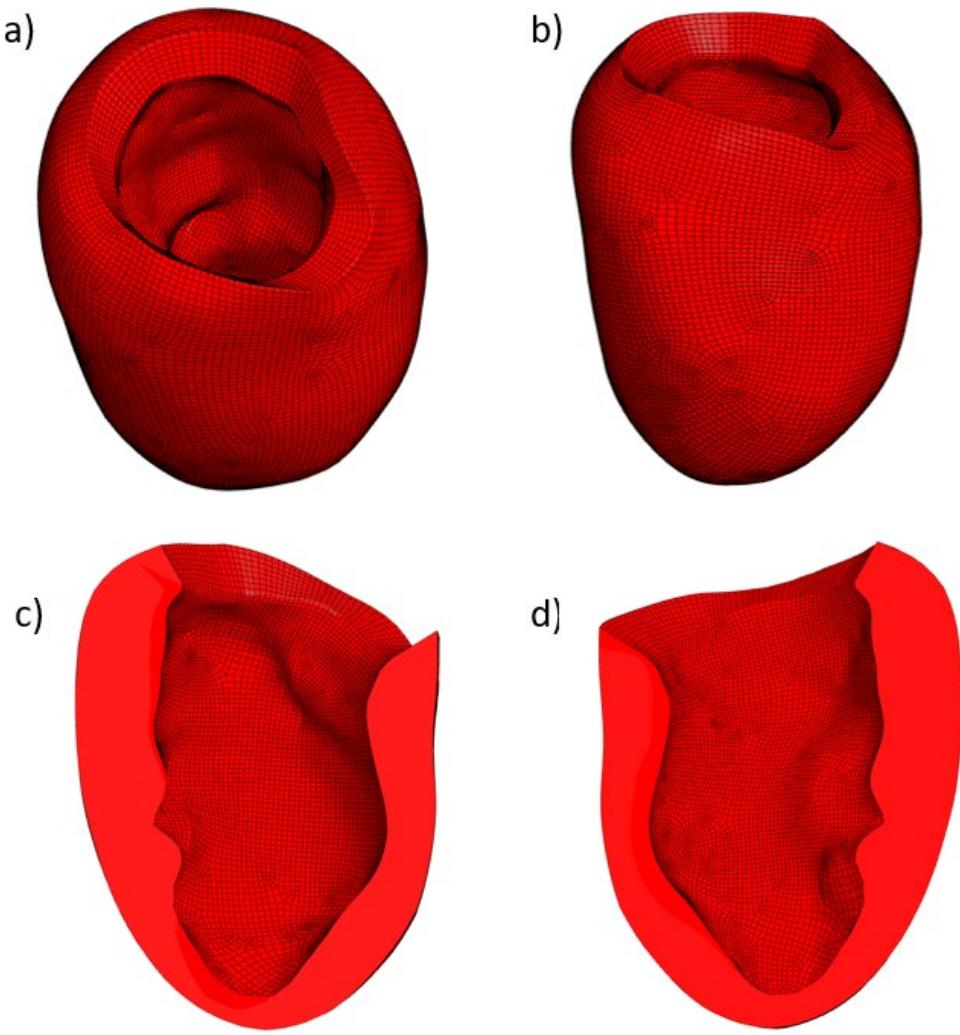


Figure 3.7 Meshing representation of a realistic model of the left ventricular myocardium

3.3.1 Fiber Field

Considering the fact that our constitutive model is strongly reliant on the orientation of defined fiber orientations, it is crucial to establish a state-of-the-art algorithm capable of constructing the fiber field with pre-defined boundaries (specified endocardial and epicardium orientations) while being flexible to varied geometries. To accomplish this, we used MATLAB to implement the work presented by [47] and [48]. It is a good approximation to clinical DT-MRI measurement and is often employed in the field

when clinical DT-MRI data are unavailable. In summary, the algorithm's central concept is to provide a continuous coordinate-free fiber interpolation by using computer graphics techniques driven by feature-based interpolation [47].

The algorithm is explained in details in [47], but, in principle, it has as its main focus to solve the given Poisson equation: $\operatorname{div}(\mathbf{K} \cdot \nabla \theta) = 0$ in \mathcal{B} , in which the Dirichlet boundary conditions are applied ($\theta = \bar{\theta}$ on $\partial\mathcal{B}_0$) and θ defines the angle of the fiber related to \mathbf{f}_0 , \mathbf{s}_0 and \mathbf{n}_0 . In other words, the algorithm starts by computing the main cross-sections that composes the given mesh; it then applies Poisson distribution of the fibers along the found cross-sections while interpolating their direction based on their longitudinal coordinates. Moreover, the code was implemented in MATLAB and tested on several files. The output was proven to be working correctly and two samples are shown below in Figure 3.8. The images 3.8a-b-c illustrates the fiber distribution for a epicardium and endocardium orientations of 65.697 and -69.612 degrees respectively, while Figure 3.9a-b-c computes for 59.810 and -71.693 degrees.

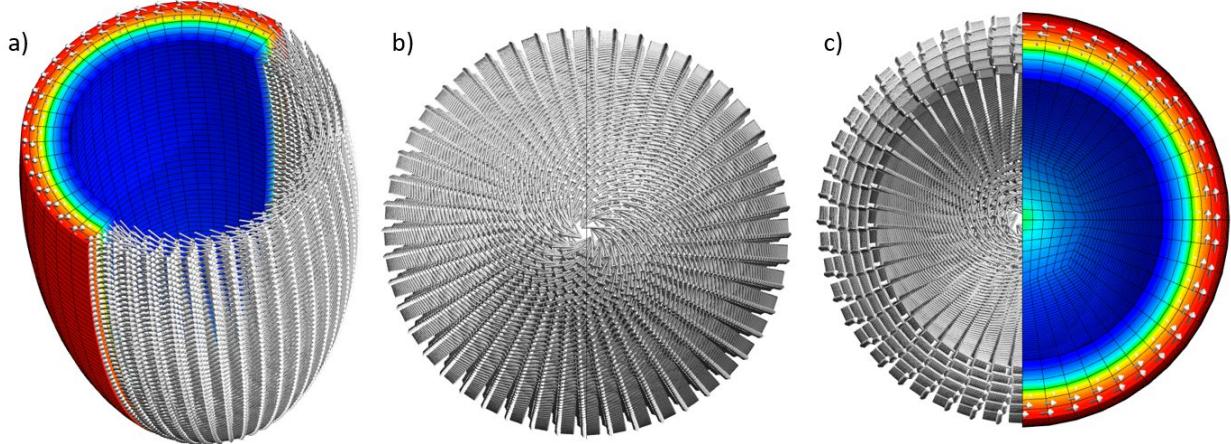


Figure 3.8 Fiber distribution of 65.697 and -69.612 degrees in the endocardium and epicardium directions respectively.

Moreover, our implementation of the algorithm for the computation of fiber orientations is proved to be mesh independent. As it can be seen on Figure 3.10, there is no significant difference when using the same set of fiber directions for the all-hex mesh (3.10a) and the hexagonal mesh with pentahedral

elements at the apex (3.10b). In addition, the color bar identifies the value of the fiber vector in the longitudinal direction. Similar results were obtained when comparing several fiber orientations on both meshes.

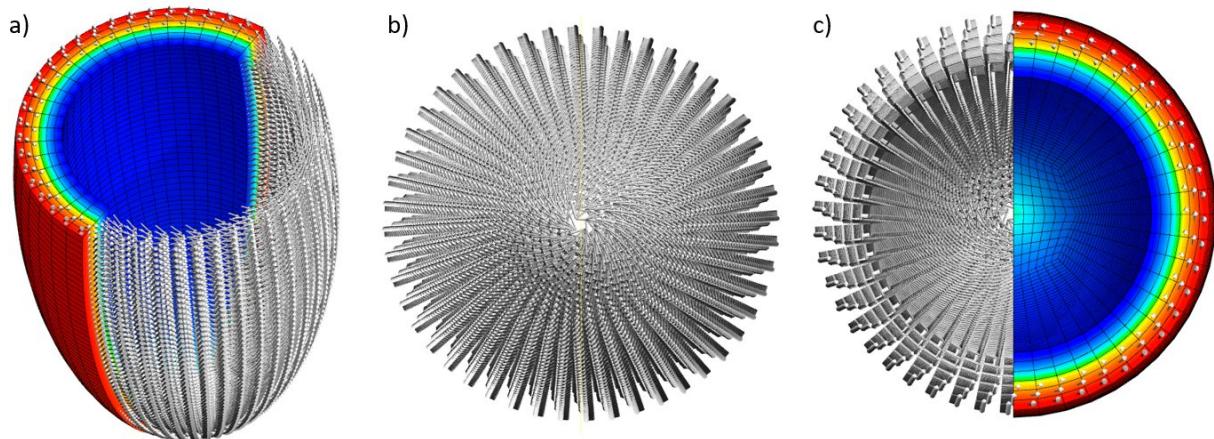


Figure 3.9 Fiber distribution of 59.810 and -71.693 degrees in the endocardium and epicardium directions respectively.

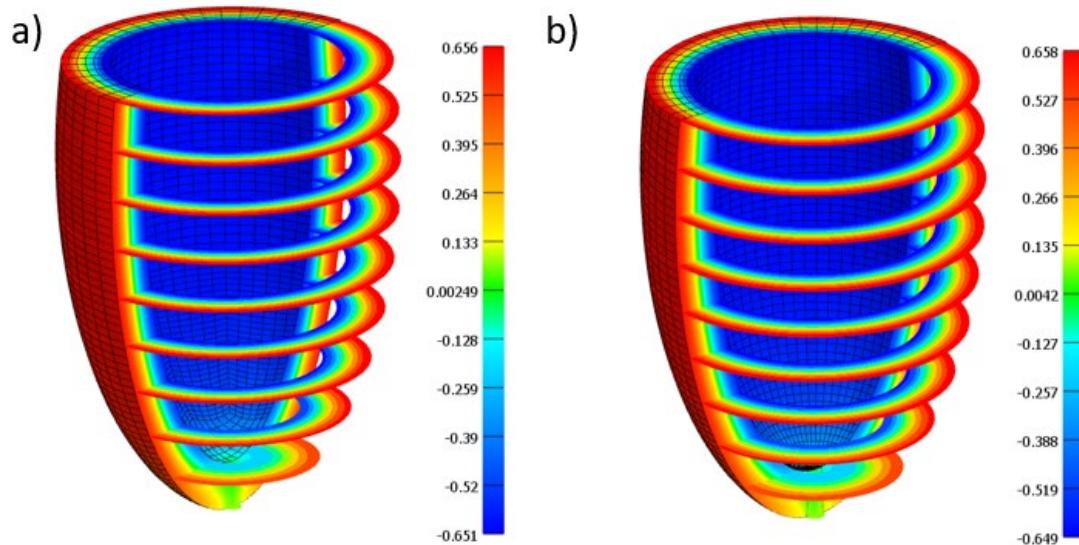


Figure 3.10 Comparison of the distribution of z component of \mathbf{f}_0 using all-hexahedral mesh (a) and with pentahedral elements at the apex (b) using fiber directions of 65.697 and -69.612 degrees in the endocardium and epicardium directions respectively.

3.3.2 Mesh Independence

To determine whether the constitutive model was able to encompass diverse types of meshes and, therefore, be mesh independent, a study was performed with three main types of mesh densities (coarse, medium, and fine) that determined the overall number of nodes. Moreover, considering the laminar fiber orientations, we also wanted to establish if the number of layers in the thickness direction would influence the final results. To do so, we composed different meshes with different number of layers while adjusting the number of elements in the circumferential direction to maintain uniform volume and within the three density categories. Sample meshes are shown in figure 3.11, where mesh density is increasing from left to right (along with the number of layers), and a sample tetrahedral mesh is shown on the right.

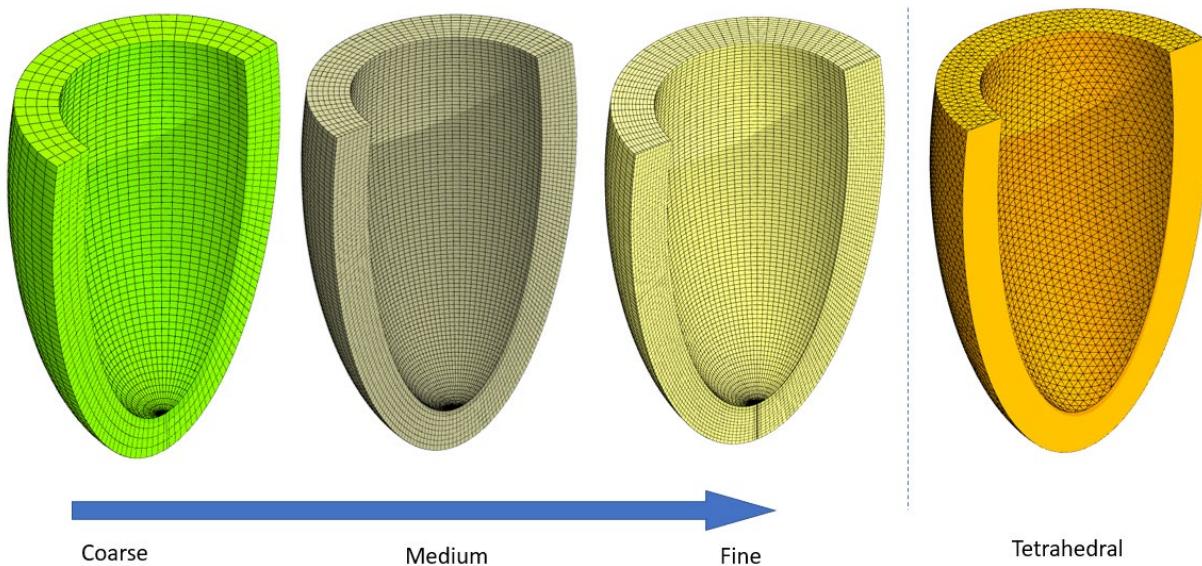


Figure 3.11 Representative differences in mesh densities used in the mesh study (left) and a sample tetrahedral mesh (right).

Each of these meshes was loaded with uniform pressure ranging from 0 to 16kPa at the endocardium surface. The results are shown in table 3.2 and are summarized in Figure 3.12 and 3.13. In addition, the mesh identifier (ID) is composed of three elements: mesh type (hexahedral or tetrahedral), mesh density and the number of layers.

Table 3.2 Results of mesh independence study.

ID	Maximum Stress (kPa)	Top 1% Maximum Stress (kPa)	Max Total Displacement (mm)
<i>Tet medium 5</i>	119.972447	113.502027	12.55714943
<i>Tet coarse 4</i>	119.0380652	112.5195214	12.27227307
<i>Tet fine 7</i>	110.4557194	103.0193334	12.93560499
<i>Hex coarse 4</i>	103.2305835	98.74748023	13.14202916
<i>Hex medium 5</i>	105.6144709	94.22194799	13.00021118
<i>Hex fine 6</i>	106.2496735	94.13301426	12.91502854
<i>Hex medium 7</i>	101.9968569	94.64374806	12.84988293
<i>Hex medium 8</i>	106.8863437	94.30277956	12.81919666
<i>Hex fine 8</i>	107.0516399	97.32435255	12.78999126
<i>Hex coarse 8</i>	106.4483216	93.80781408	12.79002257
<i>Hex fine 10</i>	107.23268	96.3676952	12.77512131
<i>Hex medium 10</i>	107.2020433	96.10077507	12.77235834
<i>Hex medium 12</i>	106.8860683	96.03210159	12.74132767
<i>Hex fine 12</i>	107.3755164	96.66774643	12.7480287
<i>Hex coarse 12</i>	107.2448423	96.1719763	12.7398294
<i>Hex coarse 14</i>	107.2353251	96.32200883	12.72128535
<i>Hex coarse 15</i>	107.262283	94.97135735	12.71458994
<i>Hex medium 16</i>	107.1684987	96.52101358	12.71001366
<i>Hex coarse 17</i>	107.2785817	95.56768661	12.70726881
<i>Hex coarse 18</i>	107.3952305	96.46411856	12.7049204
Mean	107.9612595	97.87042491	12.77030667
STD	4.179705904	5.415023129	0.167624299
Tet Mean	116.4887439	109.680294	12.5883425
Tet STD	4.283013388	4.727058707	0.271700898
HEX Mean	106.4564094	95.78633037	12.80241799
Hex STD	1.48946284	1.263901198	0.114694419

As table 3.2 demonstrates, there is minimal significant difference in terms of maximum displacement, while a standard deviation of 0.16mm for all meshes. However, the data also indicates slight deviation in terms of maximum stress and top 1% of maximum stresses, which is indicated by the difference in mesh type. Even though the deviation for tetrahedral elements was higher than for hexahedral elements, it is still very low and does not indicate mesh dependance.

Moreover, for illustration purposes, Figure 3.12 displays the data presented in the previous table.

As it can be seen, the variation within hexahedral meshes is minimal, while there is a slight deviation when using tetrahedral elements. The reasoning is applied to Figure 3.13, which visually shows the difference in maximum displacements within different meshes. The final data indicates that the number of layers is the most important characteristic of the mesh, as the number of elements in other directions do not significantly alter the results. Moreover, as the number of layers is increased, the FEA results get closer to the reference value (finest mesh). In addition, when the number of layers is larger than 8, mesh independence is reached (results are less than 5% from reference value). Lastly, it was shown that tetrahedral meshes in general provide relatively poor results. Besides the linear element test stated above, we also conducted the mesh independence study on quadratic elements. Similarly, we found that when the number of layers is equal or larger than 6, mesh independence is reached.

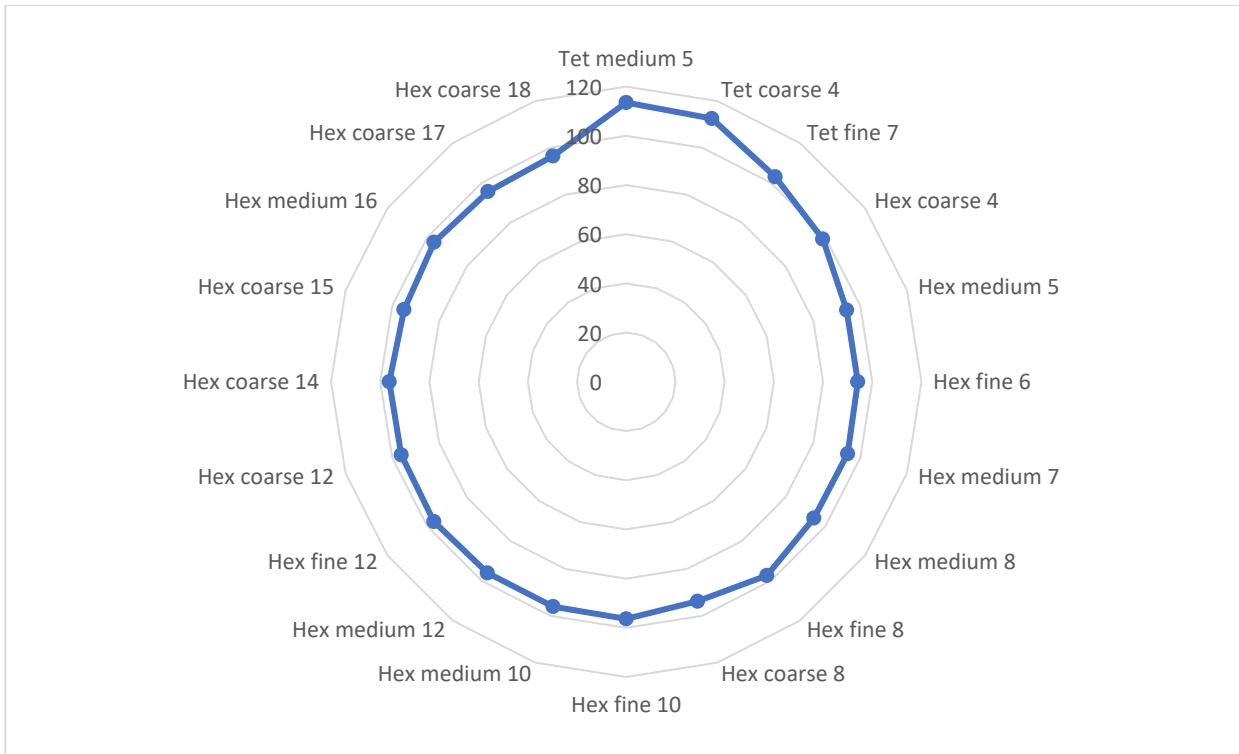


Figure 3.12 Average of stress values greater than 1% of maximum value for corresponding mesh configurations (kPa).

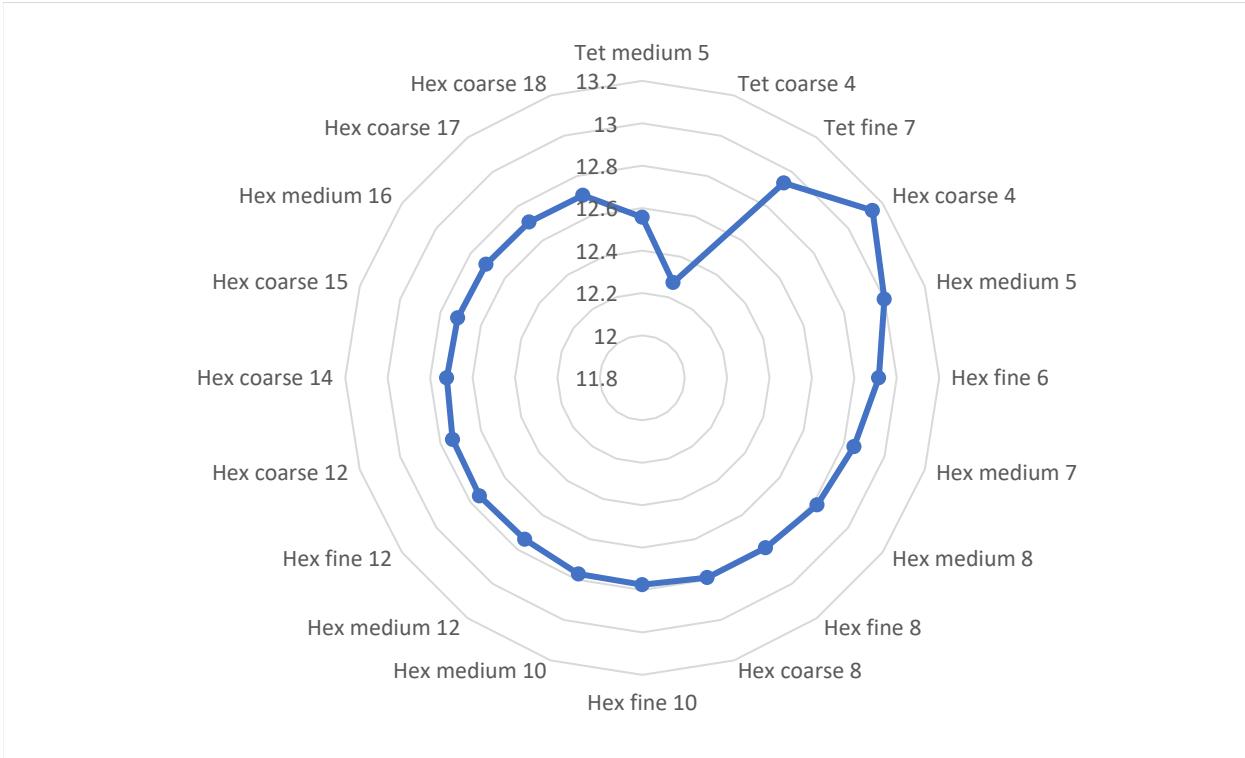


Figure 3.13 Maximum total displacement for corresponding mesh configurations (mm).

3.3.3 Boundary Conditions

It is critical to conduct FEA simulations using not only realistic loading scenarios, but also plausible boundary conditions. As a result, we naturally applied boundary conditions along the face that encompasses the myocardium's base region. In Figure 3.14, the boundary condition is illustrated in yellow in the idealized model, which, as can be seen, is formed of the horizontal plane at the highest part of the geometry. Additionally, to accommodate fiber rotation throughout the model, we applied constraints only in the longitudinal direction of the selected plane, which is aligned with the z axis in this case. We found that this boundary condition ensures the realistic deformation of myocardium under pressure loads and also prevents the undesired rigid body motion. Meanwhile, spatially uniform pressure load is applied on the endocardial surface in a cardiac cycle.

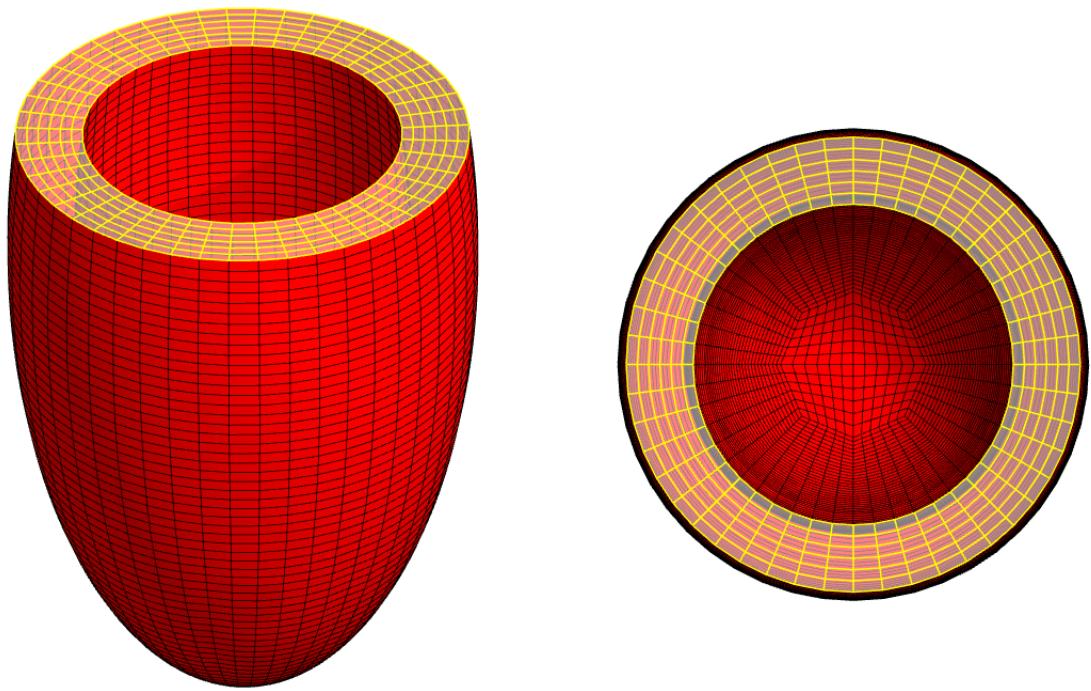


Figure 3.14 Boundary conditions used in all simulations performed with idealized myocardium.

Chapter 4: Constitutive-based Deep Learning Model

4.1 Data Acquisition and Manipulation

The "unreasonable effectiveness" of data for machine-learning systems has been a source of debate for years [49] [50]. Additionally, it has been proposed that several significant advancements in the area of Artificial Intelligence have been limited by the scarcity of high-quality datasets, rather than algorithmic developments [51]. The underlying principle going through these debates is the critical role of data in achieving state-of-the-art machine learning models.

To support this line of thought, it is empirical that the data extracted to train our deep learning model not only satisfy the broad domains underlying the contraction of the left ventricular myocardium, but also consist of clean and relevant information. In this section, we summarize the two main data extraction and cleaning process used for training, validating, and testing of our deep learning models.

It is noteworthy that our datasets were later incorporated into TensorFlow Dataset Generator [52] to improve efficiency when processing vast amounts of data and when training, validating, and testing deep learning models.

4.1.1 General Myocardium Features Under Specified Pressure-Volume Values

The primary dataset for this study is based on the relationship between pressure-volume values and clinically relevant parameters such as ejection fraction, apex torsion, longitudinal shortening, radial shortening, and wall thickness. Furthermore, we expanded our dataset by incrementally raising the pressure on the endocardial wall- and recoding adapted clinical parameters for each instance, rather than using singular values from ESV and EDV. For instance, the volumetric fraction was used as a proxy for the ejection fraction, in which the stepwise myocardium volume was compared to its initial state. Similarly, all other parameters listed in section 4.1.1 were determined by comparing their step value to

their corresponding initial state. Hence, we obtained a precise relationship between the myocardium's internal pressure and cavity volume along specified local features.

As shown in table 4.1, we conducted multiple FEA simulations of linearly distributed gamma values (in intervals of 0.01) along with cross-distributed values (ranging from 40 to 80 degrees in steps of 10 degrees) for fiber directions on both the endocardium and epicardium; for each simulation we applied linearly incremental pressure load from 0kPa to 16kPa. The cavity volume and clinical metrics were recorded after each simulation. After collecting the data, outliers were removed using the maximum standard deviation and IQR (Interquartile range) techniques, as well as other standard data cleaning techniques.

Table 4.1 Summary of dataset containing clinical metrics of the myocardium when exposed to linear loading ranging from 0 to 16kPa, gamma varying from 0 to 0.25, and fiber orientations ranging from 40 to 80 degrees in the epicardium direction and from -40 to -80 degrees in the endocardium direction.

	Volume	Pressure	Volumetric Fraction	Apex Torsion	Longitudinal Shortening	Radial shortening	Wall thickness	Gamma	Epi	Endo
mean	120779.41	6.86	0.02	0.75	-0.03	0.01	-0.03	0.12	62.42	-59.48
std	42847.72	5.12	0.35	0.41	0.08	0.13	0.12	0.07	13.02	12.58
min	52283.13	0.00	-0.86	0.00	-0.25	-0.32	-0.25	0.00	40.00	-80.00
25%	83963.53	1.92	-0.25	0.40	-0.10	-0.09	-0.12	0.06	50.00	-70.00
50%	115185.14	6.40	0.06	0.74	-0.03	0.01	-0.03	0.12	70.00	-60.00
75%	153383.63	11.2	0.32	1.07	0.03	0.11	0.05	0.18	70.00	-50.00
max	228570.10	16.0	0.57	1.90	0.14	0.31	0.36	0.25	80.00	-40.00

Although we collected approximately 17,592 relevant data points, we considered that this was insufficient to encompass the entirety of our desired domain. To address this problem, we enhanced our dataset by conducting two-dimensional interpolation along primary axes, gamma and pressure, for each combination of fiber orientations. This culminated in a dataset of 250,000 data points, about 14.21 times the size of our initial dataset. As an example, the volumetric fraction data augmentation procedure is shown in Figure 4.1, with the coarse cleaned data in (a) and the dense augmented data in (b). As can be shown, pertinent data is retained during the data augmentation process.

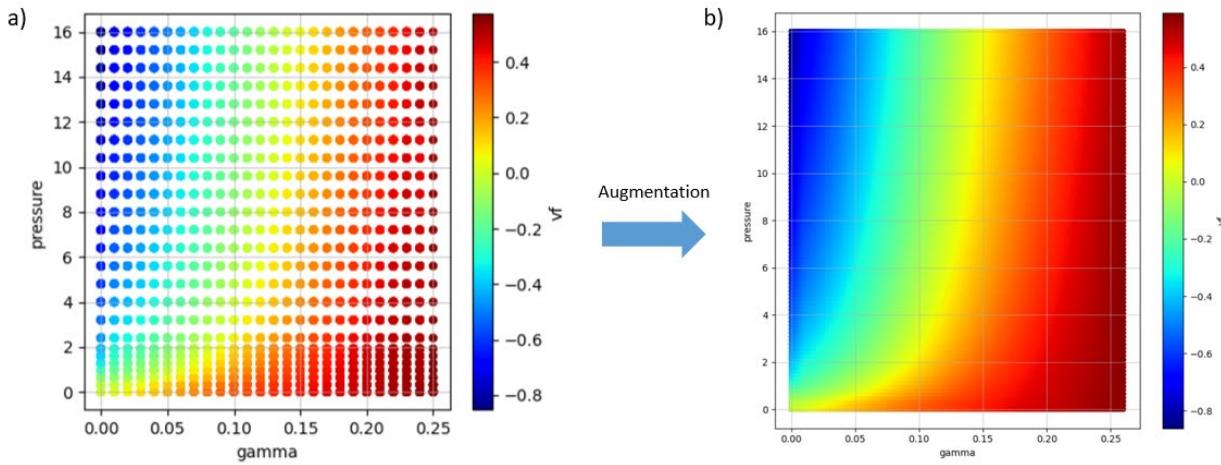


Figure 4.1 Sample data augmentation of volumetric fraction: Raw data is shown on left (a) and augmented data is shown on right (b)

Moreover, Figure 4.2 exemplifies the distribution of our augmented dataset along the endocardium (endo) and epicardium (epi) dimensions by illustrating the values of volumetric fraction. Seeing that all data values obey a consistent linear relationship along horizontal planes parallel to the fiber orientation domains, we concluded that additional data augmentation in these domains is unnecessary.

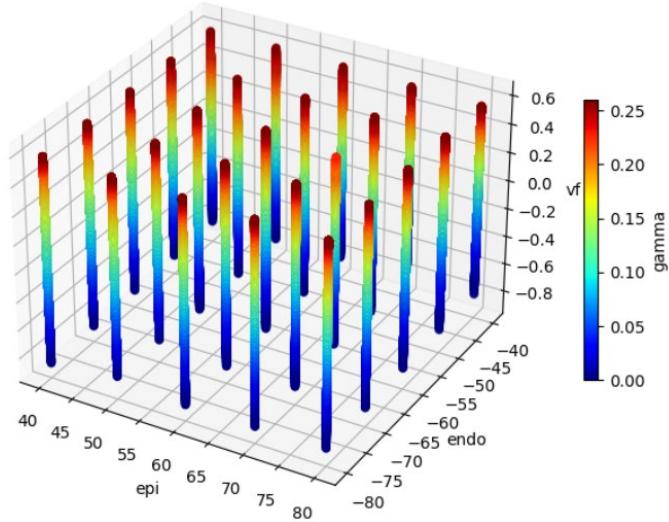


Figure 4.2 Sample distribution of gamma values along different ranges of endocardium and epicardium fiber directions.

4.1.2 Data Acquisition for PV Loop

The second dataset was used to train, validate, and test a variational autoencoder, which generates reliable PV loops based on cardiovascular clinical parameters, such as EDP, ESP and ESV, and other related parameters, such as minimum volume, minimum pressure and maximum pressure (often denoted as mean arterial pressure). To compose this dataset, PV loops were extracted from representative images utilizing a typical graphing extracting technique with point-wise feature extraction.

Additionally, to enhance the model's capability and to ensure consistency with the previous dataset, the data was normalized in both dimensions utilizing the figure's maximal reference point. This resulted in the PV loops having pressure and volume measurements ranging from zero to one, enabling them to be adjusted to the desired scale. This process is illustrated in Figure 4.3, where a sample image (a) is used to extract normalized datapoints (b). Due to the proximity or intersection of lines, not all curves could be identified with high precision. The augmentation of these datapoints is portrayed in figure 4.3c.

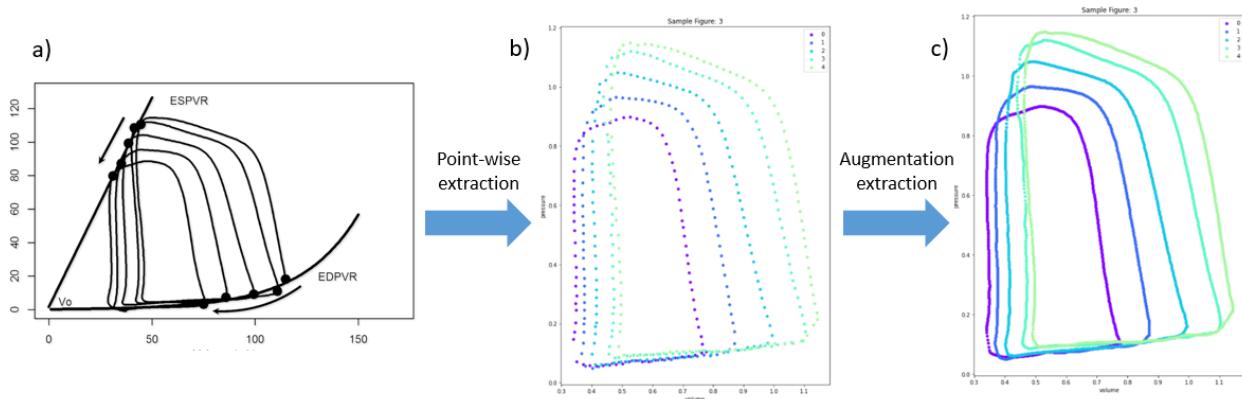


Figure 4.3 Sample PV loop data source (a) used for point-wise data extraction (b) and later augmented (c)

While this method extracted several PV loops from a variety of sources, there was only a limited amount of data available for training an effective and universal deep learning model. To address this problem, we developed a random algorithm that scaled and translated a given PV curve in both dimensions using random parameters contained within a given boundary. Additionally, the algorithm was capable of performing random mutations in either the pressure or volume dimensions of other PV curves,

resulting in unique randomized curves that adhered to the principles of a valid PV loop. The Figure 4.4 demonstrates the generation of new PV loops using random seeds; (a) shows random curves being created using default settings (without defined constraints), (b) indicates the highest volume permitted is 1.0, and (c) shows the initial and maximum volumes are all set to 1.0.

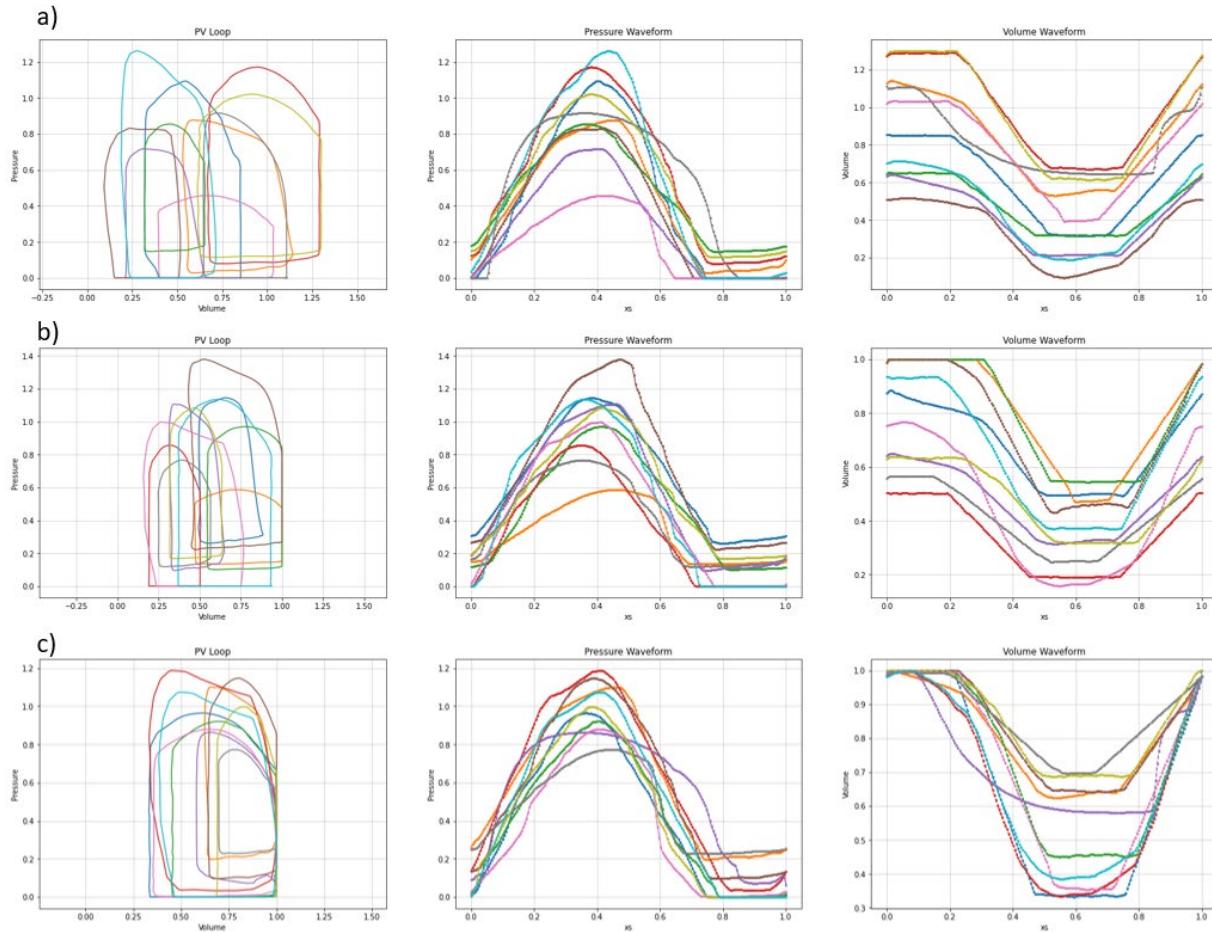


Figure 4.4 Random generated PV loops based on the proposed generation algorithm with (a) default settings, (b) volume bounded to (None, 1.0) and (c) initial and maximum volumes are set to 1.0.

4.2 PVLOOPED

PV loops were generated purely on the basis of cardiovascular data (CDs) through the development of a deep learning variational autoencoder, hence the name "PVLOOPED." As described in Section 2, autoencoders are capable of independently studying dense embodiments of the input data and are immensely beneficial for dimensionality reduction. More precisely, variational autoencoders are

probabilistic and generative models that can generate added content at random that is systematically related to the training data. However, rather than enabling the model to establish codings purely based on the PV loop, we compelled it to do so using selected normalized cardiovascular parameters that reflect the curvature of the PV loop.

The model was implemented using TensorFlow 2.0 and the dataset was based on the PV generator described in section 4.1.2, along with custom classes for analyzing and extracting useful features from the PV loop. Additionally, Figure 4.5 demonstrates a sample PV loop with representative CD details (The figure also illustrated the reference PVL point, which is determined at the lowest volume of the PV loop and it is used in other DL models). As seen, our custom algorithm is capable of autonomously detecting the major phases of a normal cardiac cycle, including isovolumetric contraction, ventricular injection, isovolumetric relaxation, and ventricles filling (which are distinguished by different colors in Figure 4.5). The minimum volume, maximum and minimum pressures, end-diastole pressure (EDP), end-systole pressure (ESP), and end-systole volume (ESV) were chosen as cardiovascular parameters for the PV curve generation. Due to the fact that the curves were normalized in both pressure and volume dimensions and the maximum volume was set to 1.0 to allow for scaling to any desired patient-specific results, the maximum volume and end-diastole volume (which is often similar to 1.0 in this case) offered no additional context to the model and were thus omitted as inputs.

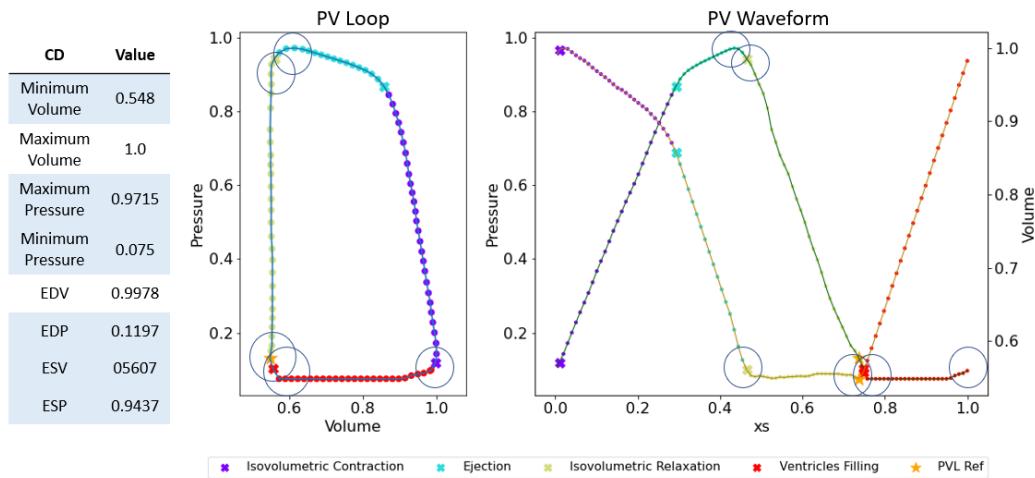


Figure 4.5 Sample PVLOOPED input with highlighted features.

4.2.1 Model Architecture

To empower the model to learn codings that resemble clinical parameters while also establishing reconstruction parameters for a complete PV loop, the autoencoder was implemented using TensorFlow's functional models in such a fashion that, during training, it has two major sub-encoders, one that receives both the PV loop and its CDs and another that receives only the second content. These encoders share the final two layers (the model's final dense layer and the output layer), which ensures that identical codings are produced. The key relations formed during the training phase were compared to the final PVLOOEP model in Figure 4.6. As shown by the orange color, the encoder is composed of two sub-encoders with shared layers (dotted connections).

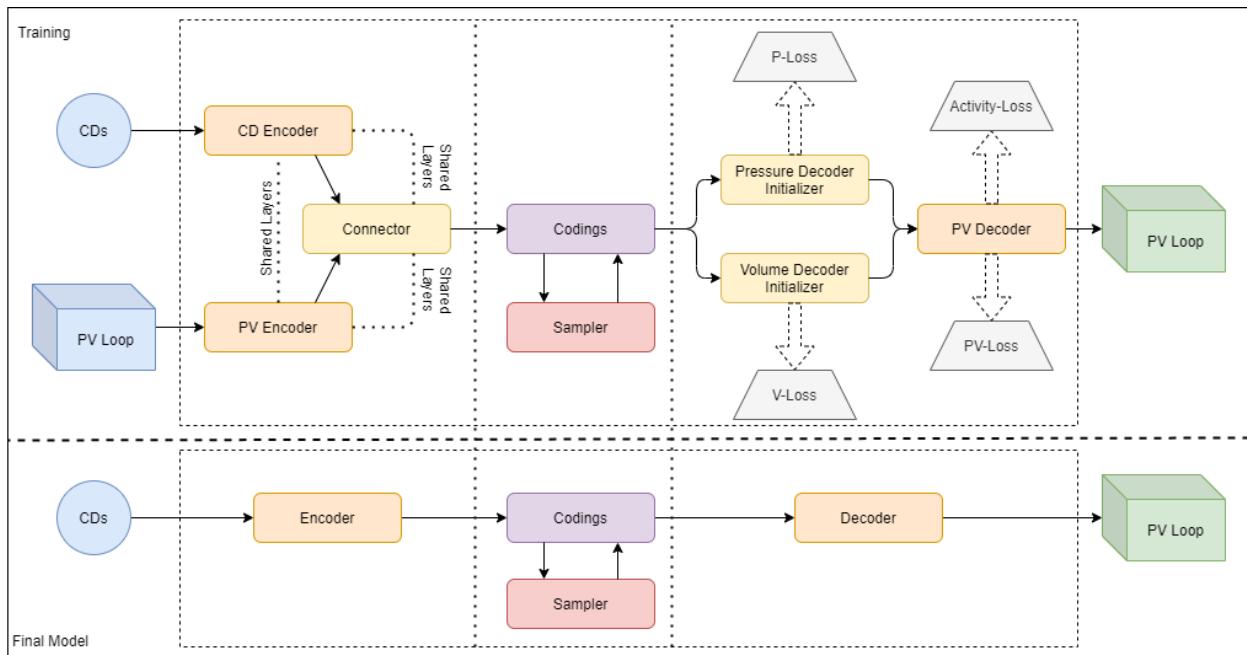


Figure 4.6 Schematics of (a) training and (b) final model for the PVLOOEP autoencoder model.

Even though these codings were generated using the same output layer, the model needs to learn relationships between their values in order to converge to the same solution provided a multitude of different inputs. To alleviate this challenge, a "connector" sub-model consisting entirely of fully connected layers was developed to accept the outputs of both sub-encoders. However, by establishing such a relationship, its outputs would differ from those of the CD encoder, and therefore the decoder

would be unable to function solely with CD values. To address this problem, the final layer of the CD encoder, and subsequently of the PV encoder, was shared with the connector's output layer. This mechanism is shown in Figure 4.6, where the PV encoder and CD encoder serve as the connector's inputs, with their output layer shared with the latter (illustrated by the dotted lines).

The final stage in the encoding scheme is to add the autoencoder's variational property, which enables it to produce new PV curves that seem to have been derived directly from the original dataset. This was achieved by setting up a "Sampler" sub-model that recognizes encodings provided by the connector and generates two additional encodings, a mean encoding μ and a standard deviation encoding σ , which are then concatenated with the original set of encodings before progressing to the decoder. Inside this sub-model, fully connected layers extract the relevant details from pure codings to produce additional codings that are used to generate PV loops that appear to have been randomly chosen from a gaussian distribution of the original dataset. Additionally, since this sampler was conditioned utilizing connector layer outputs, it can be used separately in the final model, as both the connector and the CD encoder supply identical codings. In figure 4.6, the sampler model is depicted by the red rectangle linked to the codings.

Furthermore, the decoder is responsible for reconstructing a PV loop provided a series of codings. Initially, this model was intended to be a single module with one-dimensional convolutional layers that addressed the specified codings; however, after initial trials, the model was unable to converge to a successful solution. This could have occurred as a result of the latent space's dimensionality being expressed exclusively by 1D data of singular values, while the reconstruction needed longer and systematic 2D data (input shape was [1, 8] and latent shape [1, 21], while the reconstructed PV shape was [100, 2]). Considering this, two additional sub-decoders were developed to reformulate the output, with one focusing on recreating the pressure waveform and the other on recreating the volume waveform of the PV loop. The performance of these decoders was filtered via the initial 1D convolutional decoder to achieve substantial relationships between the two waveforms and construct a single PV loop. This

mechanism is depicted in Figure 4.6, where each sub-model was held accountable for its contribution to the overall model loss depending on its role. The TensorFlow implementation is shown in Figure 4.7. As it can be seen, the model accepts 6 inputs (minimum volume, maximum and minimum pressures, EDP, ESP, and ESV), computes 21 codings and outputs a PV loop with 100 datapoints with size of 2 (one for pressure and another for volume).

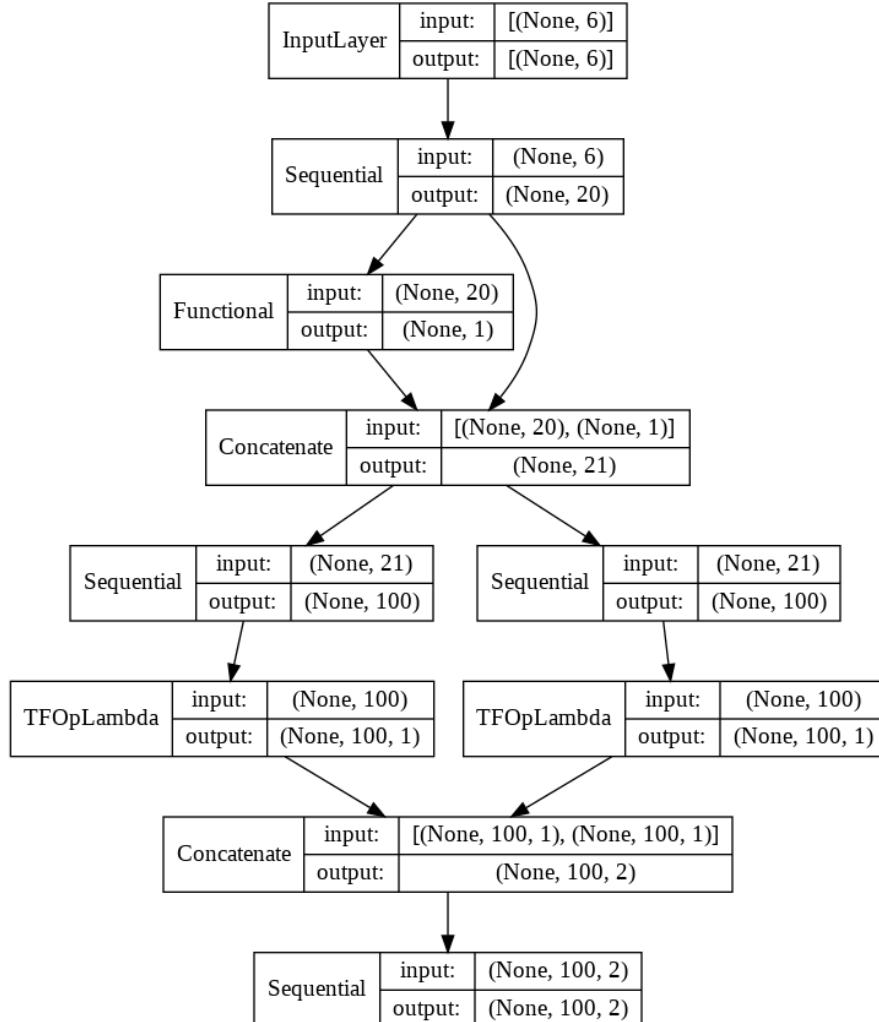


Figure 4.7 PVLOOPEd implemented model structure with respective layer connections information.

4.2.2 Training and Validating

The PVLOOPEd model was trained using randomly generated PV loops with dataset sizes of 5000, 2000 and 2000 curves for training, validating, and testing (roughly 55%, 22% and 22% data

distribution). All values were generated using the TensorFlow dataset generator described in the previous section. Additionally, to enforce the constraint imposed by minimizing the errors of pressure and volume separately via sub-models while, at the same time, establishing strong bonds between them, the model was wrapped within an outer model that provided auxiliary outputs for the results of the two sub-models. This process can be visualized in Figure 4.6, where the P-Loss and V-Loss indicates the two auxiliary outputs for loss computation. Additionally, the weights for each loss were altered to 0.25 for auxiliary outputs and 0.5 for major outputs to favor the main outcome.

Moreover, the model was compiled with Adam optimizer using the AMSGrad variant (already implemented in TensorFlow module) and the parameter for beta_1 was adjusted to 0.92 based on early trials. “Early stopping” with patience of 10 and “Reduce on Plateau” with factor of 0.8 and patience of 4 was used to prevent overfitting while stimulating learning if no significant progress is shown during training (both callbacks are provided by TensorFlow). The model was then set to train for over 500 epochs (number of training iterations) on batch sizes of 100. The training process is shown in Figure 4.8.

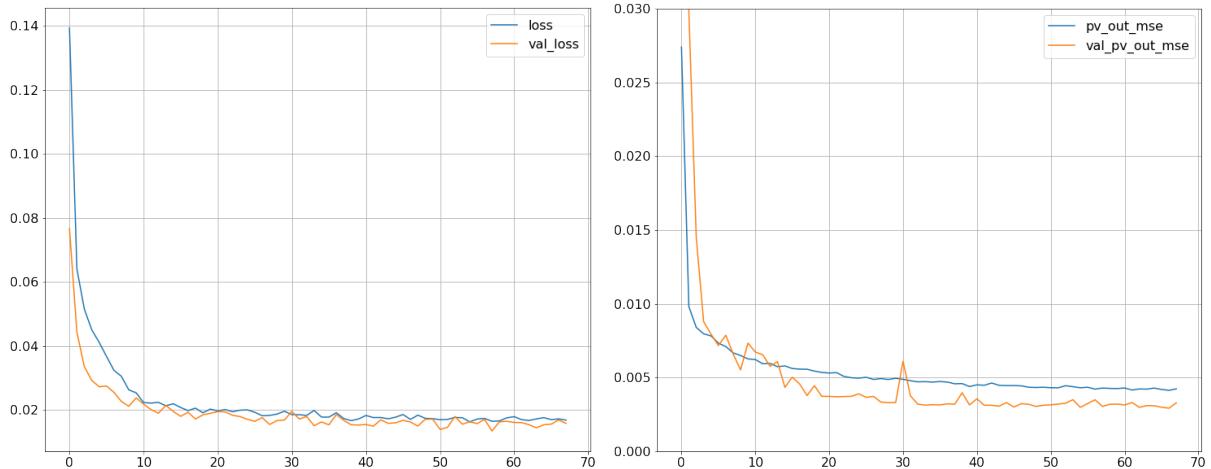


Figure 4.8 Training and validating (left) loss and (right) MSE of the PVLOOPED model.

The Figure 4.8a shows the model loss while training (blue) and validation (yellow). As it can be seen, the model was close to overfitting during training while not showing significant improvements after roughly epoch 55, triggering early stopping and ending the training a few epochs later. Moreover, the

constant variations presented in the validation loss indicates the difficulty of converging. Additionally, the plot on the right shows the Root Mean Squared Error (RMSE) values for the PV loops predictions evaluated during training (blue) and validation (yellow). Although the validation RSME was higher than the one for training during early stages (indicating overfitting) the two lines diverged after the model acquired enough information to produce better predictions while the effects of Dropout prevented overfitting to occur. The final MSE and stateful RMSE values for evaluating the model over testing dataset are shown in table 4.2, below:

Table 4.2 Evaluation of PVLOOPED model using testing dataset.

Output	MSE	RMSE
Pressure (auxiliary output)	0.0025	0.0502
Volume (auxiliary output)	0.0010	0.0319
PV Loop (final output)	0.0029	0.0537

As it can be observed, the model was able to reproduce volumetric waveforms with very low MSE, indicating high accuracy for these values. However, the model diverged when producing pressure waveforms, which indicates the increased MSE for the overall model is mainly due to pressure values. Although ideally the model should be capable of representing both data dimensions with same accuracy, it is understandable the higher deviance in pressure values since they were not as strictly bounded as values for volume. As previously described, values of volume were set to have an initial value of 1.0, which could be used to scale to any patient-specific myocardium, resulting in higher generation of values close to this region due to the shape of the PV loop (isovolumetric contraction). However, the only boundary for pressure values is that the minimum allowed value is 0.0, and it is not fixed at the initial step, resulting in randomized starting configurations.

To quickly investigate the issue, the model was trained with other optimizers: RMSprop and SGD. Different settings were used in diverse trials, however, the model did not present good convergence even if let it train for the full 500 epochs (without early stopping). The original version was then tested under 3-fold cross validation that resulted in similar values of those listed in table 4.2.

4.2.3 Testing

To verify our trained model's ability to generate PV loops, we utilized the testing dataset, which consisted of the necessary inputs for a specific PV loop and the PV loop itself. Figure 4.9 illustrates four sample comparisons between true curves taken from the dataset and the PV loop generated by our model. As it can be seen, the autoencoder's variational property enables the model to generate curves that are remarkably similar to those in the original dataset. Moreover, Figure 4.10 indicates the model is able to generate suitable PV loops solely based on a set of cardiovascular parameters.

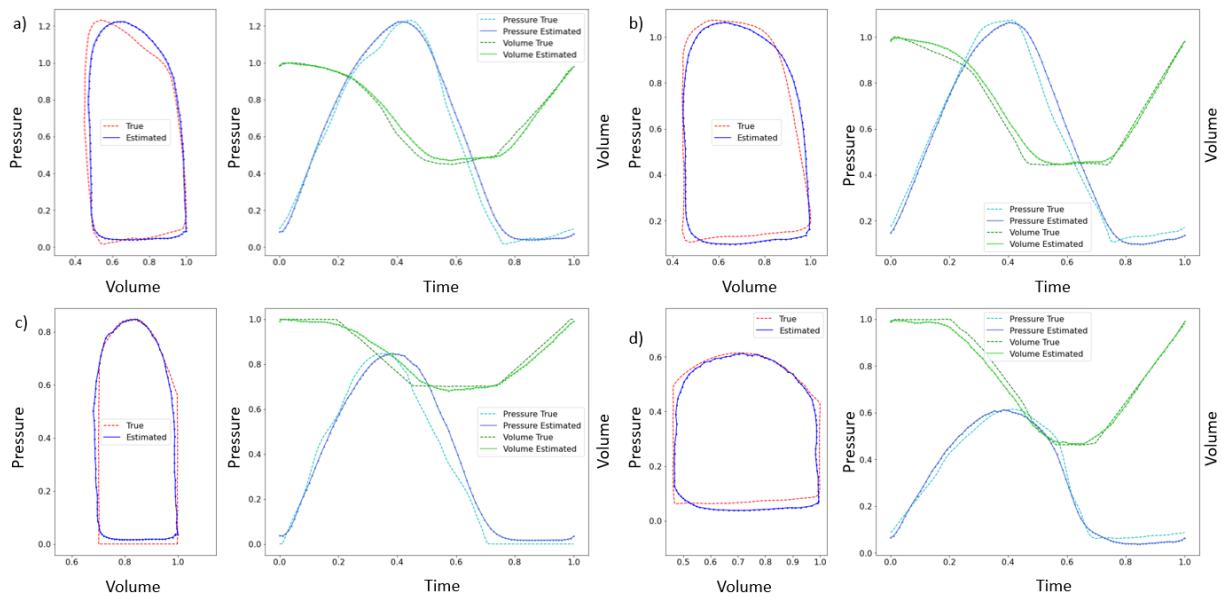


Figure 4.9 Comparison between PV loops taken from our testing dataset and generated by PVLOPED.

We discovered, however, that the model was unable to generate PV loops with a low normalized pressure (<0.07). This might be because the model managed to learn the general characteristics of a PV loop and most examples in the training and validation datasets comprised curves with higher pressure levels. Given that this model would be used only to generate curves for verifying the myocardium model, this minor shortcoming was considered acceptable, and the model was later included in the study. It is worth noting that subsequent research should include an in-depth examination of

hyperparameters, including testing of different layer connections (number of units), the number of shared layers, and additional layer types, such as normalization layers and RNNs.

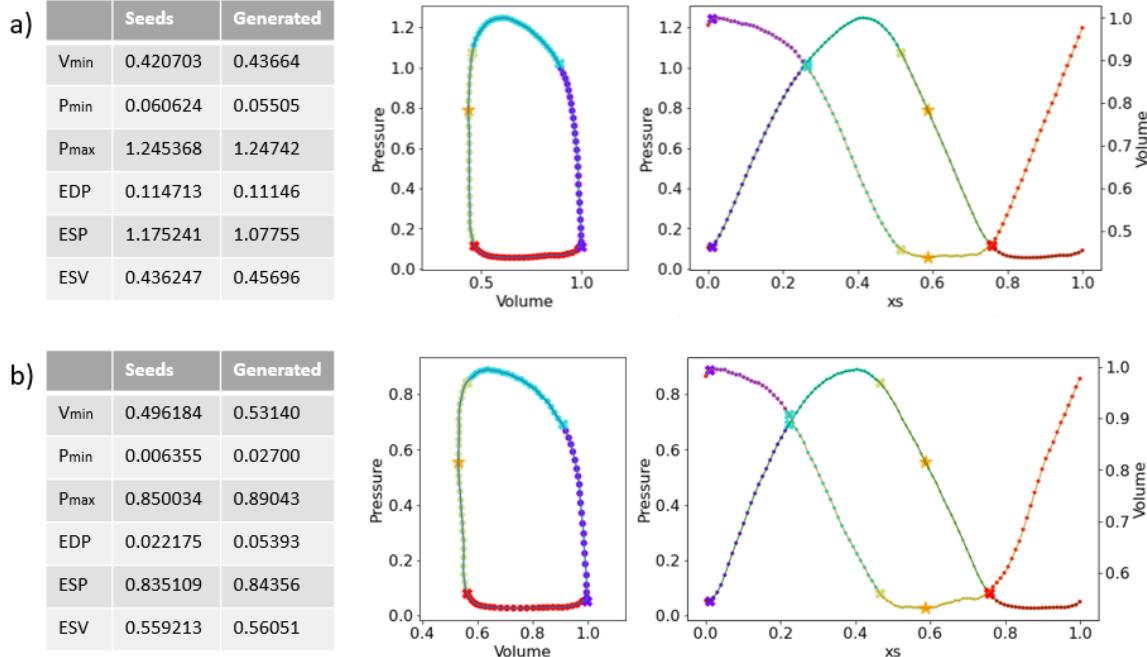


Figure 4.10 Generating PV loops using cardiovascular data with the PVLOOPED model.

4.3 CMM4

The Clinical Metrics Model with four inputs: pressure, volume, epicardium and endocardium fiber orientations (hence the name CMM4) is primarily used to regress the dataset defined in 4.1.1 to determine the associated myocardium metrics (ejection fraction, apex torsion, longitudinal shortening, radial shortening, and wall thickness).

4.3.1 Model Architecture

The first layer of the model is the TensorFlow's Normalization layer, which will ensure that its inputs are driven into a standard deviation of 1 centered around zero. This is accomplished by calculating the mean and variance of the data in advance and then using the following normalization (equation 4.1) at runtime [53].

$$\text{Normalization Layer output} = \frac{\text{Input} - \text{Mean}}{\sqrt{\text{Variance}}} \quad (4.1)$$

The remainder of the model is composed of Dense layers with *relu* activation and Dropout layers, with the exception of the final layer, which is a single Dense layer of 5 units with no activation function. In total, the model has 1,347,914 parameters, in which 4 are inputs and 5 are outputs, resulting in 9 non-trainable. The structure of the CMM4 model and the number of units per layer is illustrated in Figure 4.11.

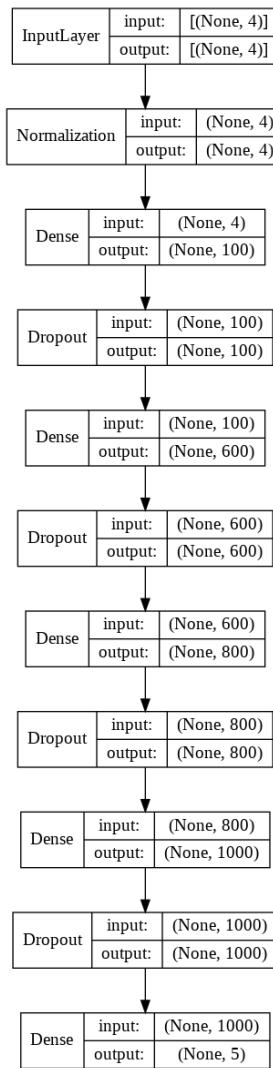


Figure 4.11 CMM4 implemented model structure with respective layer connections information.

4.3.2 Training, Validating and Testing

The CMM4 model was trained in large batches of 500 datapoints using the dataset specified in 4.1.1. The dataset was randomized and divided into train, validation, and test sets of 60%, 20%, and 20% splits, respectively. Additionally, random interleaving and batch shuffle were used to maximize the likelihood that the model would be fed with randomized data, thus promoting generalized learning, and reducing overfitting.

The Adam optimizer was used, with the learning rate calculated by an exponential decay scheduler beginning at 1e-3 and decaying at a rate of 0.88 for 299 steps. The model was trained over 500 epochs using TensorFlow's early stopping callback, which terminates training if the algorithm detects no meaningful learning improvement within a window size of n epochs (in this case, 8), thus minimizing the possibility of overfitting. Moreover, mean squared error (MSE) was used as the model's loss function. Finally, MSE, as well as mean absolute error (MAE) and stateful root mean absolute error (RMSE), were used to assess the model's efficiency. Figure 4.12 summarizes the training of the CMM4 model.

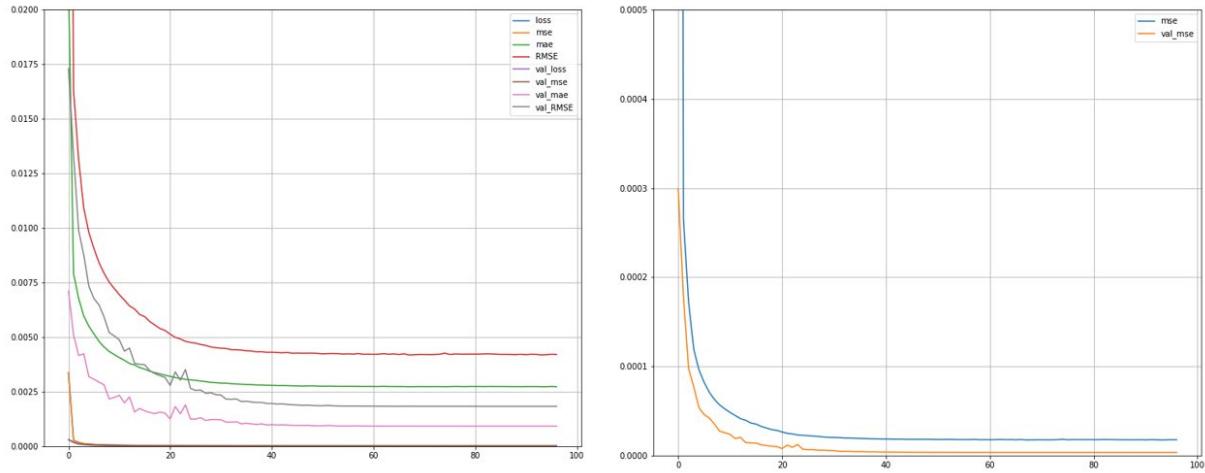


Figure 4.12 Training and validating (left) loss and (right) MSE of the CMM4 model.

The overall validating loss was constantly lower than of the training, as expected due to the usage of dropout and from the behavior of a regression model. Moreover, the plotted results also indicate that no overfitting occurred. The final MSE was 3.4205e-06, while the stateful RSME was about 0.0018. Overall,

the model was able to capture the strongest relationships between the input values (PVL and fiber directions) and the corresponding clinical metrics. Figure 4.13 illustrates the accuracy of the predictor through a scatter plot, in which the true values are on the horizontal axis and the predicted values are on the vertical axis and the red dotted line indicates the optimal result (when true values are equal to predicted values).

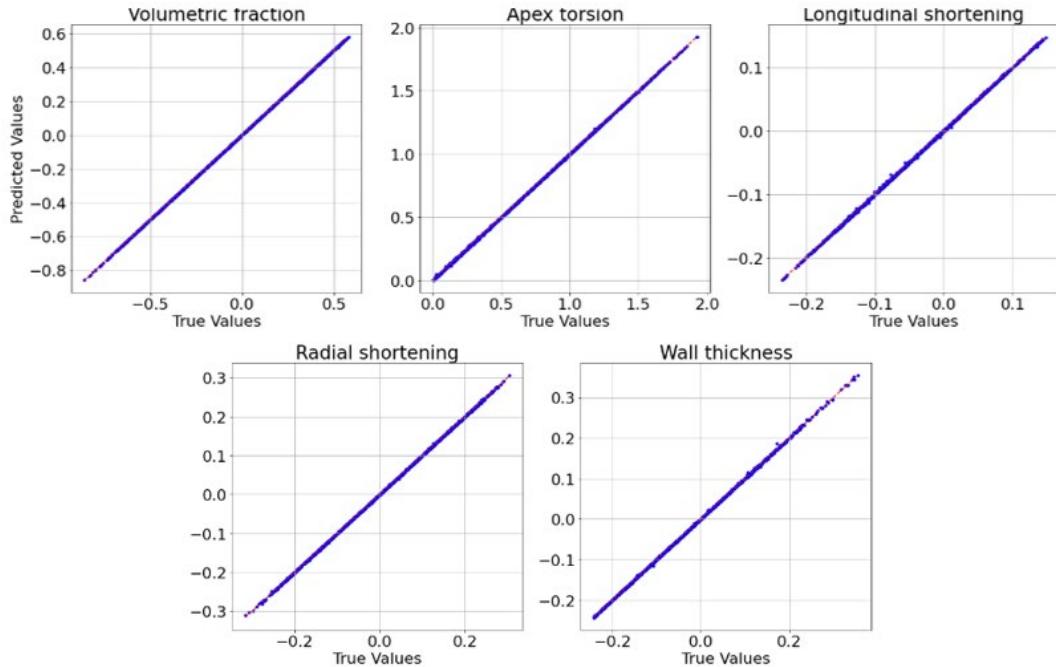


Figure 4.13 Comparison of true and predicted values in testing set of the CMM4 model.

As it can be seen, the predicted values are closely aligned with their true values, indicating the high model accuracy. It is worth noting that the prediction with least precision is the wall thickness, which could be explained by having the largest amount of noise in the dataset. Moreover, the low RMSE score, along with high accuracy precisions, indicated the model was applicable to our study and no hyperparameter tuning was performed. For future analysis, it would be interesting to understand how different parameters, such as learning rate, optimizer, or number of units in the dense layers would affect the model's performance and if another set of parameters would increase the model's accuracy for all predicted values.

4.4 GM4

During initial study, it was discovered that values of gamma were highly related to the fiber directions and could be defined solely by this set of parameters with high accuracy. With this logic in mind, we created a deep learning model, that accepts a pair of epicardium and endocardium fiber orientations, as well as a series of pressure-volume datapoints, and outputs the corresponding gamma value. The model was named based on the output and the number of input parameters, hence GM4.

4.4.1 Model Architecture

Given the fact that the GM4 model functions as a regressor and exhibits behavior close to that of the CMM4 model, its architecture is similar to that of the latter. It needs four inputs: two fiber directions (epicardium and endocardium) and pressure and volume values. It then proceeds to the Normalization layer defined in the preceding section; followed by a sequence of Dense layers with *relu* activation function and Dropout layers. This model's output layer contains a single unit that is responsible for returning the gamma value for the stated inputs. Figure 4.14 portrays the overall model structure along with the number of units in each layer.

4.4.2 Training, Validating and Testing

The GM4 model was trained in large batches of 500 datapoints, similar to the CMM4 model, using the dataset defined in 4.1.1. The dataset was randomly divided into train, validation, and test sets with splits of 60%, 20%, and 20%, respectively. Similarly, another tool used to enhance model training efficacy was random interleaving and batch shuffle, which allowed training with truly randomized data to increase generalization, which in turn decreased overfitting.

Moreover, the Adam optimizer was used, with an exponential decay scheduler setting the initial learning rate to 1e-3 and decaying at a rate of 0.90 for 307 steps. TensorFlow's early stopping callback with patience set to 7 was used to train the model over 500 epochs. Additionally, the mean squared error

(MSE) was used as the loss function for the model and its efficiency was evaluated using the MSE, mean absolute error (MAE), and stateful root mean absolute error (RMSE).

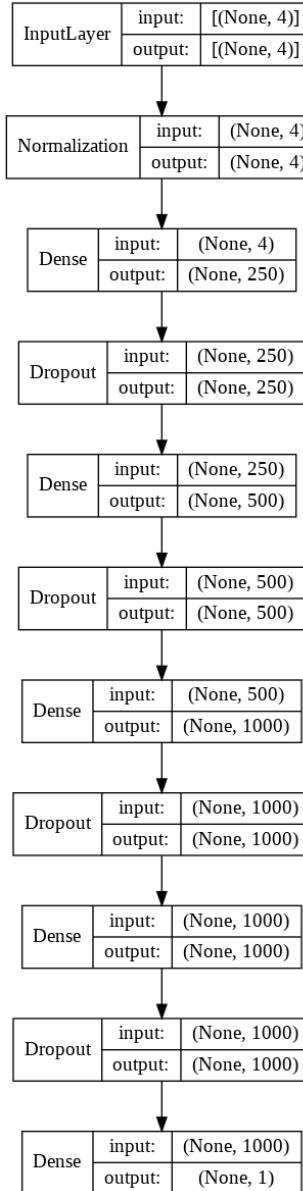


Figure 4.14 GM4 implemented model structure with respective layer connections information.

An hyperparameter tunning was performed to discover the best model architecture. The main hyperparameters were the number of hidden layers and units for each of them, as well as their activation function and the dropout rate used for all dropout layers. The full list of hyperparameters evaluated is shown on table 4.3:

Table 4.3 Hyperparameters explored for tuning GM4 model.

Hyperparameter	Values
Number of units in first layer	100, 200, 300
Number of units in second layer	200, 300, 400
Number of units in third and (if included) forth layer	400, 500, 800, 1000
Include forth layer	Yes, No
Activation function	<i>relu</i> , <i>selu</i>
Dropout rate	0.0001, 0.001, 0.01

The diverse MSE found for each combination of hyperparameter is illustrated in Figure 4.15.

Results indicate that *selu* activation function leads to higher MSE for both training and validating. The same reasoning is applied to high values of dropout rate (0.001 and 0.01). Moreover, the model achieved best performances when the fourth layer was included with high number of units (800 and 1000) for both the third and fourth layers. Moreover, the models with lowest MSE values were found with the first layer containing 200 and 300 units while the second layer was made up with 400 units.

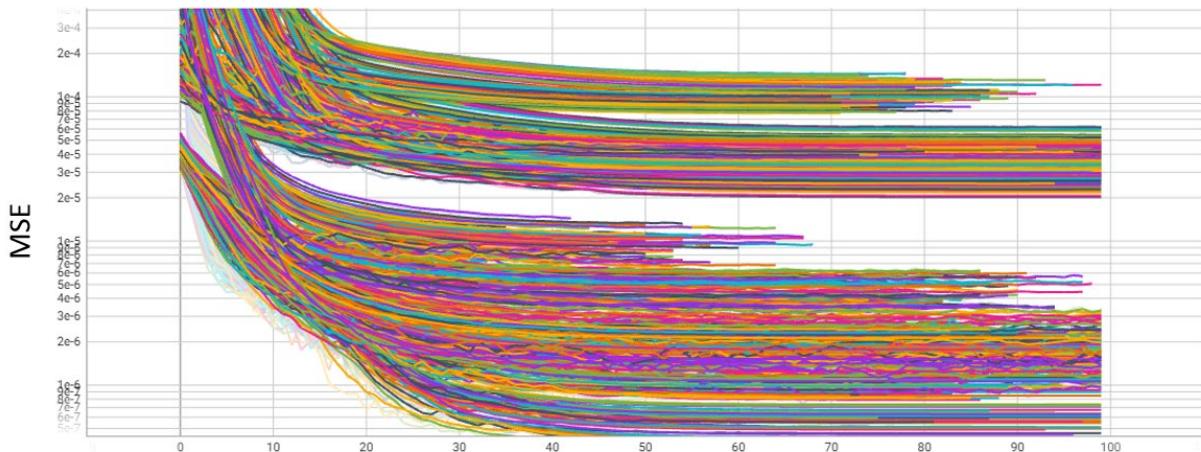


Figure 4.15 MSE values found during hyperparameter tuning of the GM4 model.

Considering these results, a few additional tests were performed to perceive whether the model's performance could be increased until we discovered that the model described in the previous section was able to achieve the lowest MSE. Figure 4.16 summarizes the training process for the final model. As it can be seen the validating loss was constantly lower than of the training, as expected due to the usage of

dropout. Moreover, the plotted results also indicate that no overfitting occurred. Using the testing dataset, we discovered the model's MSE was 2.205e-07, while the stateful RSME was about 4.6957e-04.

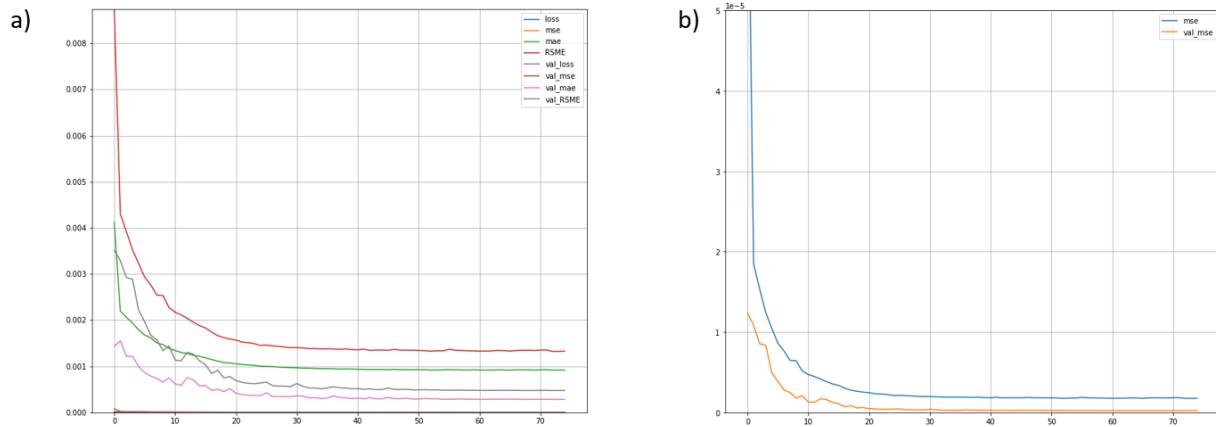


Figure 4.16 Training and validating (left) loss and (right) MSE of the GM4 model.

Table 4.3 shows a few comparisons between true and estimated values obtained by the GM4 model when testing with an unforeseen dataset. As expected, the predicted values are extremely close to their true counterparts.

Table 4.4 Comparison of true and predicted values of gamma using GM4 model in random testing values.

True	Predicted
0.24687	0.24695
0.01576	0.01627
0.12081	0.12107
0.02364	0.02364
0.11818	0.11856
0.24162	0.24171
0.03414	0.03420
0.24949	0.24938
0.11030	0.11022

4.5 GWM9

The final component of this study was the deep learning model used to discover the relationship between the active parameter gamma in the constitutive model, the orientation of the fibers, the myocardium-specific clinical metrics (CMs), and their temporal relationships with PV loops. The primary

objective of the model is to expand on the GM4 concept by generating a gamma waveform for a given PV loop along with clinical metrics at end systole, whereby the model autonomously selects the best-fitting set of fiber orientations (in both epicardium and endocardium). Additionally, since the PV loop is sequential, it is critical that the final model recognize not only the punctual correlations between the PV curve and the output gamma waveform, but also their temporal characteristics. As a result, the proposed model is built on recurrent neural networks, which, as discussed in Section 2.5, are capable of processing sequences of any length and recognizing underlying temporal associations.

In short, the model embraces nine inputs: seven clinical parameters, two of which are the values of pressure and volume at end systole, and a PV loop (of shape [any, 2]), and outputs the corresponding gamma waveform (hence the name GWM9), as well as the epicardium and epicardium fiber orientations. The model was implemented using TensorFlow 2.0 functional models and trained using the data described in 4.1. Due to the fact that the PVLOOPED model was developed later in the project to assess gamma waveforms for specific clinical conditions, the GWM9 model was trained using randomly generated curves.

Furthermore, to analyze the gamma waveform's formulation, the GM4 model was used to produce 'true' gamma waveforms artificially. At this stage, one might suggest why not simply use the GM4 model, which was used to train the GWM9 model. The primary reason for this is attributable to two facts: (1) Although the GM4 works well when a set of fiber orientations is well defined, the study's primary objective is to automatically extract this information (rather than to provide it), and the optimal way to estimate fiber directions is to train a model that converges to an optimal solution for both the final gamma waveform and its fiber orientation simultaneously; and (2) as previously noted, the final model should be capable of handling PV loops of varying lengths, which necessarily involves the model adopting the shape of a recurrent network (or other architecture that handle sequential data, but, for this study, we limited to recurrent networks only).

4.5.1 Custom Layers

The GWM9 model has two tailored layers that were built to increase its performance. The first custom layer is a slightly modified variant of the residual layer structure, as shown in Figure 4.17. It is primarily composed of three principal alignments of dense, dropout, layer normalization, and activation blocks (shown in yellow). Although blocks M and N are virtually similar, they lack their own activation functions (which are substituted by the gate modules) and are thus seen in orange.

The first alignment, denoted by the letter Z and shown by the diagram's right column, functions as a standard series of three fully connected blocks. The middle block, M, and the first gate, which is regulated by an activation mechanism, comprise the second alignment. The final alignment is an extension of block M and is bounded by block N and the second gate, that is also controlled by a function. The outputs of the two gates are first multiplied element-by-element and then added to the output of Z. Finally, the final output is forwarded to the network through the final activation mechanism. The primary advantage of this architecture is that it adds knowledge to the layer's output that may have been missed in the fully connected alignment, similarly to a residual network; however this information is filtered by the two gates that serve as active learners during training, so that only meaningful knowledge is shared.

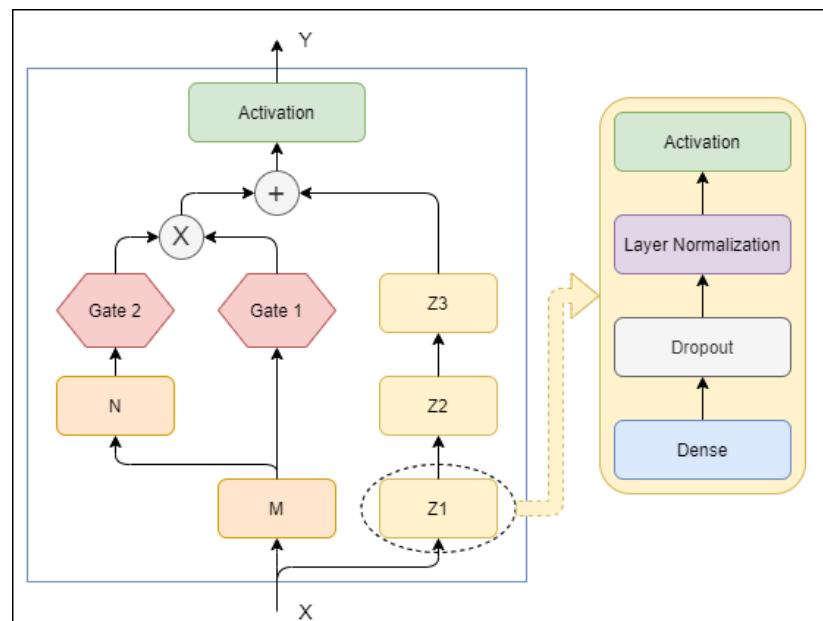


Figure 4.17 Schematics of a Gated Residual Layer with details of its main building block.

The second custom layer is a wrapper for a standard RNN layer rather than a new layer class. One of the most critical factors of RNNs is their default starting state, which is zero for the majority of applications and computational implementations. Bringing this into consideration, although RNNs make extremely accurate predictions for sequential data, their initial collection of predictions is imprecise when evaluating sequences that do not commence at zero, culminating in error propagation and poor model evaluation since the model can never make accurate predictions at its preliminary steps. To address this problem, we can "warm" its start by merely repeating, mirroring, or reflecting its inputs and then conducting regular predictions. Belatedly, we crop the additional initial data before sending it forward in order to maintain a consistent data size. The schematic mechanism depicted in Figure 4.18 shows how this wrapper configuration is implemented. We have introduced additional features for potential use, such as masking and mask consumption for padding, as shown in the figure.

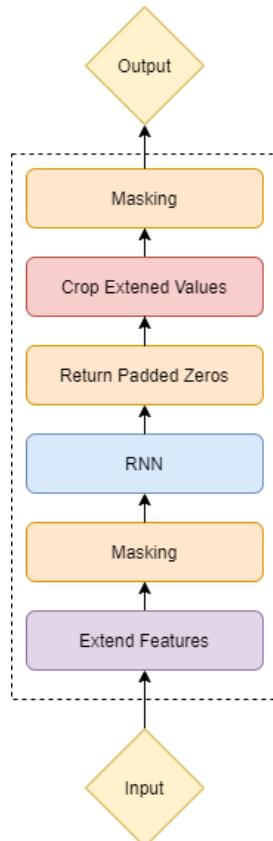


Figure 4.18 Schematics of a 'Warm-start' Recurrent Neural Network layer wrapper.

4.5.2 Model Architecture

The GWM9 Model consists primarily of two components: a fibers regressor and a recurrent neural network. As defined previously, the first is responsible for choosing the optimal selection of fibers depending on the clinical metrics associated with the myocardium (4.4). While the second seeks to discover the fundamental gamma waveform from within the collection of inputs. The overall model diagram is depicted in Figure 4.19.

As it can be observed, the CMs' signals, combined with the PV values at minimum volume (PVL) are directly sent to the fiber regressor, which generates a set of fibers. A copy of values of the fibers are released as an output of the model, while another is concatenated with the PVL. These are then broadcasted to match the PV's data length and then are concatenated with it, as shown by the purple box. The information then passes through the recurrent network, which analyses the entirety of the data and sends the final gamma waveform output.

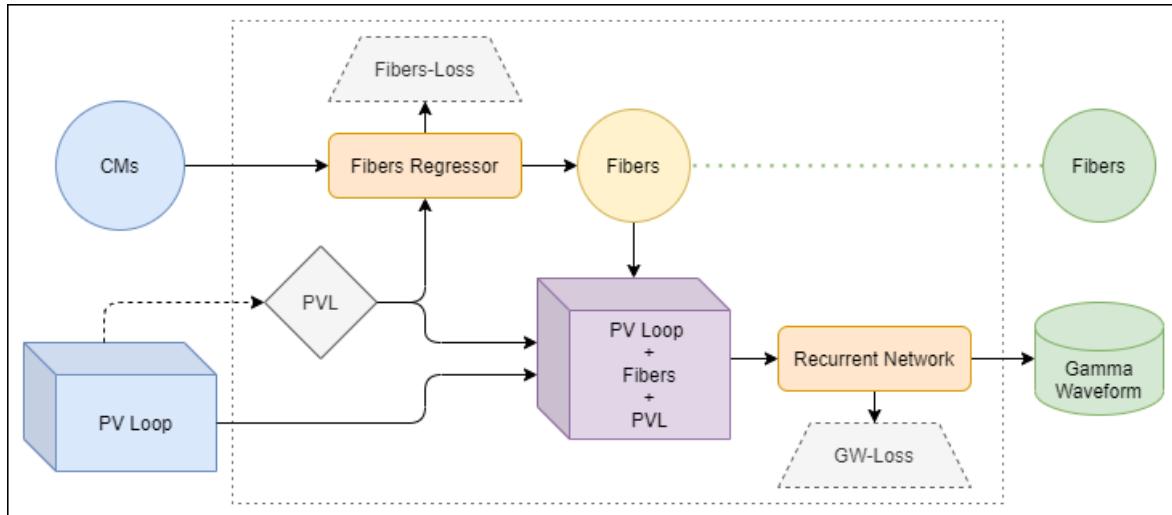


Figure 4.19 General schematic of the GWM9 model.

Alternatively, Figure 4.20 shows in-depth information on how these connections were implemented in TensorFlow. In contrast to the previous schematic, to facilitate the model's implementation, PVL datapoints were extracted before sending them to the model and are already concatenated to the CMs before sending it to the model.

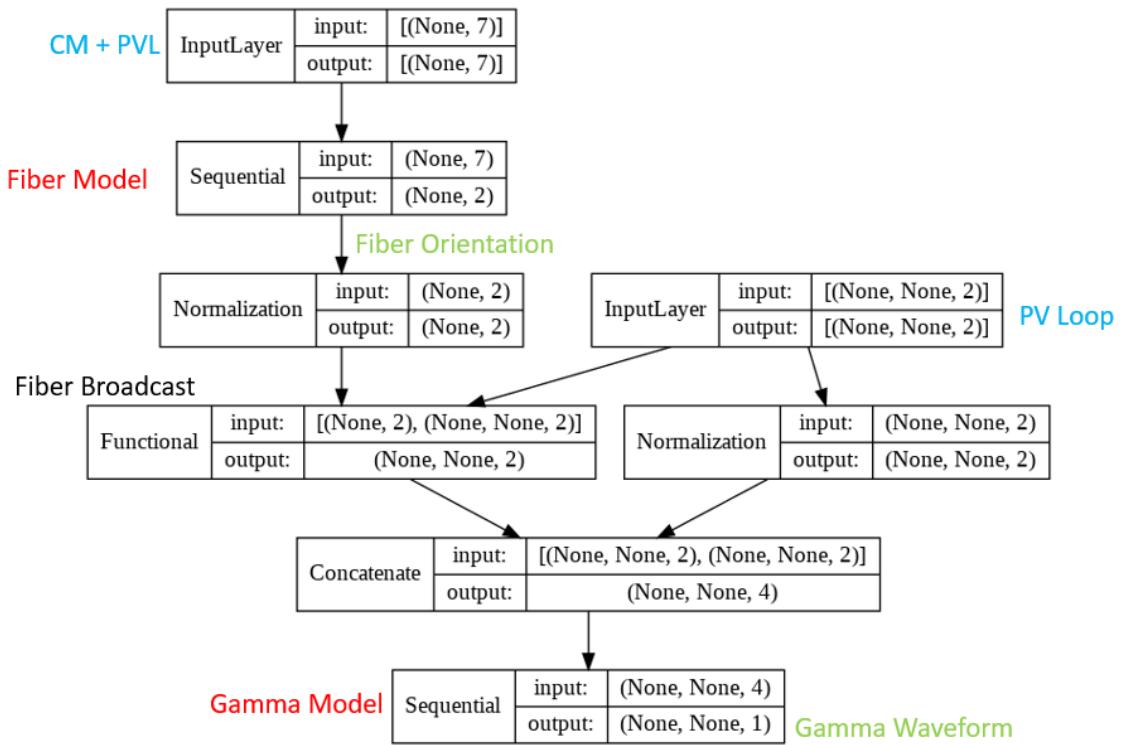


Figure 4.20 GWM9 implemented model structure with respective layer connections information.

Additionally, the model's training consists primarily of two losses: the fiber loss, which compares predicted fibers with their true values, and the gamma waveform loss, which compares estimated gamma waveform values to theoretical values obtained using the GM4 model.

Furthermore, preliminary trials indicated that simple fully connected layers were incapable of achieving appropriate convergence for fiber analysis. In summary, although the model was capable of retrieving general details about the data, it had an MAE of approximately 12-15 degrees in the fibers dimension. To ensure data was not being lost in the process of regressing fiber directions at the same time as minimizing the cost function of the gamma waveform (which was strongly related to the precision of fiber orientations), the Residual Gated layer stated earlier was used in place of dense layers in the fiber regressor. Immediately, the error was diminished to about 2-5 degrees. Similarly, the suggested "Warm-start" RNN wrapper was used to increase the gamma waveform's accuracy.

4.5.3 Training and Validating

To train GWM9 and its sub-models (the fibers regressor and gamma waveform recurrent network) a modified version of the dataset described in 4.1.1 and 4.1.2 was used. The dataset generator first randomly generated a PV loop using the generating function describes in 4.1.2, which then extracted the PVL information (pressure and volume at end systole). Next, it generated a series of fiber orientations following an algorithm to randomly generate fibers in a specified manner: the range of allowed fiber directions for the endocardium and epicardium was first divided into subgroups that each served as boundaries for the generation of fibers, where each of these were then combined with each other to form a randomized dataset that represented the entire domain of possible fibers. For each pair of fiber directions and PVL, the CMM4 model was used to acquire the corresponding myocardium clinical metrics (CMs).

Therefore, for each PV loop generated, its PVL was extracted and the CMs were computed to serve as the input for the GWM9 model, along with the PV loop itself. Moreover, to be able to evaluate and train the gamma waveform, the GM4 model was used to compute the ‘true’ values, which was then used as the labels during training, validating, and testing. Figure 4.21 illustrates the training schematic. As it can be seen, the PVL information is extracted from the PV loop and used with the CMM4 model to gather the CMs. At the same time, fibers are randomly generated and used as inputs for the CMM4 model and the GM4 model. Moreover, the PV loop is used as input for both the GWM9 and GM4 models. The final outputs of the GWM9 model are the fiber orientations and the gamma waveform. The latter is compared with the outputs of the GM4 model with the same fiber orientations. It is important to mention that the weights of GM4 and CMM4 models were frozen as they functioned as fully trained models.

Moreover, it is worth mentioning that a normalized version of the Mean Squared Error low function was used to consider the difference in unit size between gamma values and fiber directions. Early trials were performed to validate model convergence that provided promising results. However, to further investigate the capabilities of the model, two main hyperparameter studies were performed. The first was mainly used to observed how certain constructing parameters of the model, such bidirectionality,

inclusion of LayerNormalization, type of RNN cell and number of units in each cell, number of blocks used for the RNN sub-model (each block was composed of two RNN layers followed by a LayerNormalization if selected as the hyperparameter), would affect the model's performance.

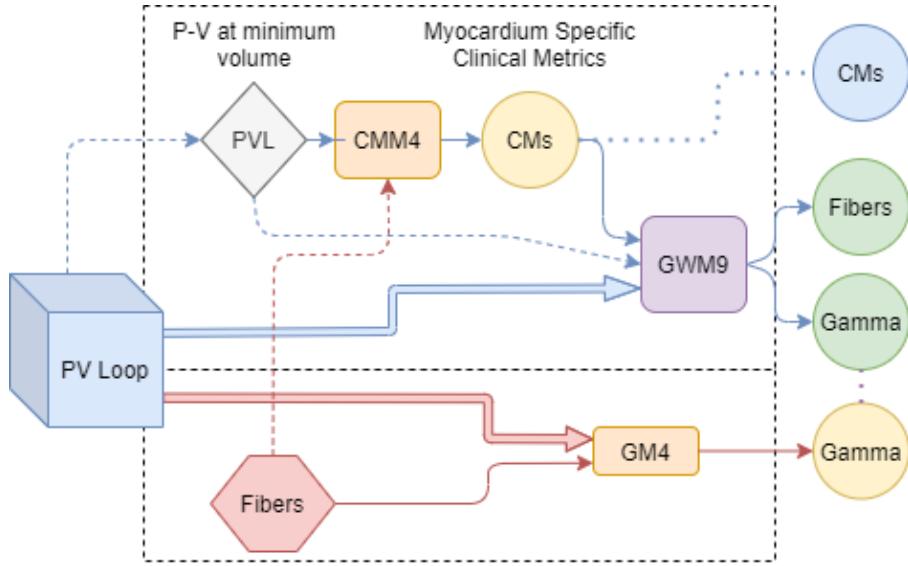


Figure 4.21 Training schematics of the GWM9 model illustrating the usage of CMM4 and GM4 models.

To illustrate the hyperparameter search, Figure 4.22 shows the variation for fiber and gamma waveform losses during training and validating throughout the hyperparameter study. As it can be seen, the training of the fibers was slightly more complex than the training of the gamma waveform, however, all models converged to a loss of approximately 0.002 for the sub-fibers model and 0.0005 for RNN sub-model.

As the first hyperparameter search indicated, the usage of LayerNormalization does not benefit the model, all other combinations reached lower MSE without it. Moreover, it is shown that the GRU cell type of RNN have higher performance than the LSTM cells. The usage of bidirectional RNNs slightly improved the overall model's performance, however, their benefit is not significant and does not justify the added computational complexity. Similar reasoning can be reached when observing the number of blocks and number of RNN units, where the lowest number tests is sufficient to achieve expected model's performance.

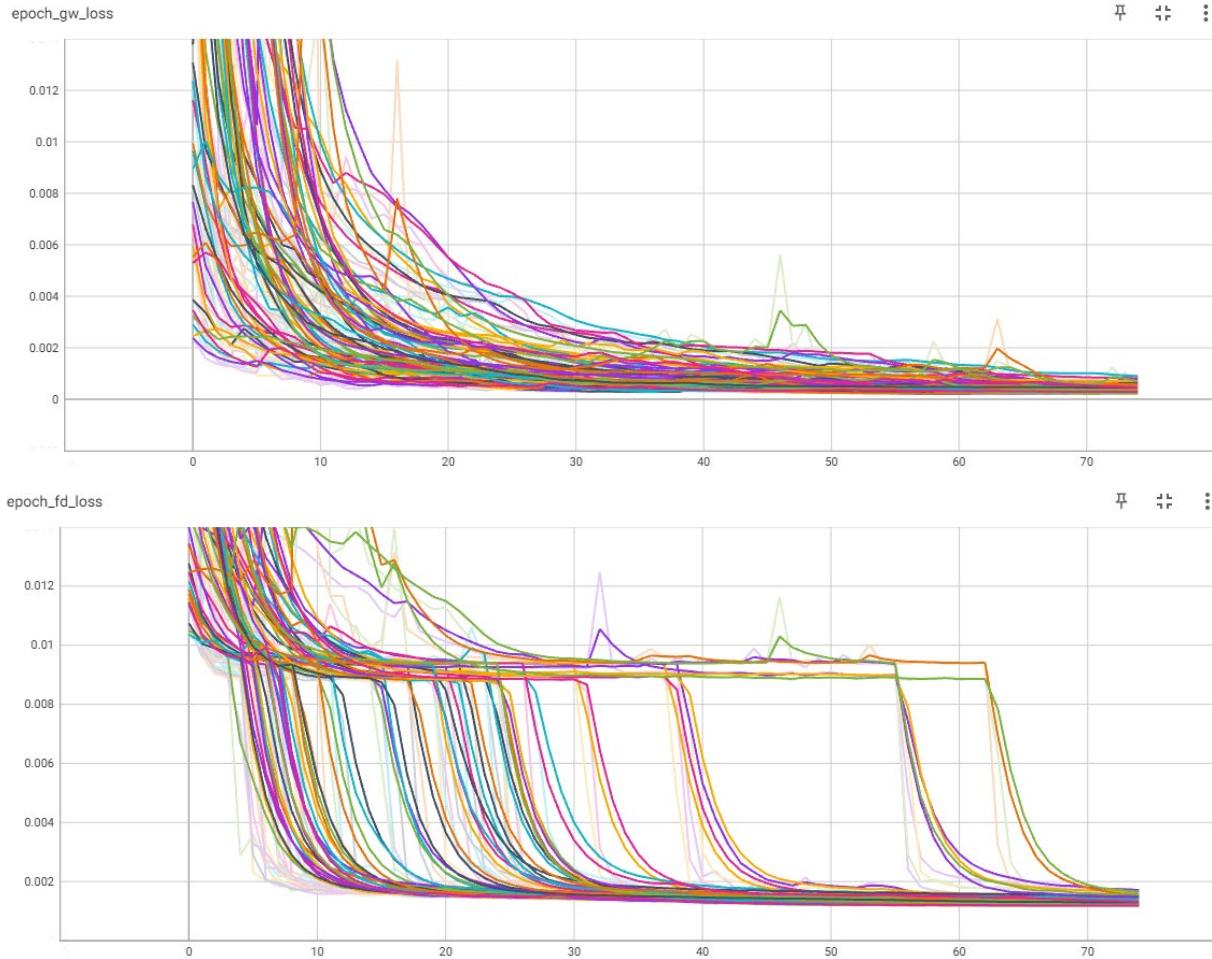


Figure 4.22 Validation and training losses of gamma waveform (top) and fiber orientation (bottom).

The second hyperparameter analysis involved a quick analysis of the number of units of the time-distributed dense layers, located after the RNN layers. This analysis indicated that these hyperparameters did not directly affect the computation of the gamma waveform, therefore the lowest values for each dense layer were selected, since they result in lower computational complexity.

The evaluation of the best-performing model during hyperparameter tuning is shown in table 4.4. As it can be seen, validating values for both fiber orientation and gamma waveform were lower than those of training, indicating that no overfitting occurred.

Furthermore, as the gamma waveforms used as ‘true’ values during the training of this model were generated based on the dataset described in 4.1.1 through the usage of the GM4 regressor, evaluating the model using our testing dataset does not directly reflect the actual model’s performance on

real case scenarios. Instead, to perform the final evaluation, we used the method discussed in the next section, in which the model's gamma waveform is used within the FE- simulation to compare with 'true' PV loops. Considering the online feature of the GWM9 model, where new streams of data can be used to re-train the model and enhance its performance, further study is suggested to train the model using new FEA data gathered by simulations created with the model's prediction of gamma waveform.

Table 4.5 Evaluation of GWM9 model during training and validating when using best parameters identified by the hyperparameter study.

	MSE - Training	MSE – Validating	RMSE - Training	RMSE - Validating
Fiber Orientation	4.2483	0.91917	2.0611	0.95873
Gamma Waveform	0.000025504	0.000013850	0.0050501	0.0037216

Chapter 5: Summary of the Model and its Effectiveness

The main goal of this study was to establish a physics-based deep learning model that is able to predict the parameters in the constitutive modeling of left ventricular myocardium based on few clinical metrics. Meanwhile, we expect the proposed constitutive model could capture the behavior of the myocardium deformation in a cardiac cycle. The main output of the model is a material parameter (gamma) directly related to active contraction properties and correlated fiber orientations in the myocardium that allow the heart to contract according to a specified pressure-volume (PV) relationship. Thus, for this model to work, a PV loop needs to be supplied, and, due to the absence of available data, a deep learning model was designed to generate realistic PV loops based on specified cardiovascular data.

The full schematic is presented in Figure 5.1, in which the required inputs are illustrated in blue on the left side, while the outputs are shown in green on the right. As a set of cardiovascular data is provided to the PVLOOLED model in (a), a PV loop is generated. From this, a feature denoted as PVL that indicates the pressure-volume datapoint at the lowest volume of the PV loop is extracted and supplied to the GWM9 model. At the same time, a set of myocardium-related clinical metrics is also supplied to the GWM9 model. This model, in turn, computes the related fiber orientations presented in the myocardium, as well as the active's model gamma waveform.

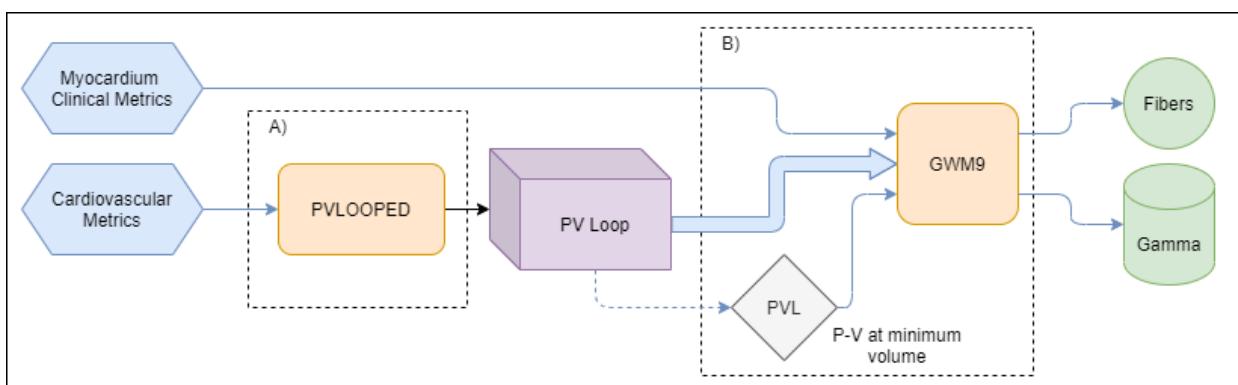


Figure 5.1 General schematic of PVLOOLED and GWM9 model usage.

From these results, FEA simulations using the proposed active constitutive model combined with the gamma waveform and a fiber field generated with the computed fiber orientations can be performed. The final evaluation of this entire process can be measured based on the comparison of volume information extracted from FE simulations and input PV loop, as well as the comparison between the input clinical metrics and those obtained during the FEA simulation. The evaluation process is illustrated in Figure 5.2. In addition, as previously discussed, this evaluation method can also be used to further enhance the capabilities of the GWM9 model by training and evaluating with data gathered by the FEA simulation.

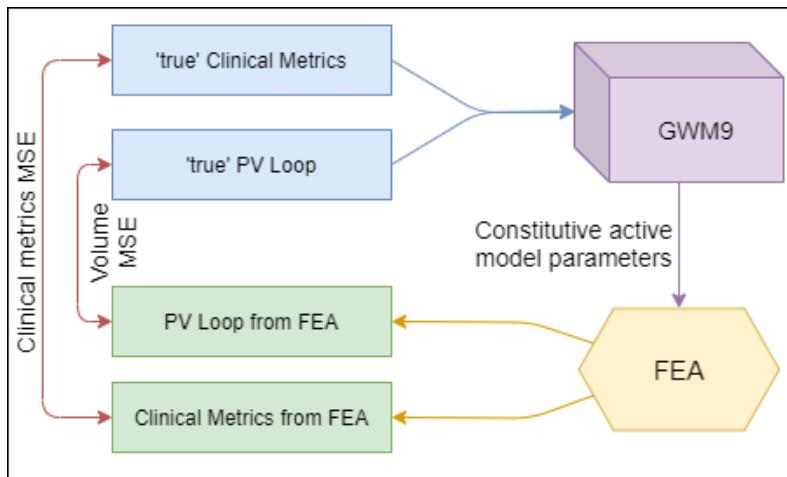


Figure 5.2 Procedure for evaluating the constitutive model parameters estimated by our deep learning model using the results obtained by FEA simulations.

Figures 5.3 to 5.6 exemplifies this process across different inputs. The first plot and second plot at the top of each figure illustrates the input PV loop for the given example, while the third plot shows the output gamma waveform from the deep learning model by using the given PV loop and clinical metrics (shown on the right). The two plots at the bottom of each figure contrast the ‘true’ and estimated values computed through FEA using the generated gamma waveform; the volumetric waveform is emphasized to evaluate the material’s behavior along the full cardiac cycle. In addition, clinical metrics gathered during FEA are compared with their ‘true’ values (input parameters) on the right, along ‘true’ with the values of fiber orientations and the ones obtained through the deep learning model.

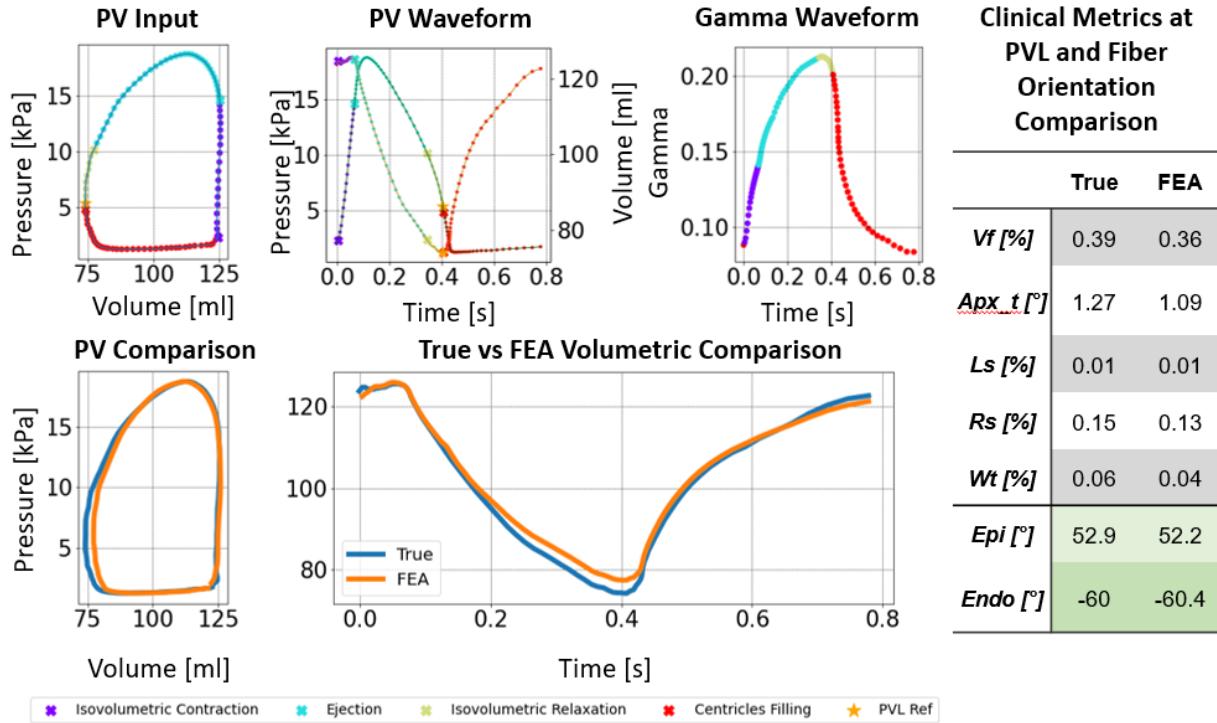


Figure 5.3 Sample comparison (1) of results from constitutive model FEA simulation using predicted gamma waveform from GWM9 model and ‘true’ value of a PV loop generated using PVLOOPEP model.

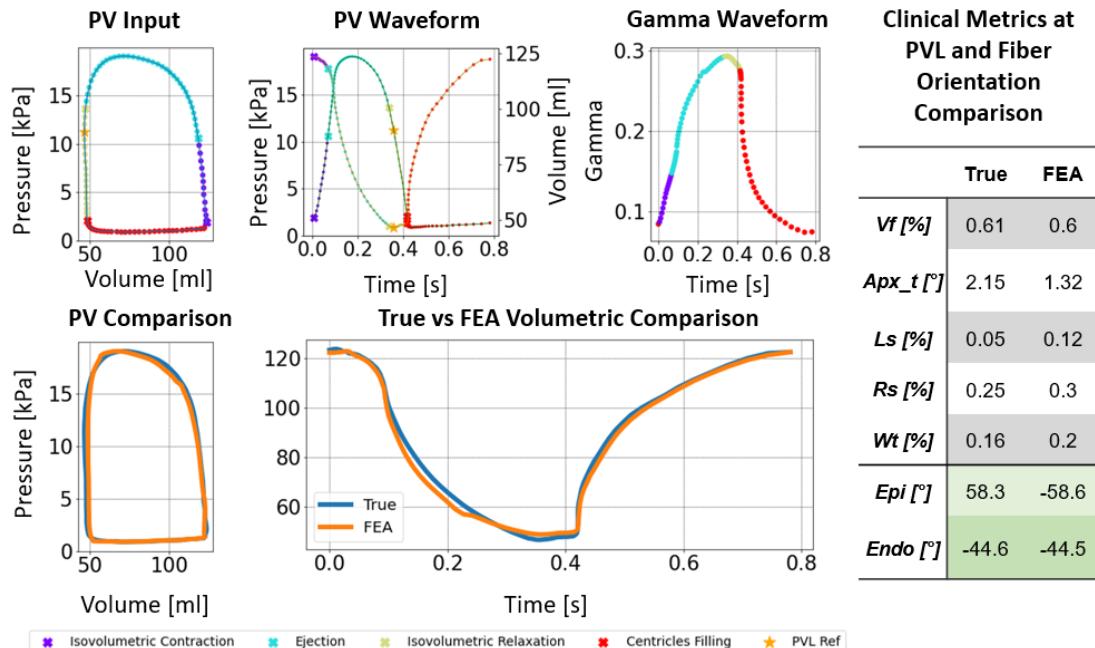


Figure 5.4 Sample comparison (2) of results from constitutive model FEA simulation using predicted gamma waveform from GWM9 model and ‘true’ value of a PV loop generated using PVLOOPEP model.

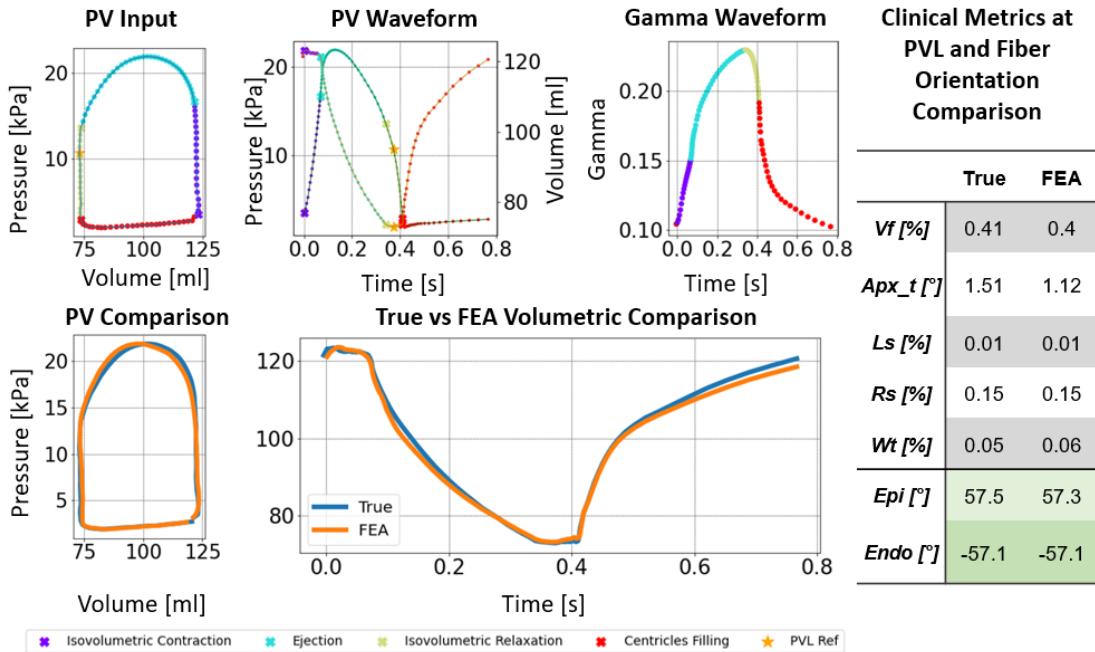


Figure 5.5 Sample comparison (3) of results from constitutive model FEA simulation using predicted gamma waveform from GWM9 model and ‘true’ value of a PV loop generated using PVLOOPED model.

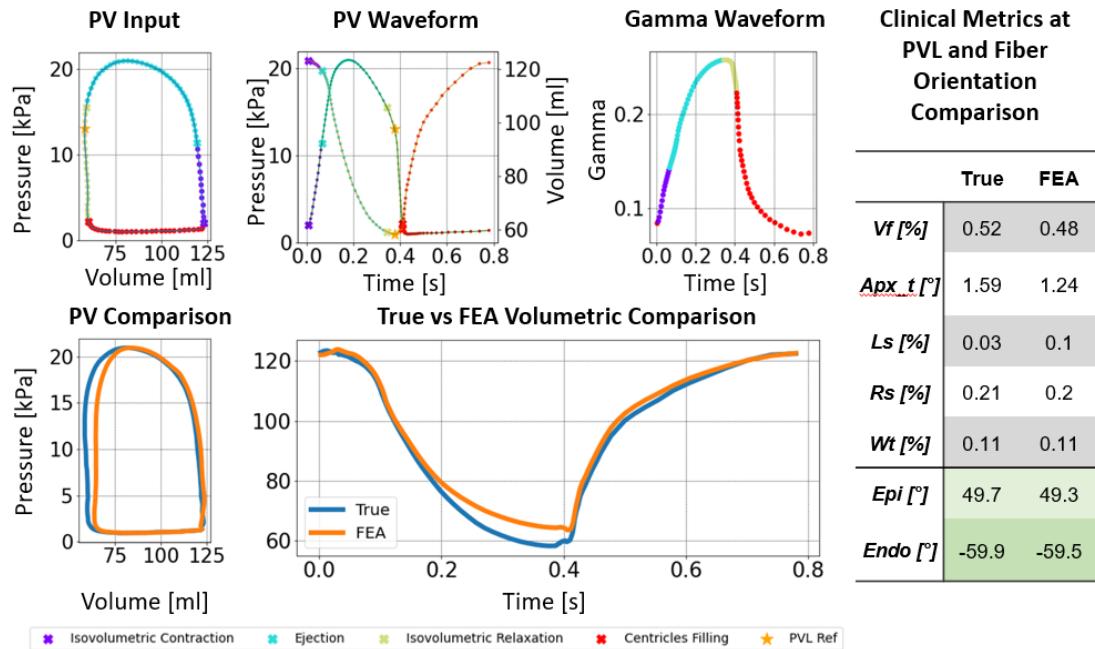


Figure 5.6 Sample comparison (4) of results from constitutive model FEA simulation using predicted gamma waveform from GWM9 model and ‘true’ value of a PV loop generated using PVLOOPED model.

This evaluation process (shown in figures 5.1 and 5.2 and exemplified by figure 5.3 to 5.6) was repeated several times and the result of 94 trials were recorded and used to evaluate the study and are shown in table 5.2. As it can be seen, the study culminated in accurate representation of the behavior of the left ventricular myocardium along diverse cardiac cycles: it presented low RMSE when comparing the volumetric waveforms between ‘true’ values given by the input PV curve and ‘predicted’ values acquired during FEA. Moreover, the deep learning model was able to estimate the fiber orientations within 1 degree of their original value for most of the cases, resulting in low RMSE. On the other hand, the RMSE found for clinical metrics was relatively moderate, which can be explained due to instabilities found at the apex region during FE simulations when high values of gamma (greater than 0.27) were used.

Table 5.1 Evaluation of study

<i>Part</i>	<i>RMSE</i>
<i>Volume</i>	7.25 (ml)
<i>Fibers orientations</i>	0.971 (°)
<i>Clinical Metrics</i>	0.203

In summary, this work demonstrated the feasibility of applying a deep learning approach to estimate the parameters for the active constitutive model of the left ventricular myocardium using conventional clinical measurements. Further, the suggested model simultaneously produced gamma waveforms and fiber orientations, which were utilized in combination with FE analysis to provide an accurate depiction of the behavior of the myocardium.

Additionally, considering the negligible computational time (less than approximated 151ms on average when using Tesla V100-SXM2-16GB GPU) when applying the GWM9 deep learning model, this approach considerably facilitates the use of constitutive models in clinical applications by offering a fast and effective method for manipulating material parameters to modulate the material behavior of specific

cases. Further optimization of the entire process potentially leads to real-time analysis of stress fields and material deformation of the left ventricular myocardium in patient specific scenarios.

However, because this study used an idealistic form for the left ventricular myocardium, additional research is needed to validate the model's ability to operate with patient-specific geometries. Likewise, to broaden the model's clinical application, forecasting the constitutive model's passive features would be desirable.

References

1. Benjamin, E.J., et al., *Heart Disease and Stroke Statistics-2019 Update: A Report From the American Heart Association*. Circulation, 2019. **139**(10): p. e56-e66.
2. B, A., S. Rao, and H.J. Pandya, *Engineering approaches for characterizing soft tissue mechanical properties: A review*. Clinical Biomechanics, 2019. **69**: p. 127-140.
3. Gasser, T.C., R.W. Ogden, and G.A. Holzapfel, *Hyperelastic modelling of arterial layers with distributed collagen fibre orientations*. Journal of the Royal Society Interface, 2006. **3**(6): p. 15-35.
4. Hoerig, C., J. Ghaboussi, and M.F. Insana, *An information-based machine learning approach to elasticity imaging*. 2017, Springer Science + Business Media: Germany. p. 805.
5. Cilla, M., et al., *On the use of machine learning techniques for the mechanical characterization of soft biological tissues*. International Journal for Numerical Methods in Biomedical Engineering, 2018. **34**(10): p. N.PAG-N.PAG.
6. Holzapfel Gerhard, A. and W. Ogden Ray, *Constitutive Modelling of Passive Myocardium: A Structurally Based Framework for Material Characterization*. Philosophical Transactions: Mathematical, Physical and Engineering Sciences, 2009. **367**(1902): p. 3445-3475.
7. Göktepe, S., et al., *Computational modeling of passive myocardium*. 2011, Wiley-Blackwells: Great Britain. p. 1.
8. Huang, D., et al., *A machine learning based plasticity model using proper orthogonal decomposition*. 2020.
9. Pezzuto, S., D. Ambrosi, and A. Quarteroni, *An orthotropic active-strain model for the myocardium mechanics and its numerical approximation*. European Journal of Mechanics / A Solids, 2014. **48**: p. 83-96.
10. Watson, C., *Cardiovascular Physiology*, in *Human Physiology*. 2015, Jones & Bartlett Learning. p. 294.
11. Talbot, S.R.R.V.T., *Vascular system*. 2019, Salem Press.
12. Last, C.S.S.R.J., *Last's anatomy : regional and applied*. 12 ed. 2006: Churchill Livingstone; 560.
13. Gilbert, S.H., et al., *Regional localisation of left ventricular sheet structure: integration with current models of cardiac fibre, sheet and band structure*. 2007, Elsevier Science B.V., Amsterdam.: Netherlands. p. 231.

14. Dokos, S., et al., *Shear properties of passive ventricular myocardium*. American Journal of Physiology: Heart & Circulatory Physiology, 2002. **283**(6): p. H2650.
15. Lunkenhimer, P.P., et al., *Three-dimensional architecture of the left ventricular myocardium*. The Anatomical Record Part A: Discoveries in Molecular, Cellular, and Evolutionary Biology, 2006. **288A**(6): p. 565-578.
16. LeGrice, I.J., Y. Takayama, and J.W. Covell. *Transverse Shear Along Myocardial Cleavage Planes Provides a Mechanism for Normal Systolic Wall Thickening*. 1995. United States: AMERICAN HEART ASSOCIATION INC.
17. Yin, F.C.P., et al., *Quantification of the mechanical properties of noncontracting canine myocardium under simultaneous biaxial loading*. Journal of Biomechanics, 1987. **20**(6): p. 577-589.
18. Rossi, S., et al., *Orthotropic active strain models for the numerical simulation of cardiac biomechanics*. 2012, Wiley-Blackwells: Great Britain. p. 761.
19. Kelly, D., et al., *GENE EXPRESSION OF STRETCH-ACTIVATED CHANNELS AND MECHANOELECTRIC FEEDBACK IN THE HEART*. 2006, Blackwell Publishing Ltd: Great Britain. p. 642.
20. Kohl, P., P. Hunter, and D. Noble, *Stretch-induced changes in heart rate and rhythm: clinical observations, experiments and mathematical models*. 1998, PERGAMON PRESS: Great Britain. p. 91.
21. Dumensil, J.G. and R.M. Shoucri, *Quantitative relationships between left ventricular ejection and wall thickening and geometry*. Journal of Applied Physiology, 1991. **70**(1): p. 48-54.
22. Lindsey, M.L., et al., *Guidelines for measuring cardiac physiology in mice*. American Journal of Physiology-Heart and Circulatory Physiology, 2018. **314**(4): p. H733-H752.
23. Burkoff, D. *MECHANICAL PROPERTIES OF THE HEART AND ITS INTERACTION WITH THE VASCULAR SYSTEM*. 2002; Available from: <https://ccnmtl.columbia.edu/projects/heart/exercises/MechPropHeart/lecture.html>.
24. Feher, J., *Quantitative Human Physiology*, in *Quantitative Human Physiology*, J. Feher, Editor. 2012, Academic Press: Boston. p. ix-xi.
25. Galderisi, M., et al., *Standardization of adult transthoracic echocardiography reporting in agreement with recent chamber quantification, diastolic function, and heart valve disease recommendations: an expert consensus document of the European Association of Cardiovascular Imaging*. European Heart Journal Cardiovascular Imaging, 2017. **18**(12): p. 1301-1310.
26. Antonini-Canterin, F., et al., *The Ventricular-Arterial Coupling: From Basic Pathophysiology to Clinical Application in the Echocardiography Laboratory*. Journal of cardiovascular echography, 2013. **23**(4): p. 91-95.

27. Cheng, J. and L.T. Zhang, *A General Approach to Derive Stress and Elasticity Tensors for Hyperelastic Isotropic and Anisotropic Biomaterials*. International Journal of Computational Methods, 2018. **15**(04).
28. Boyett, M.R., J.E. Frampton, and M.S. Kirby, *The length, width and volume of isolated rat and ferret ventricular myocytes during twitch contractions and changes in osmotic strength*. Experimental Physiology, 1991. **76**(2): p. 259-270.
29. Nelson, H.H.C., *Building Machine Learning Pipelines*. 1 ed. 2020: O'Reilly Media, Inc.
30. Geron, A., *Hands-On Machine Learning with Scikit-Learn, Keras, and Tensorflow*. 2 ed. 2019: O'Reilly Media, Inc. 818.
31. Kononenko, I., *Machine learning for medical diagnosis: history, state of the art and perspective*. 2001, ELSEVIER: Germany. p. 89.
32. Valliappa Lakshmanan, S.R.M.M., *Machine Learning Design Patterns*. 1 ed. 2020: O'Reilly Media Inc.
33. Weidman, S., *Deep Learning from Scratch*. 1 ed. 2019: O'Reilly Media, Inc. 235.
34. Courville, I.G.a.Y.B.a.A., *Deep Learning*. 2016: MIT Press.
35. Hochreiter, S. and J. Schmidhuber, *Long short-term memory*. Neural Computation, 1997. **9**(8): p. 1735.
36. Vincent, P., et al., *Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion*. 2011, THE MIT PRESS: United States. p. 3371.
37. Kingma, D.P. and M. Welling, *Auto-Encoding Variational Bayes*. 2013.
38. Doersch, C., *Tutorial on Variational Autoencoders*. 2016.
39. Maas SA, L.S., Ateshian GA, Weiss JA, *A plugin framework for extending the simulation capabilities of FEBio*. Biophysical Journal, 2018. **115**(9):1630-1637.
40. Maas SA, E.B., Ateshian GA, Weiss JA, *FEBio: Finite elements for biomechanics*. Journal of Biomechanical Engineering, 2012. **134**(1).
41. Schmid, H., et al., *Myocardial Material Parameter Estimation-A Comparative Study for Simple Shear*. 2006, AMERICAN SOCIETY MECHANICAL ENGINEERS: United States. p. 742.
42. Schmid, H., et al., *Myocardial material parameter estimation; A non--homogeneous finite element study from simple shear tests*. Biomechanics and Modeling in Mechanobiology, 2008. **7**(3): p. 161.
43. Yin, F.C., et al., *Quantification of the mechanical properties of noncontracting canine myocardium under simultaneous biaxial loading*. J Biomech, 1987. **20**(6): p. 577-89.

44. Wang, H.M., et al., *Structure-based finite strain modelling of the human left ventricle in diastole*. 2013, Wiley-Blackwells: Great Britain. p. 83.
45. Di Donato, M., et al., *Left ventricular geometry in normal and post-anterior myocardial infarction patients: sphericity index and 'new' conicity index comparisons*. European Journal of Cardio-Thoracic Surgery, 2006. **29**(Supplement_1): p. S225-S230.
46. Slinde, G.A., *Numerical Modelling and Analysis of the Left Ventricle*, in *Department of Structural Engineering*. 2015, Norwegian University of Science and Technology: NTNU. p. 117.
47. Wong, J. and E. Kuhl. *Generating fibre orientation maps in human heart models using Poisson interpolation*. 2014. Great Britain: Taylor & Francis.
48. Bayer, J.D., et al., *A Novel Rule-Based Algorithm for Assigning Myocardial Fiber Orientation to Computational Heart Models*. 2012, Springer Science + Business Media: Great Britain. p. 2243.
49. Halevy, A., P. Norvig, and F. Pereira, *The Unreasonable Effectiveness of Data*. IEEE Intelligent Systems, Intelligent Systems, IEEE, IEEE Intell. Syst., 2009. **24**(2): p. 8-12.
50. Mueller-Freitag, M., *10 Data Acquisition Strategies for Startups*. 2016: Medium.
51. Wissner-Gross, A. 2016 : *What do you consider the most interesting recent [scientific] news? What makes it important?* 2016.
52. *TensorFlow Datasets: a collection of ready-to-use datasets.*; Available from: <https://www.tensorflow.org/datasets>.
53. *TensorFlow Core v2.5.0 - tf.keras.layers.experimental.preprocessing.Normalization*. Available from: https://www.tensorflow.org/api_docs/python/tf/keras/layers/experimental/preprocessing/Normalization.

Appendix A: Copyright Permissions

The permission below is for the use of Figure 2.3 and Figure 2.4



This is a License Agreement between Igor A.P. Nobrega ("User") and Copyright Clearance Center, Inc. ("CCC") on behalf of the Rightsholder identified in the order details below. The license consists of the order details, the CCC Terms and Conditions below, and any Rightsholder Terms and Conditions which are included below.

All payments must be made in full to CCC in accordance with the CCC Terms and Conditions below.

Order Date	22-Jun-2021	Type of Use	Republish in a thesis/dissertation
Order License ID	1127810-1	Publisher	ROYAL SOCIETY
ISSN	1364-503X	Portion	Image/photo/illustration
LICENSED CONTENT			
Publication Title	Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences. Series A. Mathematical, physical, and engineering sciences	Country	United Kingdom of Great Britain and Northern Ireland
Author/Editor	Royal Society (Great Britain)	Rightsholder	The Royal Society (U.K.)
Date	01/01/1996	Publication Type	Journal
Language	English	URL	https://royalsociety.org/journals/
REQUEST DETAILS			
Portion Type	Image/photo/illustration	Distribution	United States
Number of images / photos / illustrations	2	Translation	Original language of publication
Format (select all that apply)	Electronic	Copies for the disabled?	No
Who will republish the content?	Academic institution	Minor editing privileges?	No
Duration of Use	Current edition and up to 5 years	Incidental promotional use?	No
Lifetime Unit Quantity	Up to 499	Currency	USD
Rights Requested	Main product		
NEW WORK DETAILS			
Title	A Constitutive-Based Deep Learning Model for the Identification of Active Contraction Parameters of the Left Ventricular Myocardium	Institution name	University Of South Florida
		Expected presentation date	2021-06-22
Instructor name	Wenbin Mao, Ph.D		
ADDITIONAL DETAILS			
Order reference number	N/A	The requesting person / organization to appear on the license	Igor A.P. Nobrega
REUSE CONTENT DETAILS			
Title, description or numeric reference of the portion(s)	2, 3	Title of the article/chapter the portion is from	Constitutive modelling of passive myocardium: a structurally based framework for material characterization
Editor of portion(s)	n/a	Author of portion(s)	Royal Society (Great Britain)
Volume of serial or monograph	n/a	Issue, if republishing an article from a serial	N/A
Page or page range of portion	3450,3451	Publication date of portion	2009-09-13