



---

*Research article*

## Estimation of left ventricular parameters based on deep learning method

Li Cai<sup>1,2,3</sup>, Jie Jiao<sup>1,2,3,\*</sup>, Pengfei Ma<sup>1,2,3</sup>, Wenxian Xie<sup>1,2,3</sup> and Yongheng Wang<sup>1,2,3</sup>

<sup>1</sup> Xi'an Key Laboratory of Scientific Computation and Applied Statistics, Xi'an 710129, China

<sup>2</sup> NPU-UoG International Cooperative Lab for Computation and Application in Cardiology, Xi'an 710129, China

<sup>3</sup> School of Mathematics and Statistics, Northwestern Polytechnical University, Xi'an 710129, China

\* **Correspondence:** Email: [jiaoj@mail.nwpu.edu.cn](mailto:jiaoj@mail.nwpu.edu.cn).

**Abstract:** Estimating material properties of personalized human left ventricular (LV) modelling is a central problem in biomechanical studies. In this work we use deep learning (DL) method to evaluating the passive myocardial mechanical properties inversely. In the first part of the paper, we establish a standardized geometric model of the LV. The geometric model parameters are optimized based on 27 different healthy volunteers. In the second part, we use statistical methods and Latin hypercube sampling (LHS) to obtain the geometric parameters data. The LV myocardium is described using a structure-based orthotropic Holzapfel-Ogden constitutive law. The LV diastolic pressure-volume (PV) curves are calculated by numerical simulation. In the third part, we establish the multiple neural networks to predict PV curve parameters. Then, instead of using constrained optimization problems to solve constitutive parameters, DL was used to establish the nonlinear mapping relationship of geometric parameters, PV curve parameters and constitutive parameters. The results show that the deep learning method can greatly improve the computational efficiency of numerical simulation and increase the possibility of its application in rapid feedback of clinical data.

**Keywords:** left ventricular geometric parametric model; deep learning method based on neural network; inverse problem solving

---

### 1. Introduction

The dynamic modeling and simulation of cardiac function using patient-specific three-dimensional (3D) geometric heart structure is studied by a large number of researchers [1]. Estimating the constitutive parameters non-invasively remains a great challenge for the LV modelling community [2].

Various geometric models are used to approximate the shape of the LV. Hanna et al. [3] experimentally dissected the LV of a dog and proposed a method to represent the shape of the LV membranes

with elliptic curves. To study the transmission of ecg and the structure of the LV myocardial fibers, Pravdin et al. [4] improved Hanna's method by fitting the shape of the LV with conic curve formed by spiral curve. To blackuce the complexity of the finite element simulation of the LV, Achille et al. [5] used a rotation curve based on elliptic curve to represent the geometric features of the LV.

In general, finite element method (FEM) is applied to the field of cardiac modeling and calculation. Gould et al. [6] used FEM to study the stress and strain of the ventricular wall. This is the first time the FME has been applied to cardiac modeling. Sermesant et al. [7] calculated the local myocardial stress during systole using MRI images and other cardiac data. Peskin et al. [8] proposed the Immersed Boundary (IB) method. The IB method is often used to solve the problem of fluid-structure interaction and has been widely popularized in the field of biomechanics in recent years. Cai et al. [9] successfully used IB/FE method to numerically simulate LV at different stages of cardiac cycle.

However, it requires several iterations to obtain the optimal solution of the FEM simulation, and the calculation time is relatively long [10–12]. Such high computational costs will inhibit the practical use of these methods. In recent years, DL techniques [13–15] have gained great attention in the field of artificial intelligence [16]. DL model can establish a complex nonlinear relationship between inputs and outputs. A possible solution to the bottlenecks associated with patient-specific computational modeling is to use DL algorithms to speed up the process of material parameter identification. By designing and training a DL model on large amounts of data, it can automatically generate the desiblack output directly from the necessary inputs, without the need for expensive iterative algorithms. Once the training is complete, the DL model can make the pblackiction instantaneously. Bonnemain et al. [17] used deep learning to assess cardiac contractility. Also in another article [18], they developed a framework comprising a deep neural network and a 0D model of the cardiovascular system to pblackict parameters of LV systolic function. Ren [19] proposed a surrogate model based on the machine learning method to study cardiac mechanical parameters. In conclusion, deep learning can effectively speed up the calculation process of finite element simulation [20].

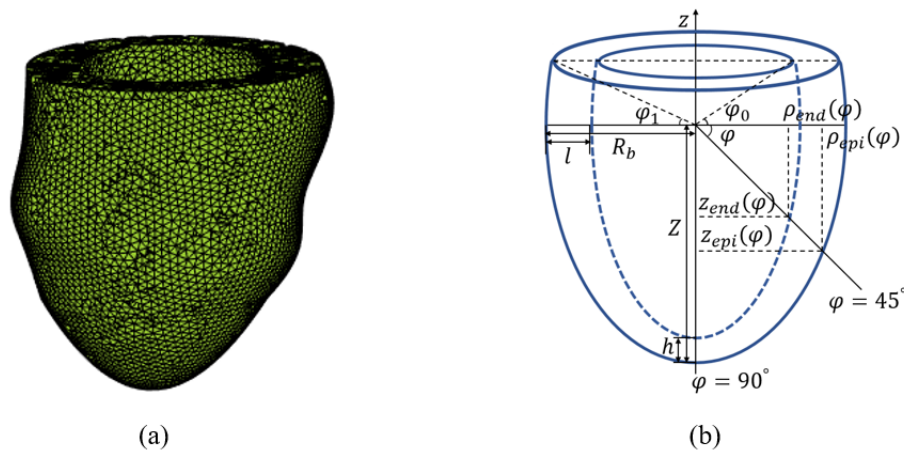
This paper presents a standardized geometric model of LV. Based on the previous research [19, 21], a PV curve parameters pblackiction model of the LV dynamics in diastole based on DL is proposed. Then the DL model is used for solving the inverse problem of finding the constitutive parameters.

## 2. Methods

### 2.1. Geometric model of LV

#### 2.1.1. Standardized model

In this study, we obtained 27 healthy LV geometries from a published study [22]. All geometries are reconstructed from in vivo cardiac MR images. Details of patient characteristics, imaging protocols and reconstruction procedures can be found in [22]. Figure 1(a) shows a reconstructed LV geometry of a healthy volunteer. The shape of the LV is irregular, and the thickness of the heart wall is not uniform. However, it is difficult to obtain a large number of MRI images of the LV. To facilitate follow-up studies, we approximate the LV with a geometric parametric model based on the previous model [3, 5, 23]. Figure 1(b) shows a schematic of an idealized LV geometry annotated with geometric descriptions of the parameters. The inner and outer surfaces of the left ventricle are respectively regarded as the surface of rotation formed by the curve rotating around the Z axis. Each point on the curve has two



**Figure 1.** The shape of LV segmented from the MRI image and geometric parameter model.

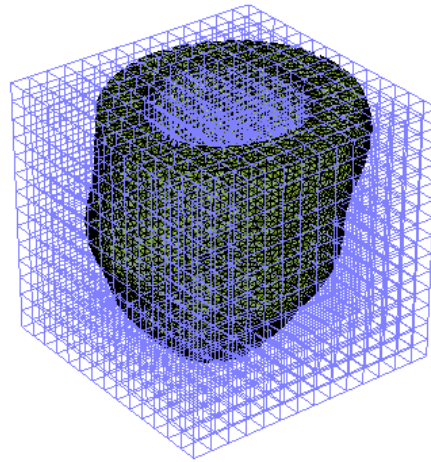
coordinates, the horizontal axis  $\rho$  and the vertical axis  $z$ . These two coordinates can be controlled by the Angle  $\varphi$ . The inner surface of  $[\varphi_0, \pi/2]$ , and outside surface of  $\varphi$  value range is  $[\varphi_1, \pi/2]$ . The parametric model was described by the following relations

$$\begin{cases} \rho_{epi} = R_b [e \cos \varphi + (1 - e)(1 - \sin \varphi)] \\ z_{epi} = Z(1 - \sin \varphi) \\ \rho_{end} = (R_b - l) [e \cos \varphi + (1 - e)(1 - \sin \varphi)] \\ z_{end}(\varphi) = (Z - h)(1 - \sin \varphi) \\ \varphi_1 = \arcsin\left(l - \frac{z_{end}(\varphi_0)}{Z}\right) \end{cases} \quad (2.1)$$

The parametric model is determined by six parameters: the maximum radius of the outer surface,  $R_b$ ; the wall thickness at the horizontal axis,  $l$ ; The distance between the origin and the lowest point of the outer surface,  $Z$ ; the wall thickness at the vertical axis,  $h$ ; the truncation angle of the inner wall,  $\varphi_0$ ; and finally, the degree of bending of the wall curve,  $e \in [0, 1]$ . When  $e = 0$ , the curve geometry becomes a cone; and when  $e = 1$ , it is an elliptic curve. In particular,  $\varphi_1$  can be calculated by  $\varphi_0$ . According to the MRI image data of the left ventricle, the initial geometric shape can be generated by setting initial values for the geometric model parameters.

### 2.1.2. Personalized geometric parameters

Based on the shape and coordinates of the original LV, the cube can be defined with  $(x_{min} - \xi, x_{max} + \xi)$ ,  $(y_{min} - \xi, y_{max} + \xi)$  and  $(z_{min} - \xi, z_{max} + \xi)$ , and the cube was evenly divided with a  $128 \times 128 \times 128$  grid. The calculation process will be slow if the mesh is too dense, and the result will not be accurate if the mesh step is too large.  $\xi$  is a constant to make the grid space slightly larger than the original LV, as shown in Figure 2. When the grid node is inside the shape of the LV wall, the node is assigned a value of 1; When the grid node is outside the shape of the LV wall, the node is assigned a value of 0. Then the original LV can be approximately represented by a three-dimensional matrix consisting of values 0 and 1, and the total number of nodes with the value of 1 in the matrix represents the shape of the left ventricle. Similarly, the idealized parametric model can also be approximated in the same cube space by a three-dimensional matrix composed of 0-1. Next, we optimize the parameter



**Figure 2.** A cube grid based on the left ventricle ( $10 \times 10 \times 10$  grids as an example).

$\varsigma = [R_b, Z, l, h, e, \varphi_0]$  of the geometric model. Establish the objective function  $G$ , as follows

$$G(I_{real}, I_{\varsigma}) = 1 - \frac{2(C_{real} \cap C_{\varsigma})}{|C_{real}| + |C_{\varsigma}| + \varepsilon} \quad (2.2)$$

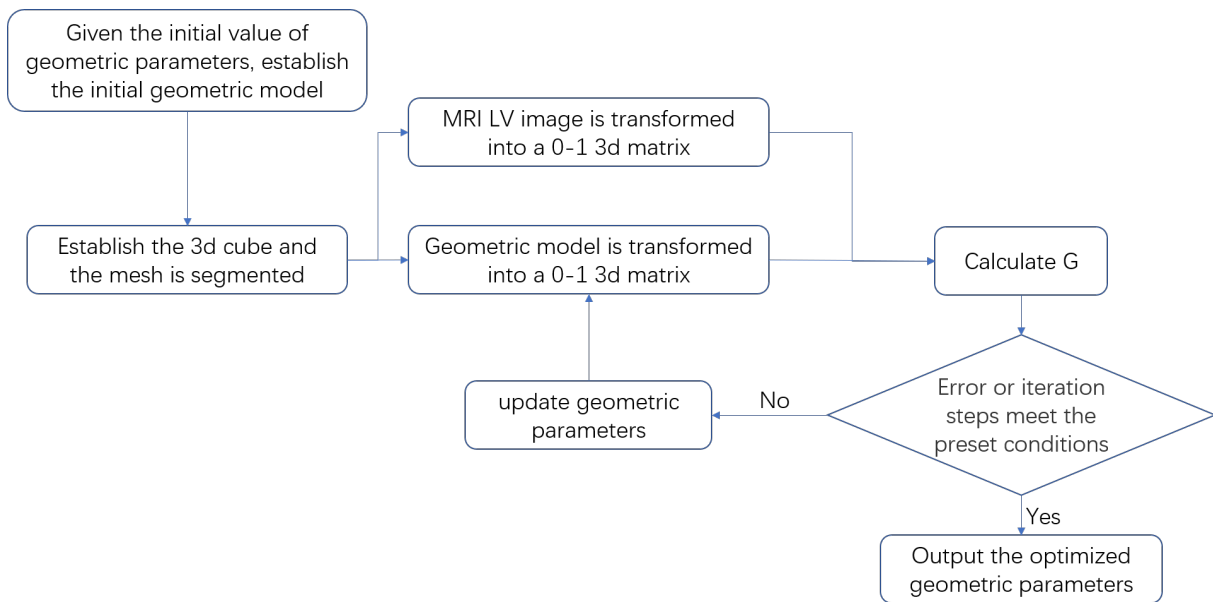
where  $I_{real}$  represents MRI LV image,  $I_{\varsigma}$  represents geometric model,  $C_{real}$  represents the 0-1 three-dimensional matrix of  $I_{real}$ ,  $C_{\varsigma}$  represents the 0-1 three-dimensional matrix of  $I_{\varsigma}$ .  $|C_{real}|$  represents the sum of all 1 in  $C_{real}$ . Similarly,  $|C_{\varsigma}|$  is the sum of all 1 in  $C_{\varsigma}$ .  $\varepsilon$  represents a very small constant in case the denominator is 0.

$G$  represents the measure of similarity between the geometric model and MRI image of the LV, and the value range is  $[0, 1]$ . The higher the degree of overlap between the two, the smaller  $G$  will be. The geometric parameter optimization model is established, and the mathematical expression is as follows

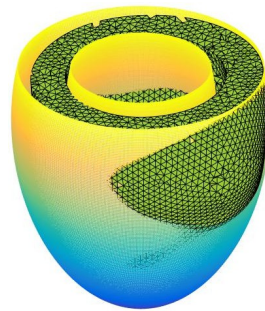
$$\hat{\varsigma} = \arg \min_{\varsigma} (G(I_{real}, I_{\varsigma}))$$

$$s.t. \begin{cases} 0 < R_b \leq R + \xi \\ 0 < l < R_b \\ 0 < Z \leq H + 2\xi - (R_b - l) \tan \varphi_0 \\ 0 < h < Z \\ 0 \leq e \leq 1 \\ 0 \leq \varphi_0 < \pi/2 \end{cases} \quad (2.3)$$

Among them,  $R$  is one-half of the widest value in the horizontal direction of the MRI LV image, and  $H$  is the longest value in the vertical direction of the MRI LV image. The optimization process uses the FMINcon function in MATLAB. The parameter optimization flow chart is shown in Figure 3. Figure 4 shows optimization results and the corresponding LV shape for a representative case.



**Figure 3.** Parameters optimization process.



**Figure 4.** 3-D view of the LV shape (represented as a green surface with a black grid) and best-fit idealized model (represented as a yellow and blue gradient surface) for a representative case.

## 2.2. Pressure-volume model

### 2.2.1. Constitutive law of LV

The constitutive relation, also known as the constitutive equation, is used to describe the relationship between force and deformation or stress and strain. The constitutive equation expresses the relationship between stress and strain as a function, and different materials have different constitutive relations. The LV myocardium is a non-uniform, thick-walled, incompressible, orthotropic nonlinear hyperelastic material. Myocardial structure strain energy functions are given below, the so-called H-O model [24, 25]

$$\Psi = \frac{a}{2b} [\exp \{b (I_1 - 3)\} - 1] + \sum_{i \in \{f, s\}} \frac{a_i}{2b_i} [\exp \{b_i (I_{4i} - 1)^2\} - 1] + \frac{a_{fs}}{2b_{fs}} \{ \exp (b_{fs} I_{8fs}^2) - 1 \} + \frac{1}{2} K (J - 1)^2 \quad (2.4)$$

in which  $a, b, a_f, b_f, a_s, b_s, a_{fs}, b_{fs}$  are unknown material parameters, The five parameters  $a, b, a_f, b_f, a_s$  are mainly used in this paper, and the remaining parameters are fixed with appropriate values based on the group's previous research [25].  $I_{4i}(i = f, s), I_{8fs}$  represent invariants along the fiber, slice and normal directions respectively. The calculation formulas are as follows

$$\begin{aligned} I_1 &= \text{tr}(\mathbb{C}) \\ I_{4f} &= \mathbf{f}_0 \cdot (\mathbb{C}\mathbf{f}_0) \\ I_{4s} &= \mathbf{s}_0 \cdot (\mathbb{C}\mathbf{s}_0) \\ I_{8fs} &= \mathbf{f}_0 \cdot (\mathbb{C}\mathbf{s}_0) \end{aligned} \quad (2.5)$$

in which  $\mathbf{f}_0$  and  $\mathbf{s}_0$  are the myofibre and sheet orientations, which are determined by a rule-based method [25] and known before numerical simulation (initial conditions).  $\mathbb{C}$  is the right Cauchy-Green deformation tensor, defined as  $\mathbb{C} = \mathbb{F}^T\mathbb{F}$ , and  $\mathbb{F}$  is the deformation gradient that describes the motion of the heart muscle. In other words, the deformation gradient describes how the shape of the heart muscle changes over time in three dimensions. The expression  $\frac{1}{2}K(J-1)^2$  represents the incompressibility of the material,  $K$  is a constant, and  $J$  is the determinant of  $\mathbb{F}$ .

However, Gao [21] had found that the sensitivity of these parameters for the changes of volume and PV curves is not quite the same. For example,  $a, b$ , and  $a_f$  had the highest sensitivity, while  $a_{fs}, b_{fs}$ , and  $b_s$  have the lowest sensitivity. Therefore, in the following, we selected three groups of parameter combinations for analysis and comparison, which are  $\{a, b, a_f, b_f\}$ ,  $\{a, b, a_f, a_s\}$  and  $\{a, b, a_f\}$ , respectively.  $a_s$  and  $b_f$  were added respectively in the first two combinations to serve as a comparison with the third group of parameters.

### 2.2.2. Numerical simulation of LV

The deformation process of the LV diastole can be simulated by numerical simulation. The specific calculation model can be described as the boundary value question below [26–28].

$$\begin{cases} \nabla \cdot \boldsymbol{\sigma} + \mathbf{b} = 0 & \text{in } \Omega \\ \boldsymbol{\sigma} \cdot \mathbf{n} = \mathbf{t} & \text{in } \Gamma^N \\ \mathbf{u} = \mathbf{u}_0 & \text{in } \Gamma^D \end{cases} \quad (2.6)$$

where  $\Omega$  represents the entire LV wall region,  $\partial\Omega$  represents the boundary of  $\Omega$ , the relation between Cauchy stress tensor and Piola Kirchhoff stress of the second kind is  $\boldsymbol{\sigma} = \frac{1}{J}\mathbf{F}\mathbf{S}\mathbf{F}^T$  (which can be derived from Formula 2.4),  $\mathbf{b}$  represents the surface force of unit volume,  $\mathbf{n}$  represents the external normal of the unit of  $\partial\Omega$ , and  $\Gamma^N$  and  $\Gamma^D$  represent the displacement boundary and pressure boundary. The displacement boundary refers to the displacement control of the plane, circumferential direction and vertical axis at the bottom of the left ventricle is 0, and the radial direction can contract or expand, while other regions are free to deform. Meanwhile, the non-sliding boundary condition was applied to the LV wall [9]. The pressure boundary condition refers to that the pressure is linearly loaded on the LV wall in 25-time steps with a pressure range from 0 to 8mmHg. The pressure of the outer wall of the left heart was set at 0mmHg. In the beginning, the left ventricle was regarded as the resting state and the end as the end-diastolic state. The hexahedron element is used to discretize space and solve it by the finite element method. The simulation process was realized by ABAQUS (Simulia, Providence, RI, USA). The volume of the corresponding LV lumen was extracted after each step of the compression process.

### 3. Generation of data sets

#### 3.1. Parameter range

27 sets of geometric parameters were obtained from the Section 2.1. Supposing that each geometric parameter follows a normal distribution  $N(\mu, \sigma^2)$ . The 27 known geometric parameters are samples of the population. Since the sample variance  $S_n^{*2}$  is the least variance unbiased estimate of  $\sigma^2$ ,  $\sigma$  can be approximated by  $S_n^*$ . Then we do an interval estimate of the population mean  $\mu$ . On the assumption that  $(X_1, X_2, \dots, X_n)^T$  is a sample from population  $\bar{X}$ , and we can use  $\bar{X}$  as a point estimate for  $\mu$ . And the sample means  $\bar{X}$  is the moment estimator of the population mean  $\mu$ , so  $\bar{X}$  can be used to approximate  $\mu$ . Therefore, the sample population score of geometric parameters can be obtained, as shown in Table 1. According to the  $3\sigma$  principle of normal distribution, the probability of sample

**Table 1.** Geometric parameters distribution.

Parameter	$e$	$R_b(mm)$	$Z(mm)$	$l(mm)$	$h(mm)$	$\varphi_0$
$\sigma$	0.0008	2.1072	5.9329	0.8454	0.6346	0.0032
$\mu$	0.9021	30.7802	62.2350	12.7331	6.9967	-0.5299
$\mu+3\sigma$	0.9046	37.1017	80.0337	15.2693	8.9007	-0.5202
$\mu-3\sigma$	0.8995	24.4587	44.4363	10.1968	5.0928	-0.5396
<i>Max</i>	0.9038	35.2236	69.1697	14.5221	8.4652	-0.5236
<i>Min</i>	0.9000	26.1913	46.3952	10.9064	6.0210	-0.5369

falling within  $[\mu - 3\sigma, \mu+3\sigma]$  is 99.7%, so  $[\mu - 3\sigma, \mu+3\sigma]$  is taken as the value range of geometric parameter. The range of the constitutive parameters has been obtained from previous studies [22], as shown in Table 2.

**Table 2.** constitutive parameters distribution.

Parameter	$a(kPa)$	$b$	$a_f(kPa)$	$a_s(kPa)$	$b_f$
<i>Max</i>	10	30	30	10	30
<i>Min</i>	0.05	0.1	0.1	0.05	0.1

#### 3.2. Sampling the parameter space

The multi-dimensional sample space is established according to the result of parameter estimation. According to different combinations of constitutive parameters, three sampling Spaces are established as follows

$$\begin{aligned}
 A_1 &= \{R_b, Z, l, H, \varphi_0, e, a, b, a_f, b_f\} \\
 A_2 &= \{R_b, Z, l, H, \varphi_0, e, a, b, a_f, a_s\} \\
 A_3 &= \{R_b, Z, l, H, \varphi_0, e, a, b, a_f\}
 \end{aligned} \tag{3.1}$$

each sampling space is composed of geometric parameters and constitutive parameters. Due to the high dimension of our sampling space, to ensure the efficiency of sampling and the uniformity of sample distribution, LHS method [29] is selected here. LHS is a multi-dimensional stratified sampling method. Its advantage is that the sampling value can fully reflect the overall distribution of the sample

and ensure that all probability intervals are covered by sampling points. The sampling process is as follows:

- 1) Let the dimension of sampling space be  $E$ , and define the number of samples to be sampled as  $N$ .
- 2) Divide each one-dimensional sampling region into  $N$  small regions with equal probability, then generate  $E$  group  $N$  small region set. Some small region in the group that it's in has a probability of  $1/N$ .
- 3) A point in a small region is randomly selected in each dimension. The  $E$  points are combined to form a sample, and the small region selected will not participate in the subsequent sampling.
- 4) Repeat the third step  $N$  times, all the small areas are just used up, and  $N$  samples are obtained.

10,000 samples were collected in each sampling space, with a total of 30,000 samples. Because the sampling of each parameter is randomly combined, there will inevitably be samples that do not conform to the real physiological situation. Such samples will lead to non-closure of the virtual LV bottom, calculation failure or compression failure in the subsequent ABAQUS simulation. These data will be eliminated.

### 3.3. Model of PV curve parameters

The ventricular PV curve represents the relationship between pressure and volume changes in the LV cavity. True cardiac PV relationships can be obtained using catheters and ultrasound monitoring.

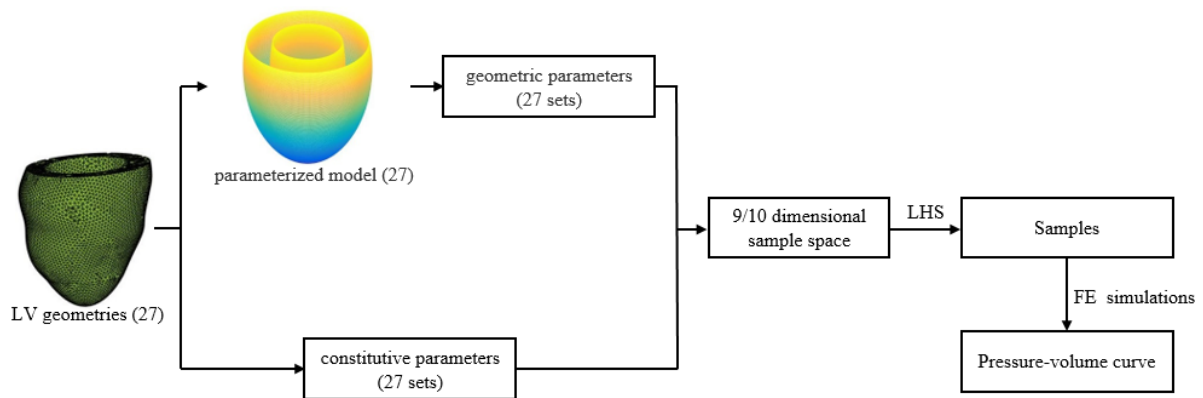
Three sample sets  $A_1$ ,  $A_2$  and  $A_3$  with 30,000 groups of material parameters samples obtained above were simulated using the method in Section 2.2.2. Then 30,000 PV relationship curves were generated. The configuration used for finite element calculation in this paper is a workstation with a 64-bit Windows 7 operating system, using the Intel Core E52609 processor and 32GB memory space. The simulation process of ABAQUS uses 16-thread parallel. Some parameter samples may have problems in the calculation process, as mentioned in Section 3.2. After eliminating through wrong data, 6985, 7742 and 7782 sets of PV curves were left. The process of calculating the relationship between load pressure and volume is shown in Figure 5.

Klotz et al. [30] found that the relationship between normalized volume ( $v_n$ ) and  $p$  can be approximated by  $p = \alpha v_n^\beta$ , where  $\alpha$  and  $\beta$  are coefficients,  $v_n = \frac{v-v_0}{v_0}$  is the normalized volume with respect to the initial volume  $v_0$ . We also use the quadratic polynomial function  $p = dv^2 + qv + c$  approximates the relationship between  $v$  and  $p$ . After the least square fitting processing, the obtained effect is shown in Figure 6. Finally, we get the values of parameters  $\alpha$ ,  $\beta$ ,  $d$ ,  $q$  and  $c$  corresponding to the three sample sets. The distribution of three-parameter PV curve parameters  $\{d, q, c\}$  and two-parameter PV curve parameters  $\{\alpha, \beta\}$  is shown in Figure 7.

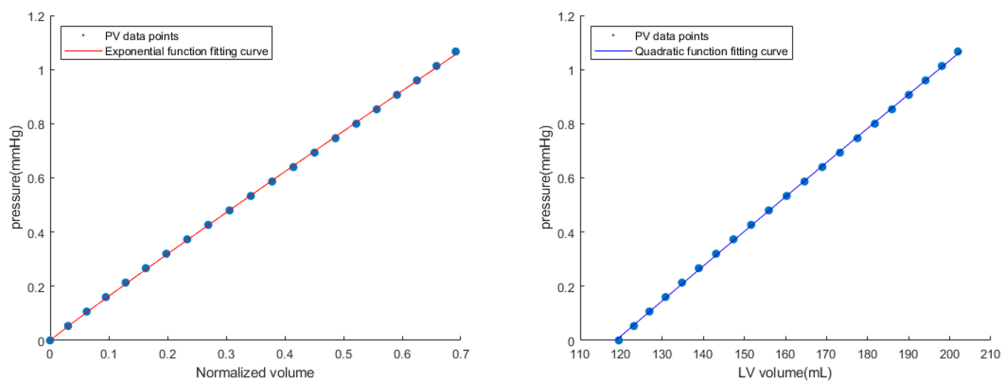
### 3.4. Partitioning the dataset

10% data are randomly selected from the entire dataset as the test set, and the remaining data are training and validation set. As a result, in  $A_1$ , the training and validation set has 6286 input-output pairs, and the test set has 699 input-output pairs; In  $A_2$ , the training and validation set has 6967 input-output pairs, and the test set has 775 input-output pairs and in  $A_3$ , the training and validation set has 7003 input-output pairs, the test set has 779 input-output pairs.

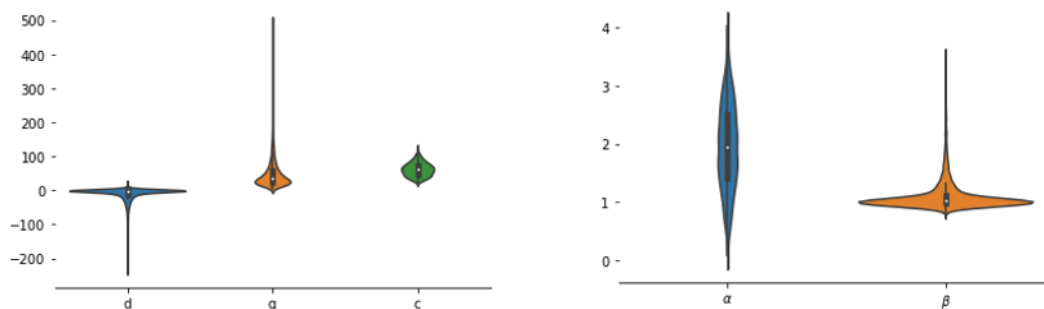




**Figure 5.** The procedure to generate the relationship between pressure and volume.



**Figure 6.** PV relationship changes at the end of diastole.



**Figure 7.** Parameters distribution of output.

## 4. Deep learning model for the PV curve parameters

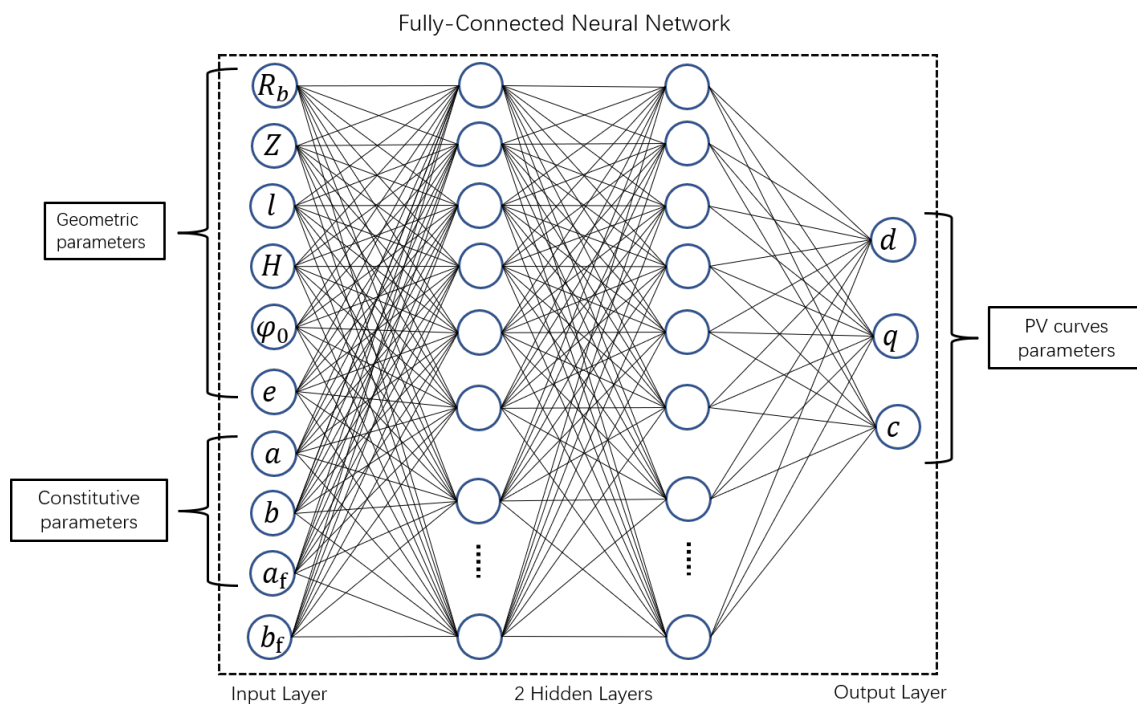
### 4.1. The neural network model

The nonlinear mapping system will map the geometric parameters, constitutive parameters to the PV curve parameters. 6 neural networks  $NN_i (i = 1, \dots, 6)$  are established. The corresponding input  $x$

and output  $\mathbf{y}$  of each neural network are as offblack

$$\begin{aligned}
 NN_1 : \mathbf{x} &= (R_b, Z, l, h, \varphi_0, e, a, b, a_f, b_f), \mathbf{y} = (d_1, q_1, c_1) \\
 NN_2 : \mathbf{x} &= (R_b, Z, l, h, \varphi_0, e, a, b, a_f, a_s), \mathbf{y} = (d_2, q_2, c_2) \\
 NN_3 : \mathbf{x} &= (R_b, Z, l, h, \varphi_0, e, a, b, a_f), \mathbf{y} = (d_3, q_3, c_3) \\
 NN_4 : \mathbf{x} &= (R_b, Z, l, h, \varphi_0, e, a, b, a_f, b_f), \mathbf{y} = (\alpha_1, \beta_1) \\
 NN_5 : \mathbf{x} &= (R_b, Z, l, h, \varphi_0, e, a, b, a_f, a_s), \mathbf{y} = (\alpha_2, \beta_2) \\
 NN_6 : \mathbf{x} &= (R_b, Z, l, h, \varphi_0, e, a, b, a_f), \mathbf{y} = (\alpha_3, \beta_3)
 \end{aligned} \tag{4.1}$$

As shown in Figure 8, this is a feedforward fully connected neural network. Each neuron has



**Figure 8.** the neural network for mapping the geometric parameters & PV curves to the constitutive parameters.

multiple inputs and one output. For example, take the forward propagation of one neuron, each neuron has a weight  $\omega_{ji}^{(l)}$  as it passes down the next layer. Namely, the weight of the connection between neuron  $i$  at layer  $L - 1$  and neuron  $j$  at layer  $l$ . The output of neurons in layer  $l - 1$  to layer  $l$  is

$$z_i^{(l)} = f^{(l)} \left( \sum_i \omega_{ji}^{(l)} z_i^{(l-1)} + b_j^{(l)} \right) \tag{4.2}$$

where  $b_j^{(l)}$  is the offset term and  $f^{(l)}$  represents the activation. If the offset term is represented as the column vector  $\mathbf{b}^{(l)}$ , the Eq (4.2) can be expressed as

$$\mathbf{z}^{(l)} = f^{(l)} \left( \mathbf{W}^{(l)} \mathbf{z}^{(l-1)} + \mathbf{b}^{(l)} \right) \tag{4.3}$$

However, offset terms can also be represented in weights. Add a neuron with output 1 on the input layer and all hidden layers, denoted as  $i = 0$ , then

$$z_0^{(l)} = 1 \quad (4.4)$$

The weight  $\omega_{j0}^{(l)}$  is introduced between the neuron  $i = 0$  at layer  $l - 1$  and the neuron  $j \neq 0$  at layer  $l$ , then the weight is the bias term. The derivation process is

$$\sum_{i=0} \omega_{ji}^{(l)} z_i^{(l-1)} = \sum_{i=1} \omega_{ji}^{(l)} z_i^{(l-1)} + \omega_{j0}^{(l)} = \sum_{i=1} \omega_{ji}^{(l)} z_i^{(l-1)} + b_j^{(l)} \quad (4.5)$$

the Eq (4.3) can be further expressed as

$$\mathbf{z}^{(l)} = f^{(l)}(\mathbf{W}^{(l)} \mathbf{z}^{(l-1)}) \quad (4.6)$$

Assume that a neural network consists of  $L$  layers, and the model from input  $\mathbf{x}$  to output  $\mathbf{y}$  is

$$\widehat{\mathbf{y}} = f^{(L)}(\mathbf{W}^{(L)} f^{(L-1)}(\mathbf{W}^{(L-1)} f^{(L-2)}(\dots \mathbf{W}^{(2)} f^{(1)}(\mathbf{x}))) \quad (4.7)$$

$\widehat{\mathbf{y}}$  represents the approximation of  $\mathbf{y}$ . Using this functional model to fit the data is the mathematical expression of a neural network. And then the training is to find the optimal  $\mathbf{W}^{(L)}$ .

The neural network has many activation functions to choose from, we choose ReLU [31] as our activation function, which is given by

$$f_{ReLU}(x) = \max(0, x) \quad (4.8)$$

The neural network model has two hidden layers, take the example of  $NN_1$  in formula 4.1, the input layer has ten neurons, and the output layer has three neurons. The number of neurons in each layer of the neural network will be determined in subsequent training. Keras is used to implement the neural network.

Before training, the input data and output data are normalized. Here, we use the min-max normalization

$$x^* = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (4.9)$$

## 4.2. Training and testing

### 4.2.1. 10-fold cross validation

Using the training/validation data set, the performance of the nonlinear mapping module is evaluated by cross-validation. The hyperparameters of the neural network are adjusted. The data set is divided into  $N$  parts, one of which is taken as the verification set each time. The other  $N - 1$  parts are put together as the training set. A total of  $N$  training sessions are conducted, and the average result is selected as the final training result. In this case,  $N$  is equal to 10.

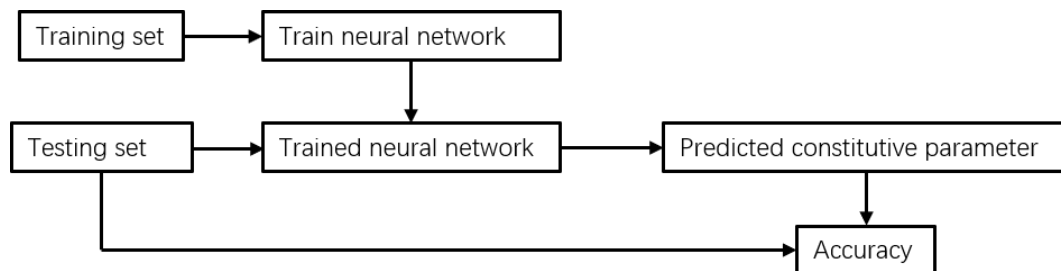
#### 4.2.2. Search for the optimal number of network nodes

Since the dimension of the input layer and the output layer of our neural network is not large, the stepwise increase method [32] is adopted to find the optimal network node number of hidden layers. The method starts from the network with a small number of nodes, and gradually increases the number of hidden nodes according to the speed change of blackucing the network output error. Firstly, the number of nodes on the second hidden layer is fixed at 30. The number of nodes on the first hidden layer gradually increases from 1 to 30. Similarly, fix the first hidden layer to the optimal value just found and increase the number of nodes in the second hidden layer from 1 to 30. It can be found that when the error drops to a certain value, it does not decrease with the increase of nodes, and the optimal number of nodes is obtained. The optimal number of nodes of each neural network is shown in Table 3.

**Table 3.** The number of hidden layer nodes of each neural network.

datasets	$y_1$	$y_2$	$y_3$	$y_4$	$y_5$	$y_6$
hidden layer 1	19	17	18	14	14	12
hidden layer 2	17	14	12	6	13	12

To evaluate the deep learning model and check the accuracy of the pblackicted results, the PV curve's parameters are pblackicted using the data in the test set. The pblackicted parameters are com-pblack with the parameters in the test set, which process is illustrated in Figure 9.



**Figure 9.** Evaluating the accuracy using the testing dataset.

#### 4.2.3. Results

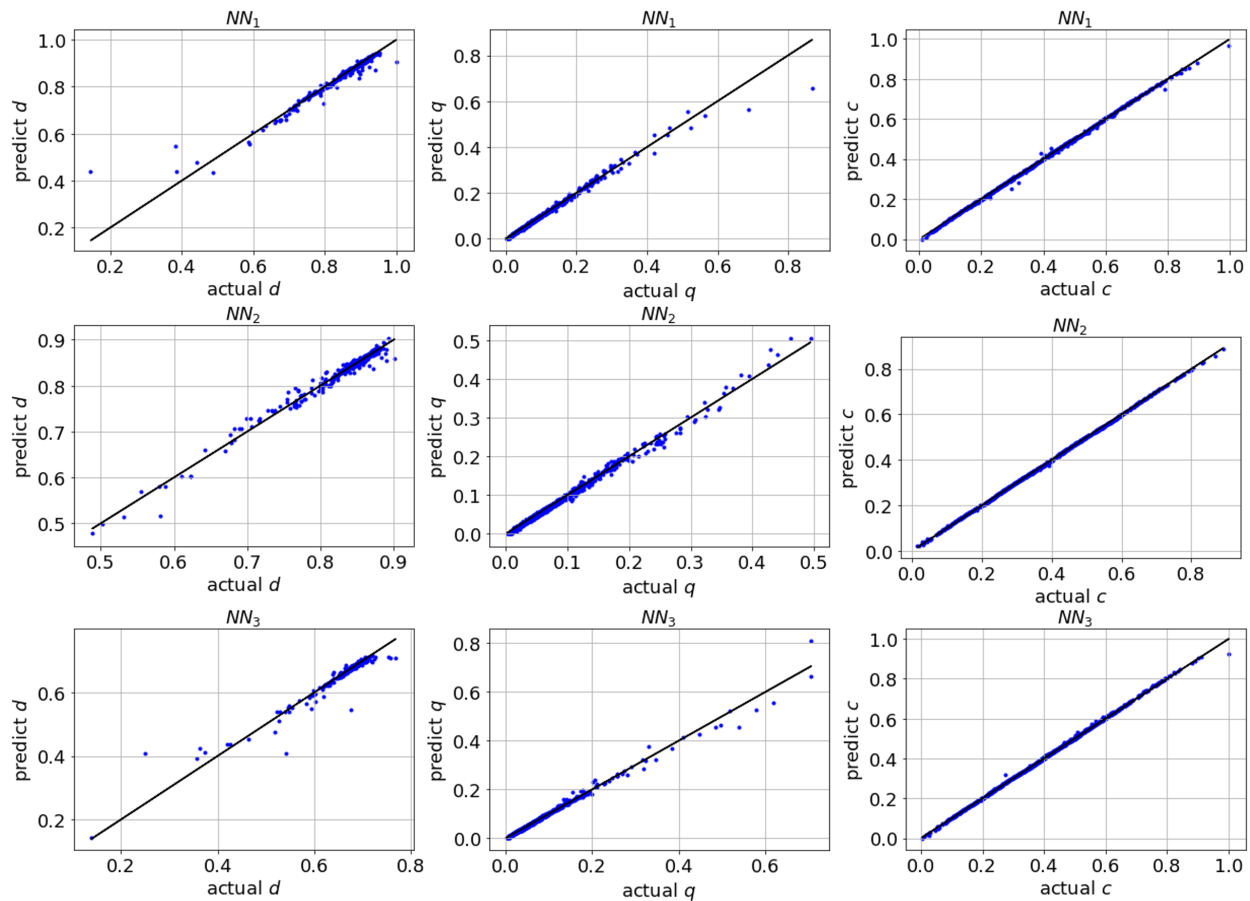
Once the network structure is determined, data can be put in for training. The mean absolute error (MAE) is used as the loss function, The training error is shown in Table 4. It can be found that the

**Table 4.** Training error (MAE).

datasets	$NN_1$	$NN_2$	$NN_3$	$NN_4$	$NN_5$	$NN_6$
Training error	0.0052	0.0048	0.0026	0.0047	0.0051	0.0051
Validation error	0.0053	0.0042	0.0032	0.0042	0.0061	0.0052

training error results of the two pressure-volume curve parameters are similar, and both have good performance. The gradient descent algorithm used in this paper is Adam optimization algorithm [33]. After training, proper weights, bias and hyperparameters are obtained for the network. Then we plug

the data from the test set into the neural network. The actual PV curve parameters versus pblackicted PV curve parameters are shown in Figures 10 and 11. The horizontal axis represents the true value of



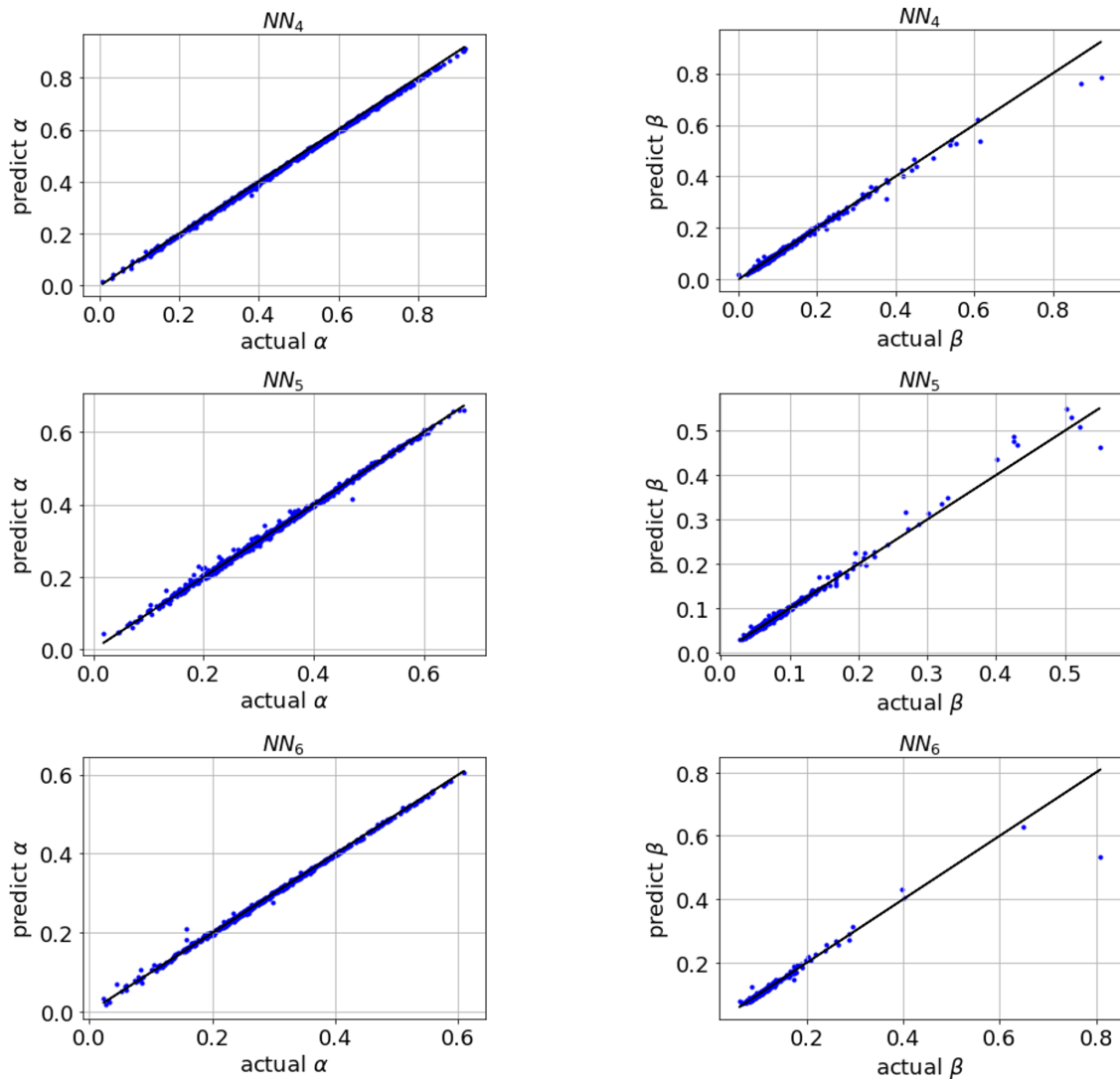
**Figure 10.** The actual and pblackicted quadratic polynomial PV parameters.

the parameter, and the vertical axis represents the pblackicted value of the parameter. The closer to the line  $y = x$  in the figure, the more accurate the results are. It can be seen that among the three parameters of the PV curve fitted by the quadratic function, the pblackiction result of  $c$  is the best, followed by  $q$ , while the pblackicted value of some points in  $d$  is slightly far from the real value. Among the two parameters of the PV curve fitted by the exponential function,  $\alpha$  pblackicted better.

The discrepancy between the actual material parameters and pblackicted material parameters in the testing set is quantified by normalized mean absolute error (NMAE) [20]. The absolute error (AE) for the  $k^{\text{th}}$  PV parameter is defined by

$$AE_k^{(l)} = |y_k^{(l)} - \widehat{y}_k^{(l)}| \quad (4.10)$$

$y_k^{(l)}$  and  $\widehat{y}_k^{(l)}$  represent the  $k^{\text{th}}$  actual and pblackicted PV parameter, respectively. The NMAE of the  $k^{\text{th}}$



**Figure 11.** The actual and predicted exponential function PV parameters.

PV parameter is defined by

$$NMAE_k = \frac{\sum_{l=1}^n AE_k^{(l)}}{n \left( \max_l (y_k^{(l)}) - \min_l (y_k^{(l)}) \right)} \times 100\% \quad (4.11)$$

where  $n$  represents the total number of training data. The NMAE for each PV parameter in the testing sets are reported in Table 5. It can be found that  $NN_1$  and  $NN_4$  have the best prediction results, and their errors are lower than 1%. The error of all parameters is within 5%. The results show that the PV parameters predicted by deep learning are in good agreement with the actual PV parameters. The results of the quadratic function PV curve and exponential function PV curve are similar, indicating that within a certain range, the variation relationship of PV can also be approximated by the quadratic function.

**Table 5.** NMAE of the pblackicted PV parameters in testing sets.

		$d$	$q$	$c$	$\alpha$	$\beta$
$NN_1$	$NMAE_k$	0.72%	0.46%	0.70%	-	-
$NN_2$	$NMAE_k$	1.99%	1.36%	4.31%	-	-
$NN_3$	$NMAE_k$	1.26%	1.42%	0.55%	-	-
$NN_4$	$NMAE_k$	-	-	-	0.41%	0.51%
$NN_5$	$NMAE_k$	-	-	-	0.83%	0.72%
$NN_6$	$NMAE_k$	-	-	-	0.70%	0.63%

Then  $R^2$  is used to evaluate the pblackiction accuracy, as defined by the following formula

$$R^2 = \frac{\sum_{i=1}^n (\hat{y}_i - \bar{y})^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (4.12)$$

The value of  $R^2$  ranges from 0 to 1, and the closer to 1, the more accurate it is. The accuracy of network pblackiction is shown in Table 6. The observation results show that all parameters are highly

**Table 6.** Parameters pblackiction accuracy ( $R^2$ ).

	$d$	$q$	$c$	$\alpha$	$\beta$
$NN_1$	0.9289	0.9484	0.9945	-	-
$NN_3$	0.9277	0.9831	0.9915	-	-
$NN_4$	-	-	-	0.9908	0.9168
$NN_5$	-	-	-	0.9648	0.9638
$NN_6$	-	-	-	0.9879	0.8744

accurate. It shows that deep learning can replace the calculation process of the finite element method and pblackict relevant parameters. The accuracy of fitting PV parameters with quadratic function is above 0.9, showing good performance.

## 5. Deep learning model for the constitutive parameters

### 5.1. The neural network model

The calculation of LV constitutive parameters is an inverse problem, which is usually solved by solving an optimization problem with additional conditions [19]. Different from the previous methods, we establish the nonlinear mapping relationship between input features and output features by taking geometric parameters and three-parameter PV curves as inputs and constitutive parameters as outputs. Three multiple neural networks are established. The input and output characteristics of each network are as follows

$$\begin{aligned}
 NN_7 : \mathbf{x} &= (R_b, Z, l, h, \varphi_0, e, d_1, q_1, c_1), \mathbf{y} = (a, b, a_f, a_s) \\
 NN_8 : \mathbf{x} &= (R_b, Z, l, h, \varphi_0, e, d_2, q_2, c_2), \mathbf{y} = (a, b, a_f, b_f) \\
 NN_9 : \mathbf{x} &= (R_b, Z, l, h, \varphi_0, e, d_3, q_3, c_3), \mathbf{y} = (a, b, a_f)
 \end{aligned} \quad (5.1)$$

Each network selects a slightly different combination of output characteristics. According to Gao's study [22], the constitutive parameters  $a$ ,  $b$  and  $a_f$  are sensitive, and the PV change has a great influence

on them. Meanwhile, parameters  $a_s$  and  $b_f$  with lower sensitivity are added to the output characteristics of  $NN_7$  and  $NN_8$  respectively as controls. The other parameters are fixed as in Section 4.

LHS method and deep learning model in Section 4 are used to expand data, and 50,000 sets of data are generated for  $NN_7$ ,  $NN_8$  and  $NN_9$  respectively, with a total of 150,000 sets of input and output pairs. Next comes data processing, initialization, and selection of hyperparameters.

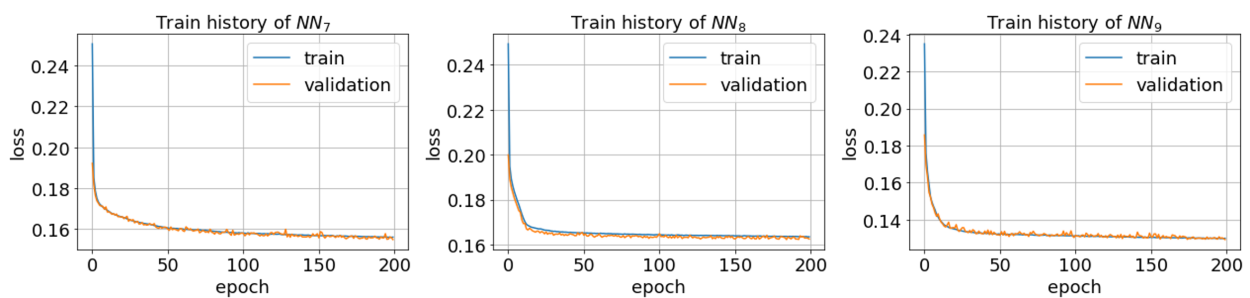
The ReLU activation function and the MAE loss function are still used here. The number of nodes in each hidden layer is determined by the stepwise increase method, the nodes of each neural network are shown in Table 7.

**Table 7.** The number of hidden layer nodes of each neural network.

datasets	$y_7$	$y_8$	$y_9$
hidden layer 1	15	13	15
hidden layer 2	9	9	12

## 5.2. Training and testing

10% of each dataset was randomly selected as the test set, and the remaining data were used for training. As a result, in each neural network, the training and validation set has 45,000 input-output pairs and the test set has 5000 input-output pairs. The training process uses 10-fold cross-validation to ensure accuracy. The loss function decline curve of the training set and verification set of the three neural networks in the learning process is shown in Figure 12. It can be found that the loss function curve of the inverse problem shows a downward trend in general. But the local loss function of the verification set always has a small fluctuation. The MAE training error of each neural network is shown in Table 8. It can be found that the error of  $NN_9$  is the smallest, and the error of  $NN_7$  and  $NN_8$  is slightly larger, because the constitutive parameters  $a_s$  and  $b_f$  are added into the output characteristics.



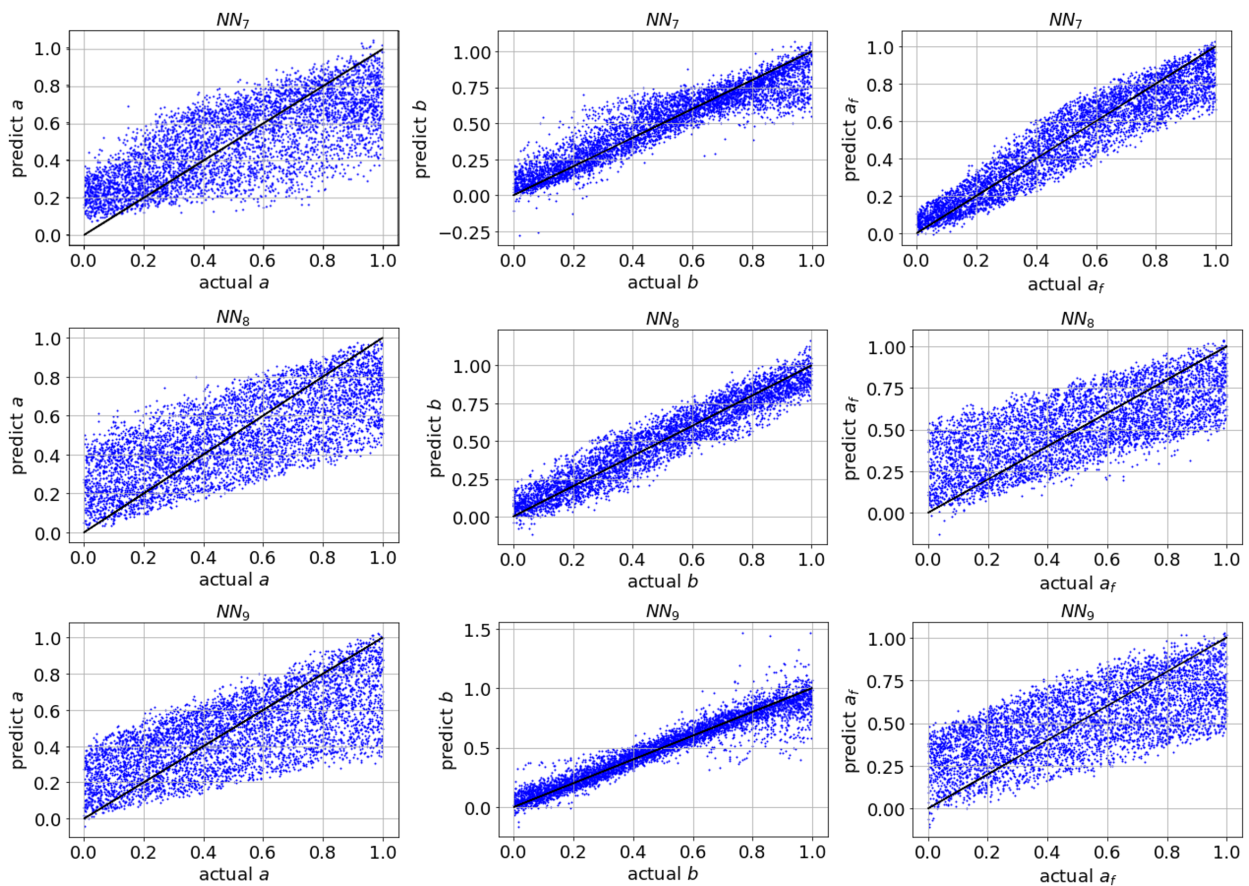
**Figure 12.** Loss function decline curve.

**Table 8.** Training error (MAE).

datasets	$NN_7$	$NN_8$	$NN_9$
Training error (mae)	0.1529	0.1617	0.1273
Validation error (mae)	0.1524	0.1605	0.1274

Next, test sets are used to test the trained neural network, The comparison between the actual value and the predicted value is illustrated in Figure 13. It can be found that among the three groups of





**Figure 13.** The actual and predicted quadratic polynomial constitutive parameters.

predicted constitutive parameters,  $b$  has the best performance, and most points are close to straight line  $y = x$ . While the distribution of parameters  $a$  and  $a_f$  is slightly loose.  $a_s$  in network  $NN_7$  and  $b_f$  in network  $NN_8$  are not sensitive to PV changes, so they cannot be well predicted, and the error of the whole network increases. The prediction accuracy of each network is measured by  $R^2$ , as shown in Table 9. According to the results in the table, the prediction accuracy of  $a$ ,  $b$  and  $a_f$  with high

**Table 9.** Prediction accuracy of constitutive parameters ( $R^2$ ).

	$a$	$b$	$a_f$
$NN_7$	0.7021	0.9013	0.8483
$NN_8$	0.7558	0.8109	0.8336
$NN_9$	0.7603	0.9434	0.8534

sensitivity is between 0.7 and 0.9. The result accuracy of 9 is higher than the other two networks, which shows the importance of feature selection. Compared with the prediction accuracy of positive questions in the previous chapter, the accuracy of negative questions decreases obviously, which is related to the unsuitability of negative questions. Estimate the constitutive parameters by a normal method generally need dozens of hours, deep learning can significantly speed up the results of the output. But the parameter estimation accuracy will be reduced, so the method is more suitable for

lack of experimental data in the inverse problem generated when a large number of data and combined with the finite element calculation, determine the initial value range of parameter iteration to improve iteration efficiency.

## 6. Discussion and conclusions

There are some limitations to this paper. 27 real heart data were used in the study, but the expected results were not obtained when the trained model was used to pblackict the parameters of these hearts. That's probably because the characteristic parameters selected in the learning of the inverse problem were insufficient or the data volume was not large enough. Later researchers will try different schemes and continue to improve the model for this problem.

Some other aspects in this paper need to be improved in subsequent research. First, the data used in the test set were generated by statistical methods, not directly from in vivo measurements of the LV. Later, more clinical data can be obtained to verify the accuracy of the model with data independent of the training set. Secondly, patient specificity needs to be considereblack. The geometric characteristics of the LV and myocardial parameters of some patients are not within the normal value range, so the model should detect and indicate abnormal points instead of making direct pblackictions. Third, in numerical simulation, the initial value of pressure on the LV is 0, while the real heart is always under pressure at any time. Future work will need to investigate how existing data can be used to model a more realistic left ventricular state.

## Acknowledgments

This work is supported by National Natural Science Foundation of China (11871399) and the UK EPSRC (EP/S030875, EP/S014284/1, EP/S020950/1, EP/R511705/1, EP/T017899/1), which are gratefully acknowledged.

## Conflict of interest

The authors declare there is no conflict of interest.

## References

1. H. Gao, K. Mangion, D. Carrick, D. Husmeier, X. Y. Luo, C. Berry, Estimating prognosis in patients with acute myocardial infarction using personalized computational heart models, *Sci. Rep.*, **7** (2017), 13527. <https://doi.org/10.5465/AMBPP.2017.13527abstract>
2. K. Mangion, H. Gao, D. Husmeier, X. Y. Luo, C. Berry, Advances in computational modelling for personalised medicine after myocardial infarction, *Heart (British Cardiac Society)*, **104** (2018), 550–557. <https://doi.org/10.1136/heartjnl-2017-311449>
3. D. D. Streeter, W. T. Hanna, Engineering mechanics for successive states in canine left ventricular myocardium, *Circ. Res.*, **33** (1973), 639–55. <https://doi.org/10.1161/01.RES.33.6.639>

4. S. F. Pravdin, V. I. Berdyshev, A. V. Panfilov, L. B. Katsnelson, V. S. Markhasin, Mathematical model of the anatomy and fibre orientation field of the left ventricle of the heart, *Biomed. Eng. Online*, **12** (2013), 54. <https://doi.org/10.1186/1475-925X-12-54>
5. P. D. Achille, A. Harouni, S. Khamzin, O. Solovyova, J. J. Rice, Gaussian process regressions for inverse problems and parameter searches in models of ventricular mechanics, *Front. Physiol.*, **9** (2018), 1002. <https://doi.org/10.3389/fphys.2018.01002>
6. P. Gould, D. Ghista, L. Brombolich, I. Mirsky, In vivo stresses in the human left ventricular wall: Analysis accounting for the irregular 3-dimensional geometry and comparison with idealised geometry analyses, *J. Biomech.*, **5** (1972), 521–539. [https://doi.org/10.1016/0021-9290\(72\)90009-7](https://doi.org/10.1016/0021-9290(72)90009-7)
7. M. Sermesant, P. Moireau, O. Camara, J. S. Marie, R. Andriantsimiavona, R. Cimrman, et al., Cardiac function estimation from MRI using a heart model and data assimilation: advances and difficulties, *Med. Image Anal.*, **10** (2006), 642–656. <https://doi.org/10.1016/j.media.2006.04.002>
8. C. S. Peskin, Flow patterns around heart valves: A numerical method, *J. Comput. Phys.*, **10** (1972), 252–271. [https://doi.org/10.1016/0021-9991\(72\)90065-4](https://doi.org/10.1016/0021-9991(72)90065-4)
9. L. Cai, H. Gao, X. Y. Luo, Y. F. Nie, Multi-scale modelling of the human left ventricle, *Sci. Sin.*, **45** (2015), 024702. <https://doi.org/10.1360/SSPMA2013-00100>
10. B. E. Griffith, X. Y. Luo, Hybrid finite difference/finite element immersed boundary method, *Int. J. Numer. Methods Biomed. Eng.*, **33** (2017), 101002. <https://doi.org/10.1002/cnm.2888>
11. H. Gao, H. M. Wang, C. Berry, X. Y. Luo, B. E. Griffith, Quasi-static imaged-based immersed boundary-finite element model of human left ventricle in diastole, *John Wiley Sons. Ltd.*, **30** (2014), 1199–1222. <https://doi.org/10.1002/cnm.2652>
12. L. Cai, H. Gao, X. Y. Luo, Y. H. Wang, Y. Q. Li, A mathematical model for active contraction in healthy and failing myocytes and left ventricles, *Plos One*, **12** (2017), e0174834. <https://doi.org/10.1371/journal.pone.0174834>
13. L. C. Yann, B. Yoshua, H. Geoffrey, Deep learning, *Nature*, **521** (2015), 436–44. <https://doi.org/10.1038/nature14539>
14. G. Litjens, T. Kooi, B. E. Bejnordi, A. Setio, C. I. Sánchez, A survey on deep learning in medical image analysis, *Med. Image Anal.*, **42** (2017), 60–88. <https://doi.org/10.1016/j.media.2017.07.005>
15. D. Shen, G. Wu, H. I. Suk, A survey on deep learning in medical image analysis, *Annu. Rev. Biomed. Eng.*, **19** (2017), 221–248. <https://doi.org/10.1146/annurev-bioeng-071516-044442>
16. C. Case, J. Casper, G. Damos, Deep Speech: Scaling up end-to-end speech recognition, *Comput. Sci.*, 2014.
17. J. Bonnemain, L. Pegolotti, L. Liaudet, S. Deparis, Implementation and calibration of a deep neural network to predict parameters of left ventricular systolic function based on pulmonary and systemic arterial pressure signals, *Front. Physiol.*, **11** (2022), 1086. <https://doi.org/10.3389/fphys.2020.01086>
18. J. Bonnemain, M. Zeller, L. Pegolotti, S. Deparis, L. Liaudet, Deep neural network to accurately predict left ventricular systolic function under mechanical assistance, *Front. Cardiovasc. Med.*, **8** (2021), 752088. <https://doi.org/10.3389/fcvm.2021.752088>

19. L. Cai, L. Ren, Y. H. Wang, W. X. Xie, H. Gao, Surrogate models based on machine learning methods for parameter estimation of left ventricular myocardium, *R. Soc. Open Sci.*, **8** (2021), 201121. <https://doi.org/10.1098/rsos.201121>
20. M. L. Liu, L. Liang, W. Sun, Estimation of in vivo constitutive parameters of the aortic wall: a machine learning approach, *Comput. Methods Appl. Mech. Eng.*, **347** (2018), 201–217. <https://doi.org/10.1016/j.cma.2018.12.030>
21. H. Gao, W. G. Li, L. Cai, C. Berry, X. Y. Luo, Parameter estimation in a Holzapfel-Ogden law for healthy myocardium, *J. Eng. Math.*, **95** (2015), 231–248. <https://doi.org/10.1007/s10665-014-9740-3>
22. H. Gao, A. Aderhold, K. Mangion, Changes and classification in myocardial contractile function in the left ventricle following acute myocardial infarction, *J. R. Soc. Interf.*, **14** (2017), 20170203. <https://doi.org/10.1098/rsif.2017.0203>
23. S. F. Pravdin, H. Dierckx, L. B. Katsnelson, O. Solovyova, V. S. Markhasin, A. V. Panfilov, Electrical wave propagation in an anisotropic model of the left ventricle based on analytical description of cardiac architecture, *Plos One*, **9** (2014), e93617. <https://doi.org/10.1371/journal.pone.0093617>
24. G. A. Holzapfel, R. W. Ogden, Constitutive modelling of passive myocardium: a structurally based framework for material characterization, *Philos. Trans.*, **367** (2009), 3445–3475. <https://doi.org/10.1098/rsta.2009.0091>
25. H. M. Wang, H. Gao, X. Y. Luo, C. Berry, B. E. Griffith, R. W. Ogden, et al., Structure-based finite strain modelling of the human left ventricle in diastole, *Int. J. Numer. Methods Biomed. Eng.*, **29** (2013), 83–103. <https://doi.org/10.37059/tjosal.2013.29.1.83>
26. L. Fan, J. Yao, C. Yang, D. Xu, D. Tang, Patient-specific echo-based left ventricle models for active contraction and relaxation using different zero-load diastole and systole geometries, *Int. J. Comput. Methods*, **16** (2017), 1842014. <https://doi.org/10.1142/S0219876218420148>
27. G. Viatcheslav, A high-resolution computational model of the deforming human heart, *Biomech. Model. Mechanobiol.*, **14** (2015), 829–849. <https://doi.org/10.1007/s10237-014-0639-8>
28. Y. Xu, *Simulation on the Mechanic Behaviour of Heart Based on Finite Element Methods*, Ph.D thesis, Harbin Institute of Technology in Harbin, 2015.
29. M. D. McKay, R. J. Conover, J. Beckmanw, A comparison of three methods for selecting values of input variables in the analysis of output from a computer code, *Technometrics*, **21** (1979), 239–245. <https://doi.org/10.2307/1268522>
30. K. Stefan, Single-beat estimation of end-diastolic pressure-volume relationship: a novel method with potential for noninvasive application, *Am. J. Physiol.*, **291** (2006), H403–H412. <https://doi.org/10.1152/ajpheart.01240.2005>
31. G. Xavier, B. Antoine, B. Yoshua, Deep sparse rectifier neural networks, *J. Mach. Learn. Res.*, **15** (2011), 315–323.
32. G. A. Montague, M. T. Tham, M. J. Willis, A. J. Morris, Predictive control of distillation columns using dynamic neural networks, *IFAC Proceed. Vol.*, **25** (1992), 243–248. [https://doi.org/10.1016/S1474-6670\(17\)50999-4](https://doi.org/10.1016/S1474-6670(17)50999-4)

- 
33. M. D. McKay, R. J. Conover, J. Beckmanw, Adam: A method for stochastic optimization, *CoRR*, 2014. <https://doi.org/10.48550/arXiv.1412.6980>



AIMS Press

©2022 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)