Material Parameter Estimation for a Viscoelastic Stenosis Model Using a Variational Autoencoder Inverse Mapper

Carola Kruse, Masha Sosonkina, Md Fayaz Bin Hossen, Yaohang Li

Material Parameter Estimation for a Viscoelastic Stenosis Model Using a Variational Autoencoder Inverse Mapper

Carola Kruse[1], Masha Sosonkina[2], Md Fayaz Bin Hossen[2], and Yaohang Li[2]

[1]CERFACS, Toulouse, France
[2]Old Dominion University, Norfolk, Virginia, USA

Abstract

Coronary artery disease, a prevalent condition often leading to heart attacks, may cause abnormal wall shear stresses near stenosed regions generating high frequent acoustic shear waves. In a previous study, a viscoelastic agarose gel was used to model the human tissue and it was shown that two material parameters of the gel could be estimated with a high certainty using a classical inverse problem. Given the high computational cost of traditional methods, this paper explores machine learning (ML) alternatives, particularly a Variational Autoencoder Inverse Mapper (VAIM). VAIM, previously successful in nuclear physics, uses neural networks to approximate forward and backward mappings and learn posterior parameter distributions. This paper validates previous research by generating data around ground truth values, demonstrating VAIM's ability to estimate two material parameters effectively. Further, it addresses realistic applications by training and testing on noisy data and generalizing findings across different intervals of signal damping.

1 Introduction

Coronary artery disease is an increasingly frequent medical condition and often a cause of a patient experiencing a heart attack. To model the effect of a plaque in a coronary artery, it is observed that the blood flow past the stenosed region becomes turbulent and creates abnormal variations in wall shear stresses [7]. These shears drive low-amplitude acoustic shear waves through the soft tissue in the thorax which appear at the chest wall and can be measured non-invasively by placing sensors on the skin [1, 20, 21].

Early steps towards the development of a non-invasive diagnostic tool for detecting a coronary artery stenosis have been explored in a previous project,

where the human tissue is described by a viscoelastic material. The problem is formulated as an inverse problem, i.e., in view of the application, by measuring the signal at the chest wall, the goal is to detect the location and the size of the stenosis. An experimental mock-up model, a cylinder filled with a viscoelastic agarose gel modeling human tissue, was available [11]. The first step was to estimate the material and damping parameters of the agarose gel [9, 10], which could then be used in more advanced inverse problems to find the position of the stenosis.

Parameter estimation is a fundamental task in various scientific and engineering disciplines, such as geophysics, medical imaging, and engineering. Traditionally, this task involves using statistical methods to estimate the parameters of a model that best fits the data. Typically, the process of an inverse problem for parameter estimation of a PDE requires the following steps (see, e.g., [4, 6]): 1. The formulation of the forward problem, used for solving the PDE with given parameters to predict the output. 2. The definition of a cost function. Common choices include the least squares difference or more sophisticated statistical measures that account for measurement noise and model uncertainties. 3. The optimization: The parameters are adjusted iteratively to minimize the cost function. These could be gradient-based methods. Literature about inverse problem approaches for parameter estimations for (visco-) elastic materials comparable to our approach includes, for example, the articles [12, 13]. In the experiments in our previous works [9, 10], a weight was attached to compress the gel in the cylinder, which was then removed abruptly and the resulting damped oscillations were measured [9]. Since the oscillations appear only in the vertical axis, the computational model was reduced to one dimension. In this first study, the experimental data was only used to have a rough estimate of the real material parameters, found manually, by trial and error. The inverse problem was then run as a proof of concept on the generated data. It was found that two of the parameters could be determined with high certainty, whereas the model was less sensitive to two more parameters.

The most costly component in the inverse problem is the forward solver. It has to be executed up to several hundreds of times in the optimization and in the particular application, the wave length of the signal is very small. We thus need a very accurate numerical method to capture well the signal in the temporal as well as spatial domain. In [8], a numerical scheme based on a discontinuous Galerkin method has been developed for solving the viscoelastic wave equation efficiently in terms of accuracy and computation time. While the strategy worked well in the above-described 1D case, it is, however, not efficient enough even for a still simplified radial-symmetric 2D setup in which oscillations were generated in the horizontal and vertical axes. Hence, a classical inverse problem setup seemed infeasible, and alternative solution techniques were necessary. One such alternative is to use machine learning (ML). To give some examples of a vast literature of ML for parameter estimation, the studies in [15, 19] use a convolutional neural network (CNN) [18] to estimate the material parameters for poro- and viscoelastic partial differential equations for ultrasound tomography. The CNN is applied to image samples, which show wave

interaction with poroelastic material. In the following work, we are, however, interested in ML models that are designed for inverse and ill-posed problems and are capable of many input parameters. In prior work [2], we have developed a Variational Autoincoder Inverse Mapper (VAIM) to address the ambiguity issue in inverse problems, which have been successfully applied to infer tens of input parameters in nuclear physics [3]. VAIM adopts an autoencoder-based neural network architecture, where the encoder and decoder neural networks approximate the forward and backward mapping, respectively, and a variational latent layer is incorporated to learn the posterior parameter distributions with respect to given observables. Here we propose VAIM as a solver to estimate the material parameters of the stenosis model problem for a given output wave signal.

In this paper, we apply the following steps to improve on the classical inverse problem solution with ML techniques:

1. We first generate data (output signals) in intervals around the ground truth values of [9] and show that VAIM is able to validate the previous results. It well estimates two of the material parameters, but is rather insensitive to two more parameters.

2. In the case of a realistic medical application, the wave signal will contain noise in the measurements. As a second experiment, we will learn and test on noisy data.

3. We train VAIM and infer on the interval $[0, 0.1]$, in which we see the principal oscillations and the dynamics of the problem. We show that the parameters found here generalize well to the interval $[0, 0.25]$, at which end the wave is damped out.

The paper is organized as follows. Section 2 describes the model problem followed by the VAIM overview and its tuning as a solver for the model problem. Results are presented in Section 4 while Section 5 concludes.

2 The Model Problem

We consider the model problem for $(x, t) \in \Omega \times I = [0, L] \times [0, T]$

$$\begin{aligned} \rho u_{tt} - \sigma_x &= f \\ u(0, t) &= 0 \\ \sigma(L, t) &= g(t) \\ u(x, 0) &= u_0(x) \\ u_t(x, 0) &= w_0(x) \ , \end{aligned} \tag{1}$$

where

$$\sigma(x, t) = E_1 u_{xt} + E_0 \left(u_x(x, t) - \int_0^t P_s(t - s) u_x(x, s) \, ds \right)$$

and

$$P(t) = P_0(t) + \sum_{n=1}^{N_P} P_n e^{-\frac{t}{\tau_n}}$$

for $P_0 > 0, \tau_n > 0, P_n \geq 0$, $n = 1, .., N_P$ and $P(0) = 1$. The integral in this formulation can be computed by numerical quadrature. However, a more convenient alternative is the *internal variable formulation* [7, 5]. It holds

$$P_s(t-s) = \sum_{n=1}^{N_P} \frac{P_n}{\tau_n} e^{\frac{-(t-s)}{\tau_n}}.$$

Define

$$\epsilon^n(t) = \int_0^t \frac{P_n}{\tau_n} e^{\frac{-(t-s)}{\tau_n}} u(s)\, ds.$$

Then

$$\epsilon_t^n(t) = \frac{P_n}{\tau_n} u(t) - \frac{1}{\tau_n} \int_0^t \frac{P_n}{\tau_n} e^{\frac{-(t-s)}{\tau_n}} u(s)\, ds.$$

To find ϵ^n, $n = 1, \ldots, N_P$ we therefore have to solve the following ordinary differential equations

$$\begin{aligned} \tau_n \epsilon_t^n + \epsilon^n &= P_n u(t) \\ \epsilon^n(0) &= 0. \end{aligned}$$

Using the definition of ϵ^n, it also follows

$$\sigma = E_1 u_{xt} + E_0 \left(u_x(t) - \sum_{n=1}^{N_P} \partial_x \epsilon^n(t) \right).$$

The quantities ϵ^n are called *internal variables*. Instead of solving the linear system (1), we therefore solve

$$\begin{aligned} \rho u_{tt} - E_1 u_{xt} - E_0 u_{xx} + E_0 \sum_{n=1}^{N_P} \epsilon_{xx}^n &= f \\ \tau_n \epsilon_t^n(t) + \epsilon^n(t) &= P_n u(t) \\ u(0,t) &= 0 \\ \sigma(1,t) &= g(t) \\ u(x,0) &= u_0(x) \\ u_t(x,0) &= w_0(x) \\ \epsilon^n(0) &= 0. \end{aligned} \tag{2}$$

We choose a material initially at rest and with reflecting boundary conditions at $x = 0$, which requires no forcing term, so we choose $f = 0$. We furthermore define the non-zero Neumann condition as an approximation to pulse traction. For obtaining a higher order regularity of the solution, we use a compactly supported input function, here a *Van Bladel* function. It is given by

$$g(t) = \begin{cases} A \cdot \exp\left(\frac{|ab|}{t(t+a-b)}\right), & \text{if } t \in (0, b-a) \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

The parameters are chosen as in [9]

$$A = 6 \cdot 10^3, \;\; a = 6 \cdot 10^{-3}, \;\; b = 20 \cdot 10^{-3}, \tag{4}$$

so that the amplitude of the wave signal corresponds to one of the early state experiments. Furthermore, the baseline material parameters are given by

$$\begin{aligned} E_0 = 2.2 \cdot 10^5 Pa, \; E_1 = 40 Pa \cdot s, \; \rho = 1010 kg/m^3, \; L = 0.053 \\ \tau_1 = 0.05 s, \; \tau_2 = 10 s, \; p_1 = 0.3, \; p_2 = 0.55. \end{aligned} \tag{5}$$

A graphical representation of the wave signal corresponding to these parameters is given in Figure 1. Note that the amplitude of the signal is of order 10^{-4} and that the signal is nearly damped out at about $T = 0.25$. The main dynamics are visible earlier, however.

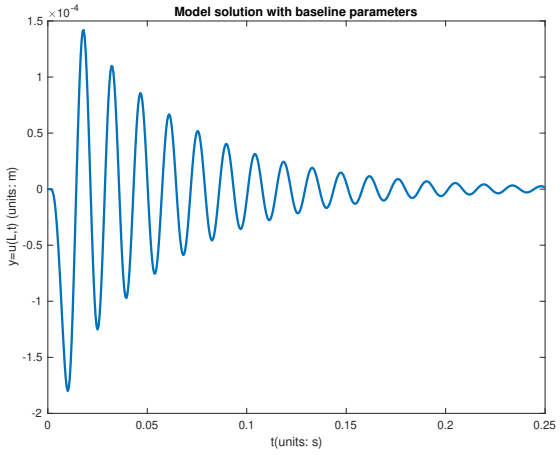


Figure 1: Signal output at $L = 0.053$ with baseline parameters (5).

3 VAIM Overview and Solution

The Variational Autoencoder Inverse Mapper (VAIM) [2] is an autoencoder-based, end-to-end neural network architecture for inverse problems. The architecture

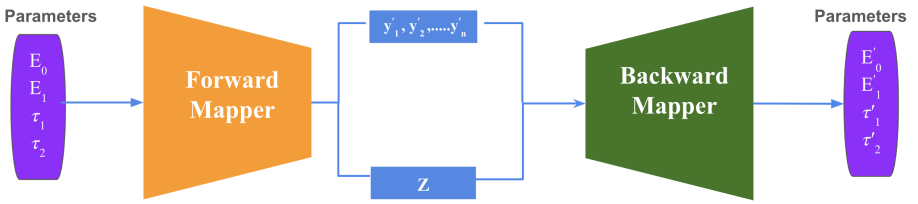


Figure 2: Architecture of VAIM for the model problem.

adopted for VAIM utilizes an autoencoder [16] consisting of two neural networks: an encoder and a decoder. The encoder and decoder approximate the forward and backward mappings, respectively. In VAIM, an inverse problem is framed as a statistical inverse problem, with parameters treated as random variables. Instead of generating deterministic parameter estimates for a given observable, VAIM approximates a probability distribution of the parameters. The most important component in VAIM is a variational latent layer, incorporated between the encoder and the decoder, whose purpose is to learn the patterns of the parameter distributions. This latent layer acts as part of the output for the encoder and part of the input for the decoder. Unlike the hidden layers of the encoder or decoder neural networks, the latent layer is constrained to follow specific well-known distributions, such as Gaussian or uniform distributions, for variational inference [14]. To solve a statistical inverse problem, sampling from the latent distribution for a given observable yields the corresponding parameter distributions. Specifically, a sensitive parameter will result in a narrow distribution, whereas an insensitive parameter will produce a wide distribution.

Figure 2 illustrates the architecture of VAIM for the model problem. During training, the forward mapper learns to map the parameters $\mathbf{P} = \{E_0, E_1, \tau_1, \tau_2\}$ to the output signals $\mathbf{Y} = \{y_1, \ldots, y_n\}$ and the latent space $\mathbf{Z}$ is used to capture the information lost during forward mapping. Then, the backward mapper reconstructs the parameter distributions $\mathbf{P}' = \{E_0', E_1', \tau_1', \tau_2'\}$ from the output signals and samples of the latent space $\mathbf{Z}$. The training process is to minimize the following loss function:

$$loss = \|\mathbf{P} - \mathbf{P}'\|_2^2 + \|\mathbf{Y} - \hat{\mathbf{Y}}'\|_2^2 + \mathrm{KL}\big(p(\mathbf{Z} \mid \mathbf{P}, \mathbf{Y}) \;||\; p(\mathbf{Z})\big),$$

where $\|\mathbf{P} - \mathbf{P}'\|_2^2$ is the error of reconstructing the parameters, $\|\mathbf{Y} - \hat{\mathbf{Y}}'\|_2^2$ is the prediction error of the output signals, and $\mathrm{KL}\big(p(\mathbf{Z} \mid \mathbf{P}, \mathbf{Y}) \;||\; p(\mathbf{Z})\big)$ denotes the Kullback–Leibler divergence of the posterior distribution $p(\mathbf{Z} \mid \mathbf{P}, \mathbf{Y})$ and prior distribution $p(\mathbf{Z})$. Here, we select a multi-dimensional standard Gaussian as the prior distribution, which is tractable while the samples are easy to generate.

3.1 Dataset Preparation

For the generation of data, we used a high-order temporal Discontinuous Galerkin discretization as described in more detail in [8]. We chose the polynomial degree

Name	$\mathbf{G_1}$	$\mathbf{G_2}$	Min	Max
E_0	213000	220000	175000	267000
E_1	42	40	34	64
τ_1	0.05	0.05	0.01	0.55
τ_2	10	10	6	36

Table 1: Ground-truth (columns $\mathbf{G_1}$ and $\mathbf{G_2}$) and boundary values (columns **Min** and **Max**) for the distribution interval of the model parameters shown in the column **Name**.

$d_t = 2$ and $M = 200$ time steps on the time interval [0,0.1]. The spatial discretization uses continuous finite elements with polynomial degree $d_x = 2$ on a mesh with 32 elements for the interval $[0, 0.053]$. We generated $N = 50\,000$ data curves, i.e., output signals.
Thinking of the final application of the stenosis detection device, different people exhibit different morphology, and thus material parameters. We address these differences in twofold manner: (1) The training and testing data were generated with $(E_0, E_1, \tau_1, \tau_2)$ uniformly distributed in a sufficiently large interval around the baseline set of values (denoted here $\mathbf{G_2}$), as provided in 5. (2) Another set of baseline parameters, $\mathbf{G_1}$, was selected inside this interval to more specifically simulate the parameter 4-tuple for another given person. Table 1 shows the distribution interval boundaries and the vales of the $\mathbf{G_1}$ and $\mathbf{G_2}$ parameter sets, which we call ground truths.

We denote the time series y_j, $j = 1, \ldots, N$ as *ideal output signal*. It is generated for $i = 1 \ldots, M$ time steps as

$$y_j^i = u(L, t_i) \,, \tag{6}$$

such that u is the numerical solution of the system (2) at $L = 0.053$ with parameters $(E_0, E_1, \tau_1, \tau_2)_j$ drawn uniformly in the intervals given in Table 1. We have noticed that the training loss is of order 10^{-4}. Since the signal itself is also of this amplitude, we observed that training on unscaled data leads to meaningless results. We therefore normalize the parameters and output signals into $[0, 1]$.

We also investigate the case when the data is not exact, which models the effect of noise such as measurement errors. We call these data points *observational data*. Observational data $\tilde{y}_j$ is generated by adding a normally distributed additive noise with mean 0 and different noise level l as a standard deviation to the ideal output signal

$$\tilde{y}_j^i = y_j^i + l\mathcal{N}(0, 1) \tag{7}$$

at each time step i, $i = 1, \ldots, M$. We choose two different noise levels: with $l_1 = 5\%$ and $l_2 = 10\%$ noise. We scale the noise by 10^{-4} to obtain noise of the same magnitude as that of the signal, and thereby $l_1 = 5 \cdot 10^{-6}$ and $l_2 = 10 \cdot 10^{-6}$. The observational data is then scaled in the same way as the data in the case of the exact (ideal) output signal.

Hyperparameter	Value
Optimizer	Adam
Loss function	MAE
Batch Size	512
Epochs	5,000 – 10,000
Learning Rate	1e-4

Table 2: VAIM hyperparameters for which reference-implementation values were chosen. "**MAE**" denotes mean absolute error.

3.2 Hyperparameter Selection

We have selected VAIM hyperparameters manually based on our knowledge of data and its scaling (see Section 3.1) and the problem difficulty as investigated using traditional numerical integration techniques and reported in our earlier paper [8]. In addition to ubiquitous hyperparameters, such as learning rate, the choice of optimizer, and loss metric type, we considered the specific VAIM building blocks, the backward and forward mappers, and sampling layer hyperparameters, while aiming to make a minimum number of hyperparameter changes so that the model remains sufficiently general, yet computationally efficient. In particular, we have left the mappers "as is" vz. the number and specification of layers except for the change of the activation function in the last layer of the decoder (the backward mapper). This change for the Sigmoid function was motivated by the MinMax dataset scaling which confined all the data in the interval between zero and one, and therefore, corresponds to the Sigmoid function output. Most of the other VAIM hyperparameters remained the same from the reference VAIM implementation or were assigned reasonable values that led to a good convergence (see Table 2 for their list and values).

4 Results

Experiments were performed on the Old Dominion University cluster Wahab using on-demand environments for Jupyter notebooks (4 nodes 1 GPU each node). The Wahab cluster consists of 158 compute nodes and 6320 computational cores using the state-of-the-art Intel "Skylake" Xeon Gold 6148 processors (20 CPU cores per chip; 40 cores per node). Each compute node has 384 GB of RAM. To accommodate the rising computing demand from machine learning and artificial intelligence workloads, there are 18 accelerator compute nodes, each of which is equipped with four NVIDIA V100 graphical processing units (GPU), for a total of 72 GPUs. A 100Gbps EDR Infiniband high-speed interconnect provides low-latency, high-bandwidth communication between nodes to support massively parallel computing as well as data-intensive workloads. Wahab is equipped with a dedicated high-performance Lustre scratch storage (350 TB usable capacity) and is connected to the 1.9 PB university-wide home/long-term research data networked filesystem. Wahab also has 45 TB of CEPH block

storage that can be provisioned for user data in the virtual environment.

The total dataset was split into train (95%) and test (5%) subsets. Out of the train set, 20% was randomly allocated for the validation set. These splitting portions are common and lead to reliable predictions, assuming there is no overfitting, due to a much smaller test-set size.

The prediction results of the four input parameters under consideration are difficult to represent on a single conventional plot, which typically contains at most 3 dimensions plus color gradation. This difficulty is exacerbated by the large difference in the raw parameter magnitudes. Hence, we resort to representing graphically the parameter scaled values in the so-called *Parallel Coordinates*, originally proposed in [17] to "transform multivariate relations into 2-D patterns, a property that is well suited for Visual Data Mining". Parallel-Coordinate (PC) plots are being extensively used in data-science papers as evidenced by more than 400 citations of [17] dating as recent as 2024.

Recall that results are obtained in pursuit of three objectives: (1) Consider an ideal case when measurements are exact and investigate corresponding inverse problem input parameter interdependence. (2) Consider a real-life scenario in which there is an error in measurements and possible uncertainty in the input parameter values. (3) Consider how the measurements on shorter initial time intervals generalize for longer periods in which fewer dynamics are observed, however. Sections 4.1 to 4.3 describe results of the three objectives, respectively.

4.1 Exact Measurements of the Output Signal

We can investigate the relative effects of the input parameters by visually representing them using PC plotting. In a PC plot, Fig. 3 visualizes the predictions made for all the test data along with the ground-truth values in the case of exact output signal measurements. Note that, in Fig. 3 and all the subsequent PC plots, the input parameter ordering $p_i,\ i = 1, 2, 3, 4$, where $p_1 = E_0,\ p_2 = E_1,\ p_3 = \tau_1,\ p_4 = \tau_2$ corresponds to the natural ordering $a_i,\ i = 1, 2, 3, 4$ of the vertical PC plot axes, such that a line between a value on axis a_{i-1} and a_i indicates that the corresponding parameter values p_{i-1} and p_i belong to the same 4-tuple of input parameters defining an output signal and there is a magnitude relationship between the two parameters. The 4-tuple corresponding to ground truth is connected with dashed lines: Red-colored for the set $\mathbf{G_1}$ of ground truth parameter values $E_0 = 213000, E_1 = 42, \tau_1 = 0.05, \tau_2 = 10$ and yellow-colored for the set $\mathbf{G_2}$ of ground truth parameter values $E_0 = 220000, E_1 = 40, \tau_1 = 0.05$, and $\tau_2 = 10$. The predicted parameter value sets are $\mathbf{G_1'}$ and $\mathbf{G_2'}$ with 4-tuples $(212746.92, 42.05, 0.265, 24.80)$ and $(219682.62, 40.07, 0.262, 24.97)$, respectively.

We observe in Fig. 3 that the ground truth for the two parameters E_0 and E_1 is well inside the cyan "cover" of all the prediction curves. Note that the almost symmetrical shape of the "cover" is to be expected since all the parameters for the datasets were generated using uniform distribution (see Section 3.1). Hence, we consider these two parameters to be well-predicted as also demonstrated by the PC plot shown in Fig. 4a in which a prediction only for a single tested

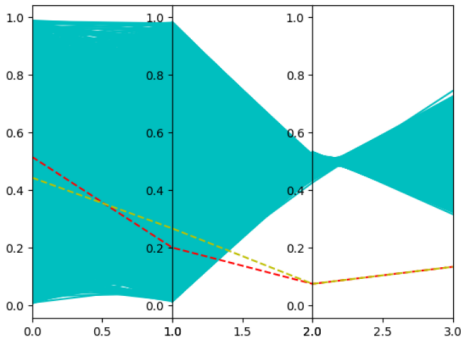


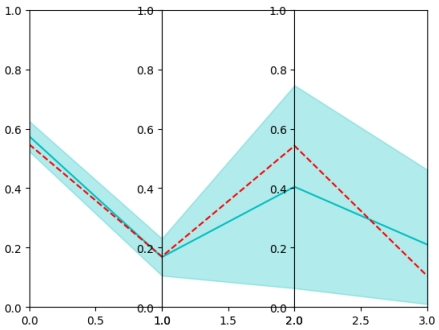
Figure 3: PC plot of the predicted (cyan) and two sets (yellow and red) of the ground-truth input parameter values. The output variable y is measured exactly (without noise) and datasets are scaled.

4-tuple is shown out of the entire test dataset. The prediction (cyan curve) is very close to the tested values for the E_0 and E_1 parameters while there is a larger spread (as determined from the larger width of the shaded-cyan area) of the actual and predicted values for the τ parameters. The shaded-cyan area width shows the two standard deviations (std) of the average difference between the predicted (from the latent space) and tested input parameters.Figure 4b presents the plots of both the training and validation losses obtained in training epochs, starting with the epoch 50 for plot scaling sake. The two curves exhibit the same decreasing tendency and overlap. Hence overfitting of the model does not occur.

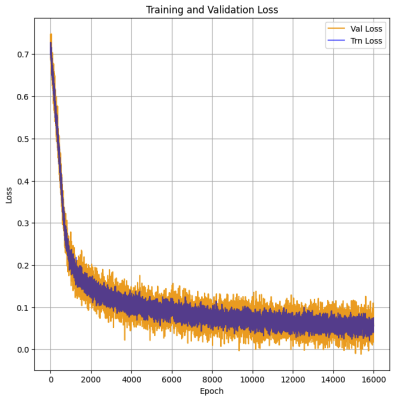
On the other hand, in Fig. 3, the two sets of the ground truths τ_1 and τ_2 are outside of their predicted value set (shown on axes a_3 and a_4, respectively), Moreover, the widths of their predicted-value "covers" are very narrow suggesting that VAIM did not have enough train-data variability to search their values. We hypothesize that for each τ parameter, either "one value fits all" or that these parameter values are unimportant for the output signal y definition. To verify this hypothesis, we used the predicted values of τ_1 and τ_2, which were far from those in the ground truth, along with the well-predicted E_0 and E_1 for the construction of the output signal y using Matlab, as shown in Fig. 5. Then, we compared the obtained y signal output values with the actual y measurements. For both $\mathbf{G_1}$ and $\mathbf{G_2}$ sets of input parameter values, the two curves are practically indistinguishable. The means, standard deviations, and standard errors of the two ground truths are presented in Table 3.

4.2 Noisy Output Signal

For a real-life scenario in which there may be an error in output signal y measurements, we use the *observational data* for y as generated using Eq. (7). For the two noise levels, 5% and 10%, the PC plots of the entire test set are provided in Fig. 6, respectively, with the same ground truths $\mathbf{G_1}$ and $\mathbf{G_2}$ as



(a) PC plot of a single predicted (cyan) and test (red) input parameter value 4-tuple. The shaded-cyan area corresponds to the two-std's of the average difference. The output variable y is measured exactly (without noise) and datasets are scaled.

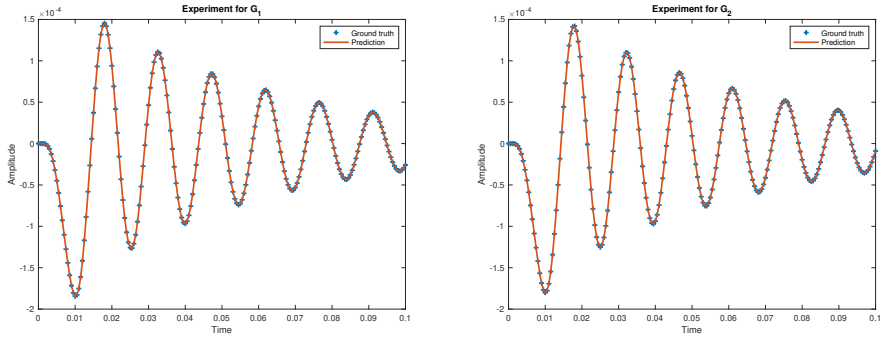


(b) Training and validation losses corresponding to the training of the model that produced the prediction in Fig. 4a.

Figure 4: Analysis of the resulting VAIM ML model.

	$\mathbf{G_1'}$	$\mathbf{G_2'}$
MAE	8.9125e-08	1.0178e-07
std	5.7237e-08	6.5028e-08
SE	4.0372e-09	4.5867e-09

Table 3: Mean absolute error (**MAE**), standard deviation (**std**), and standard error (**SE**) for the difference between the predictions $\mathbf{G_1'}$ and $\mathbf{G_2'}$ and their corresponding ground-truth output signals on the interval $[0, 0.1]$.



(a) Baseline parameters $\mathbf{G_1}$. (b) Baseline parameters $\mathbf{G_2}$.

Figure 5: Output y signals corresponding to the predicted and two ground-truth sets of parameters.

in Section 4.1. By comparing Fig. 6 and Fig. 3, we note that the area covered by the predicted (cyan) curves is wider while almost symmetrical. This indicates that there is a great deal of variability in data to make a prediction. On the other hand, the ML model does not appear to overfit as seen in Fig. 7, which plots the model training loss per epoch of training (blue) along with the obtained validation loss (orange curve) in the given epoch for 5% and 10% noise levels. The figures show that both curves exhibit the same general decreasing trend. The two ground-truth sets of values predicted for the observational (noisy) data are

$$\mathbf{G_1^{(5)}} = (214255.88, 40.64, 0.324, 24.40) \qquad \mathbf{G_2^{(5)}} = (221522.55, 37.78, 0.141, 31.41)$$
$$\mathbf{G_1^{(10)}} = (215087.39, 40.41, 0.383, 17.91) \qquad \mathbf{G_2^{(10)}} = (216846.45, 40.00, 0.306, 22.53)$$

for the 5% and 10% noise, respectively. Similar to Fig. 4a, the plot in Fig. 8 demonstrates the two-std width (shaded-cyan area) between the actual and predicted test parameter values, but for the 10%-noise level in this case. For the E_0 and E_1 parameters, the wider shaded-area band indicates more uncertainty in these parameters with noise, while no qualitative effect is noticeable in the predictions of the accuracy of the τ_1 and τ_2 predictions with noise.

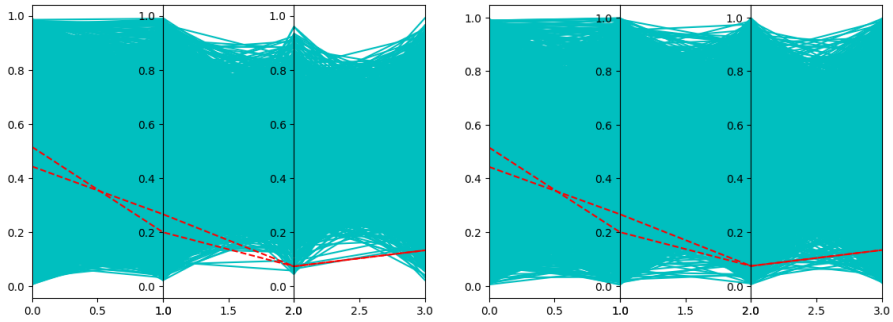
Figure 9 compares the output signal y obtained with the baseline ground-truth parameter values y to those obtained with predicted values from noisy output signals. For both $\mathbf{G_1}$ and $\mathbf{G_2}$ sets of input parameter values, the signals from predicted values still match the *ideal* signal well, which is reflected in Table 4 showing the mean, standard deviation, and standard errors.

4.3 Generalization: Extrapolation to Longer Time Spans

As shown in Fig. 1, the wave is damped out quickly, and the main dynamics occur early on. We have also observed that the material parameters can be

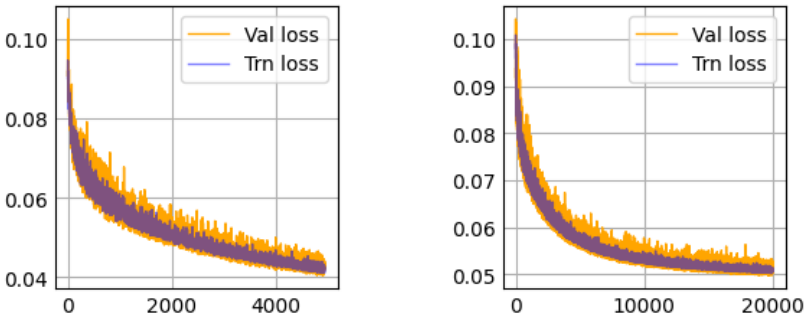
	$\mathbf{G_1^{(5)}}$	$\mathbf{G_2^{(5)}}$	$\mathbf{G_1^{(10)}}$	$\mathbf{G_2^{(10)}}$
MAE	3.5502e-06	8.0698e-06	4.4056e-06	4.9998e-06
std	1.9491e-06	2.3782e-06	2.8909e-06	3.2332e-06
SE	1.3748e-07	1.6775e-07	2.0391e-07	2.2805e-07

Table 4: Mean absolute error (**MAE**), standard deviation (**std**), and standard error (**SE**) for the difference between the predictions generated from the noisy output signal with 5% and 10% noise levels and the corresponding ground truths on the interval $[0, 0.1]$.



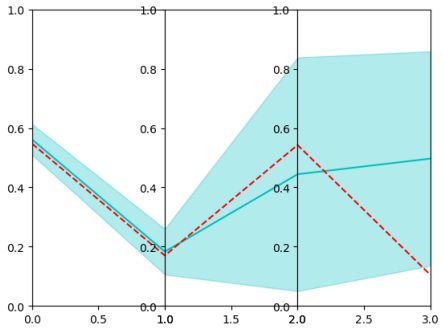
(a) Noise 5%. (b) Noise 10%.

Figure 6: PC plot of the predicted (cyan) and two sets (red) of the ground-truth input parameter values for the *observed* output variable y that include noise levels: (a) 5% (b) 10%. Datasets are scaled.



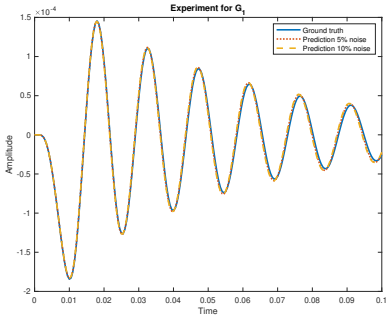
(a) Noise 5%. (b) Noise 10%.

Figure 7: Training and validation losses corresponding to the training of the model that produced the predictions in Fig. 6 for the observable data. The plots display curves starting with the 50th epoch for the sake of visualization.

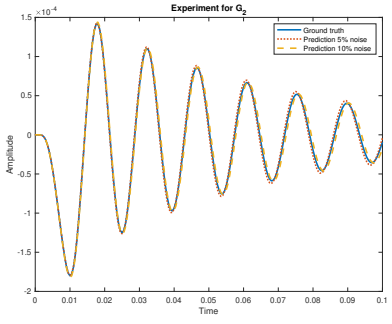


(a) Noise 10%.

Figure 8: PC plot of a single predicted (cyan) and test (red) input parameter value 4-tuple. The shaded-blue area corresponds to the two-std's of the average difference for the *observed* output variable y that includes a 10% noise level. Datasets are scaled.

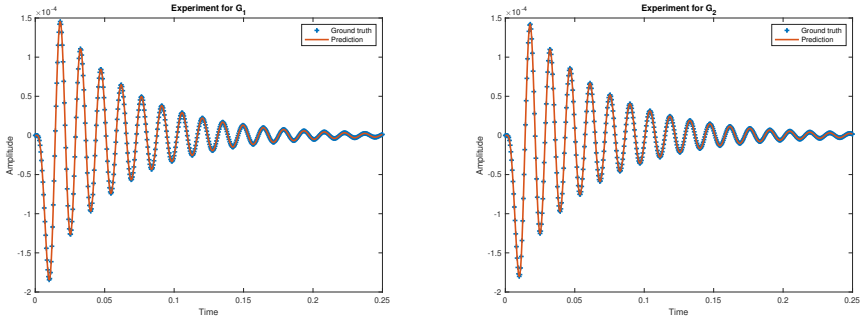


(a) Baseline parameters $\mathbf{G_1}$.



(b) Baseline parameters $\mathbf{G_2}$.

Figure 9: Output signals for baseline parameters and their predictions with 5% and 10% noise.



(a) Baseline parameters $\mathbf{G_1}$. (b) Baseline parameters $\mathbf{G_2}$.

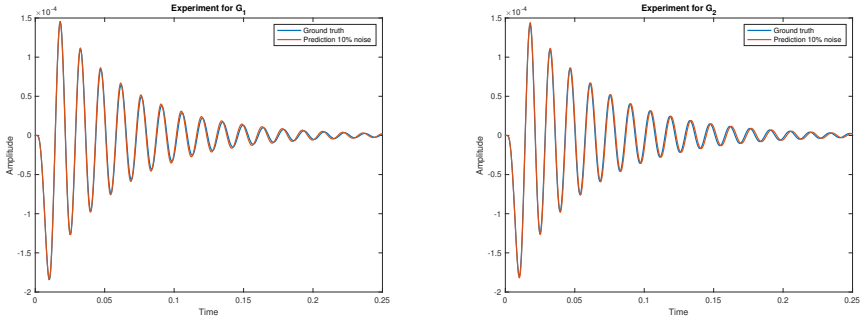
Figure 10: Output signal y extrapolated to the interval $[0, 0.25]$ corresponding to the predicted and two ground-truth sets of parameters.

	$\mathbf{G_1'}$	$\mathbf{G_2'}$	$\mathbf{G_1^{(10)}}$	$\mathbf{G_2^{(10)}}$
MAE	6.5735e-08	7.7070e-08	3.3165e-06	3.7970e-06
std	4.9783e-08	5.6500e-08	2.4924e-06	2.8033e-06
SE	2.2241e-09	2.5242e-09	1.1135e-07	1.2524e-07

Table 5: Mean absolute error (**MAE**), standard deviation (**std**), and standard error (**SE**) for the difference between the predictions generated from the exact and 10%-noise output signals and the corresponding ground truths extended to the interval $[0, 0.25]$.

accurately predicted within the interval $[0, 0.1]$, even though the wave is only fully damped out around $t = 0.25$. In our final experiment, we aim to determine if it is possible to use the predicted values from the interval $[0, 0.1]$ to still achieve a good fit over the entire interval $[0, 0.25]$. The advantage of generalizing from the smaller to the larger interval is the reduction in the number of grid points in the time domain (here by a factor of 2.5). This translates into fewer computational operations and lower energy consumption.

We first present the results for the ideal output signal, i.e., without noise. To preserve the discretization accuracy, we keep the same interval size and use 500 points. As presented in Fig. 10, the curves corresponding to the ground truth and their predicted parameters $\mathbf{G_1'}$ and $\mathbf{G_2'}$ extended to the interval $[0, 0.25]$ are almost indistinguishable. The mean, standard deviation, and standard errors are given in Table 5, in which the same metrics are shown for the 10%-noise level predictions $\mathbf{G_1^{(10)}}$ and $\mathbf{G_2^{(10)}}$. The latter predicted input parameters generate the output y on the interval $[0, 0.25]$ as shown in Fig. 11.



(a) Baseline parameters $\mathbf{G_1}$. (b) Baseline parameters $\mathbf{G_2}$.

Figure 11: Output signals y extrapolated to the interval $[0, 0.25]$ corresponding to the predicted with 10% noise and two ground-truth sets of parameters.

5 Conclusions

In this paper, we have presented the VAIM deep learning model to predict material parameters of an acoustic visco-elastic wave equation for a stenosis detection model. The observations of the previous paper [9], in which a classical inverse problem has been formulated for this parameter estimation, have been confirmed. For ideal data, i.e. without noise, we were able to predict well two material parameters. The model is, however, insensitive to two more parameters, suggesting that the inclusion of a memory integral into the model is unnecessary. For data with 5% and 10% noise, the higher variability in the data led to better predictions for the memory integral parameters. While the signal is damped out only after 0.25 seconds, we have shown that it is enough to learn and to infer up to $T = 0.1$. The signal can then be extended to the larger interval without adding a significant error, all this with the advantage of training on a smaller dataset. In addition to [9], we were able to show that we may predict material parameters on a large interval around the baseline values. From the application point of view, this is necessary for a stenosis detection tool, as different specimens will have different material properties.

Acknowledgement

The work is partially supported by the U.S. Department of Energy (DOE) Research on Artificial Intelligence and Machine Learning For Autonomous Optimization And Control Of Accelerators And Detectors grant, under award number DE-SC0024603 and by DOE Office of Science, Office of Basic Energy Sciences, Computational Chemical Sciences (CCS) Research Program under work proposal number AL-18-380-057 through the Ames National Laboratory, operated by Iowa State University under contract No. DE-AC00-07CH11358.

References

[1] Y. M. Akay, M. Akay, W. Welkowitz, and J. Kostis, Noninvasive detection of coronary artery disease, IEEE Engineering in Medicine and Biology Magazine, 13 (1994), pp. 761–764.

[2] M. Almaeen, Y. Alanazi, N. Sato, W. Melnitchouk, M. P. Kuchera, and Y. Li, Variational autoencoder inverse mapper: An end-to-end deep learning framework for inverse problems, in 2021 International Joint Conference on Neural Networks (IJCNN), 2021, pp. 1–8.

[3] M. Almaeen, T. Alghamdi, B. Kriesten, D. Adams, Y. Li, and H.-W. L. ans Simonetta Liuti, Vaim-cff: A variational autoencoder inverse mapper solution to compton form factor extraction from deeply virtual exclusive reactions, 2024.

[4] R. C. Aster, B. Borchers, and C. H. Thurber, Parameter estimation and inverse problems, Elsevier, 2018.

[5] H. Banks and N. Luke, Modelling of propagating shear waves in biotissue employing an internal variable approach to dissipation, COMMUNICATIONS IN COMPUTATIONAL PHYSICS Commun. Comput. Phys, 3 (2008), pp. 603–640.

[6] H. Banks and H. Tran, Mathematical and Experimental Modeling of Physical and Biological Processes (1st ed.), Chapman and Hall/CRC, 2009.

[7] H. T. Banks, J. H. Barnes, A. Eberhardt, H. T. Tran, and S. Wynne, Modeling and computation of propagating waves from coronary stenoses, 2000.

[8] H. T. Banks, M. J. Birch, M. P. Brewin, S. E. Greenwald, S. Hu, Z. R. Kenz, C. Kruse, M. Maischak, S. Shaw, and J. R. Whiteman, High-order space-time finite element schemes for acoustic and viscodynamic wave equations with temporal decoupling, International Journal for Numerical Methods in Engineering, 98 (2014), pp. 131–156.

[9] H. T. Banks, M. J. Birch, M. P. Brewin, S. E. Greenwald, S. Hu, Z. R. Kenz, C. Kruse, S. Shaw, and J. R. Whiteman, Material parameter estimation and hypothesis testing on a 1d viscoelastic stenosis model: Methodology, Journal of Inverse and Ill-Posed Problems, 21 (2013), pp. 25–57.

[10] H. T. Banks, M. J. Birch, M. P. Brewin, S. E. Greenwald, S. Hu, Z. R. Kenz, C. Kruse, S. Shaw, and J. R. Whiteman, Model validation for a noninvasive arterial stenosis detection problem, Mathematical Biosciences and Engineering, 11 (2014), pp. 427–448.

[11] M. P. Brewin, M. J. Birch, D. J. Mehta, J. W. Reeves, S. Shaw, C. Kruse, J. R. Whiteman, S. Hu, Z. R. Kenz, H. T. Banks, and S. E. Greenwald, Characterisation of elastic and acoustic properties of an agar-based tissue mimicking material, Annals of Biomedical Engineering, 43 (2015), pp. 2587–2596.

[12] S. Catheline, J.-L. Gennisson, G. Delon, M. Fink, R. Sinkus, S. Abouelkaram, and J. Culioli, Measurement of viscoelastic properties of homogeneous soft solid using transient elastography: An inverse problem approach, The Journal of the Acoustical Society of America, 116 (2004), pp. 3734–3741.

[13] S. Chen, M. Fatemi, and J. F. Greenleaf, Quantifying elasticity and viscosity from measurement of shear wave speed dispersion, The Journal of the Acoustical Society of America, 115 (2004), pp. 2781–2785.

[14] A. K. David M. Blei and J. D. McAuliffe, Variational inference: A review for statisticians, Journal of the American Statistical Association, 112 (2017), pp. 859–877.

[15] K. M. Graczyk and M. Matyka, Predicting porosity, permeability, and tortuosity of porous media from images by deep learning, Scientific Reports, 10 (2020), p. 21488.

[16] G. E. Hinton and R. R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science, 313 (2006), pp. 504–507.

[17] A. Inselberg and B. Dimsdale, Parallel coordinates: a tool for visualizing multi-dimensional geometry, in Proceedings of the First IEEE Conference on Visualization: Visualization '90, 1990, pp. 361–378.

[18] Y. LeCun, Y. Bengio, and G. Hinton, Deep learning, nature, 521 (2015), pp. 436–444.

[19] T. Lähivaara, L. Kärkkäinen, J. M. J. Huttunen, and J. S. Hesthaven, Deep convolutional neural networks for estimating porous material parameters with ultrasound tomography, The Journal of the Acoustical Society of America, 143 (2018), pp. 1148–1158.

[20] J. F. Sangster and C. M. Oakley, Diastolic murmur of coronary artery stenosis., Heart, 35 (1973), pp. 840–844.

[21] J. Semmlow and K. Rahalkar, Acoustic detection of coronary artery disease, Annual Review of Biomedical Engineering, 9 (2007), pp. 449–469.