Discovering uncertainty: Bayesian constitutive artificial neural networks

Kevin Linka[a], Gerhard A Holzapfel[b,c], Ellen Kuhl[d,e]

[a]*Institute of Applied Mechanics, RWTH Aachen, Aachen, Germany*
[b]*Institute of Biomechanics, Graz University of Technology, Graz, Austria*
[c]*Department of Structural Engineering, Norwegian University of Science and Technology, Trondheim, Norway*
[d]*Department of Mechanical Engineering, Stanford University, Stanford, CA, United States*
[e]*Institute of Applied Mechanics, FAU Erlangen–Nürnberg, Erlangen, Germany*

Abstract

Understanding uncertainty is critical, especially when data are sparse and variations are large. Bayesian neural networks offer a powerful strategy to build predictable models from sparse data, and inherently quantify both, aleatoric uncertainties of the data and epistemic uncertainties of the model. Yet, classical Bayesian neural networks ignore the fundamental laws of physics, they are non-interpretable, and their parameters have no physical meaning. Here we integrate concepts of Bayesian learning and constitutive neural networks to discover interpretable models, parameters, and uncertainties that best explain soft matter systems. Instead of training an individual constitutive neural network and learning point values of the network weights, we train an ensemble of networks and learn probability distributions of the weights, along with their means, standard deviations, and credible intervals. We use variational Bayesian inference and adopt an efficient backpropagation-compatible algorithm that approximates the true probability distributions by simpler distributions and minimizes their divergence through variational learning. When trained on synthetic data, our Bayesian constitutive neural network successfully rediscovers the initial model, even in the presence of noise, and robustly discovers uncertainties, even from incomplete data. When trained on real data from healthy and aneurysmal human arteries, our network discovers a model with more stretch stiffening, more anisotropy, and more uncertainty for diseased than for healthy arteries. Our results demonstrate that Bayesian constitutive neural networks can successfully discriminate between healthy and diseased arteries, robustly discover interpretable models and parameters for both, and efficiently quantify uncertainties in model discovery. We anticipate our approach to generalize to other soft biomedical systems for which real-world data are rare and inter-personal variations are large. Ultimately, our calculated uncertainties will help enhance model robustness, promote personalized predictions, enable informed decision-making, and build confidence in automated model discovery and simulation.

Our source code, data, and examples are available at https://github.com/LivingMatterLab/CANNs.

Keywords: uncertainty; Bayesian inference; Bayesian neural networks; constitutive artificial neural networks; automated model discovery

1. Motivation

When someone asks you *'How confident are you in your neural network?'*, wouldn't you want to respond *'Very.'*? But what makes you believe that you are very confident? And how exactly do you quantify very? This is precisely what this manuscript is about.

Plain neural networks are prone to overfitting and non-interpretable [9]. They make overly confident decisions, generalize poorly, and are unsuitable for model discovery [40]. Here we address these limitations by using regularized variational Bayesian learning to *sparsify the weight vector* of a constitutive neural network [26] and *quantify the uncertainty* in the remaining non-zero weights [30]. The underlying idea is to represent the weights of a constitutive neural network by their probability distributions. Then, instead of training an *individual* constitutive neural network to learn point values of network weights [21], we train an *ensemble* of networks to learn probability distributions

of the weights, with means, standard deviations, and credible intervals [25]. In other words, each network of the ensemble has its own network weights that we draw from shared probability distributions. We learn these probability distributions using Bayesian inference [2].

In practice, probability distributions of neural network weights can be quite complex, and, to complicate matters, the probabilities of individual weights can dependent on one another [3]. This makes, the *exact Bayesian inference* of the network weights a challenging if not infeasible task. Instead, we can numerically approximate Bayesian inference, and three classes of algorithms have emerged to do so: Markov Chain Monte Carlo methods that approximate posterior distribution of the network weights [30], dropout methods that probe discrete ensembles by setting subsets of weights to zero [8], and variational inference [10]. Here we use *variational Bayesian inference* [17] and adopt an efficient principled backpropagation-compatible algorithm [3] that approximates the probability distributions of our network weights with simpler distributions by minimizing their divergence through variational learning [12].

Bayesian neural networks are by no means new, in fact, they were first introduced in the early 1990s [25]. Within the science and engineering communities, they are enjoying increasingly popularity because they allow us to quantify both aleatoric and epistemic uncertainties [33]. *Aleatoric* or *data uncertainties* result from an inherent randomness or variability in the data, which is irreducible. *Epistemic* or *model uncertainties* result from a lack of knowledge or limitations in the model, which are potentially reducible, either by collecting more data or by refining the model [16]. We can quantify aleatoric uncertainties with both frequentist and Bayesian statistics, but only Bayesian statistics can quantify epistemic uncertainties [8]. It does so by treating the model parameters as probability distributions [15]. This introduces uncertainties in the model itself, which we could reduce by collecting more data.

In many science and engineering applications, clean data are rare and cumbersome to collect [1]. At the same time, we have a solid understanding of the underlying physics that our scientific community has built over many decades [14]. Naturally, this raises the question how to best build this knowledge into efficient learning machines [36]. Two different strategies have emerged to integrate physics-based knowledge into neural network models: *Physics Informed Neural Networks* or *PINNs* that incorporate physics into the loss function using additional terms [37], and *Constitutive Artificial Neural Networks* or *CANNs* that hardwire the underlying physics into the neural network design [19]. Our recent application of physics informed neural networks to real-world nonlinear dynamical systems has revealed the critical need to supplement these networks with uncertainty quantification, especially in situations where the underlying physics are not entirely known [20]. Here we build on this experience and explore the importance of uncertainty quantification in the context of constitutive artificial neural networks [21]. Our guiding question is: How can we discover the best model, parameters, and uncertainties for real-world nonlinear soft matter systems?

Towards this goal, in Section 2, we introduce the concept of Bayesian constitutive neural networks and derive their characteristic three-term loss function in the context of regularized variational Bayesian inference. In Section 3, we design a family of isotropic Bayesian constitutive neural networks and discover the best model, parameters, and uncertainties to explain *synthetic* soft matter data, perturbed by aleatoric noise. In Section 4, we generalize this concept towards transversely isotropic Bayesian constitutive neural networks and discover the best model, parameters, and uncertainties to explain *experimental* biaxial stress-stretch data of healthy and diseased human tissues. We compare our observations, discuss our results, and summarize our conclusions in Section 5.

2. Bayesian constitutive neural networks

In the following, we briefly summarize the concept of Bayesian inference [2], discuss its computational realization using variational Bayesian inference [10], and derive the loss function for Bayesian constitutive neural networks towards automated model discovery [21].

Bayesian inference. Bayesian inference is a statistical method that updates the probability for a hypothesis as more information becomes available using Bayes' theorem. Bayes' theorem states that the posterior probability is equal to the likelihood times the prior probability, divided by the marginal likelihood or evidence,

$$\mathsf{P}(\boldsymbol{\theta}\,|\,\hat{\boldsymbol{P}}) = \frac{\mathsf{P}(\hat{\boldsymbol{P}}\,|\,\boldsymbol{\theta})\,\mathsf{P}(\boldsymbol{\theta})}{\mathsf{P}(\hat{\boldsymbol{P}})}\,. \tag{1}$$

Here $\mathsf{P}(\hat{\boldsymbol{P}}\,|\,\boldsymbol{\theta})$ is the *likelihood function*, in our case the conditional probability of the measured stresses $\hat{\boldsymbol{P}}$ for given fixed parameters $\boldsymbol{\theta}$; $\mathsf{P}(\boldsymbol{\theta})$ is the *prior probability distribution* of the model parameters $\boldsymbol{\theta}$; $\mathsf{P}(\hat{\boldsymbol{P}})$ is the *marginal likelihood* or *evidence*; and $\mathsf{P}(\boldsymbol{\theta}\,|\,\hat{\boldsymbol{P}})$ is the *posterior probability distribution*, the conditional probability of the parameters $\boldsymbol{\theta}$ for given measured stresses $\hat{\boldsymbol{P}}$. We can use Bayes' theorem to infer the posterior probability distribution, and with it the model, parameters, and uncertainty that best explain given data, in our case, labeled stress-stretch pairs.

Likelihood. The *likelihood* function $\mathsf{L}(\boldsymbol{\theta};\hat{\boldsymbol{P}}) = \mathsf{P}(\hat{\boldsymbol{P}}\,|\,\boldsymbol{\theta})$ quantifies the chance that some calculated parameters $\boldsymbol{\theta}$ explain the measured stress $\hat{\boldsymbol{P}}$. It measures the goodness of fit between the observed stress-stretch data $\hat{\boldsymbol{P}}_i(\boldsymbol{F}_i)$ and the model output $\boldsymbol{P}(\boldsymbol{\theta},\boldsymbol{F}_i)$, the discovered stress-stretch model of the neural network for the learnt parameters $\boldsymbol{\theta}$, at a fixed deformation gradient $\boldsymbol{F}_i$. The overall likelihood $\mathsf{P}(\hat{\boldsymbol{P}}\,|\,\boldsymbol{\theta})$ is the product of $i = 1,\ldots,\mathrm{n}$ individual likelihood functions $\mathsf{p}(\hat{\boldsymbol{P}}_i\,|\,\boldsymbol{\theta})$, one for each deformation state $\boldsymbol{F}_i$. A common choice for $\mathsf{p}(\hat{\boldsymbol{P}}_i\,|\,\boldsymbol{\theta})$ is the normal distribution for which the measurements are centered around the mean μ_i at the hidden real values $\hat{\boldsymbol{P}}_i$, with standard deviations σ_i that account for measurement errors, i.e.,

$$\mathsf{L}(\boldsymbol{\theta};\hat{\boldsymbol{P}}) = \mathsf{P}(\hat{\boldsymbol{P}}\,|\,\boldsymbol{\theta}) = \textstyle\prod_{i=1}^{\mathrm{n}} \mathsf{p}(\boldsymbol{\theta},\boldsymbol{F}_i) \qquad \text{with} \qquad \mathsf{p}(\boldsymbol{\theta},\boldsymbol{F}_i) = \frac{1}{\sqrt{2\pi}\,\sigma_i}\exp\left(-\frac{(\boldsymbol{P}(\boldsymbol{\theta},\boldsymbol{F}_i)-\hat{\boldsymbol{P}}_i)^2}{2\sigma_i^2}\right). \tag{2}$$

A well-known disadvantage of the likelihood function (2) is that it involves the product of $i = 1,\ldots,\mathrm{n}$ probabilities, which can become extremely small, especially when dealing with large datasets. Computationally, too small likelihoods can result in numerical instabilities as their product becomes too close to zero. Many machine learning tools avoid the product of probabilities in the likelihood function (2) by using the natural logarithm of the likelihood, the *log-likelihood* function $\mathsf{LL}(\boldsymbol{\theta};\hat{\boldsymbol{P}})$ instead,

$$\mathsf{LL}(\boldsymbol{\theta};\hat{\boldsymbol{P}}) = \log(\mathsf{L}(\boldsymbol{\theta};\hat{\boldsymbol{P}})) = \log(\mathsf{P}(\hat{\boldsymbol{P}}\,|\,\boldsymbol{\theta})) = \log(\textstyle\prod_{i=1}^{\mathrm{n}} \mathsf{p}(\hat{\boldsymbol{P}}_{\mathrm{i}}\,|\,\boldsymbol{\theta})) = \sum_{i=1}^{\mathrm{n}} \log(\mathsf{p}(\hat{\boldsymbol{P}}_{\mathrm{i}}\,|\,\boldsymbol{\theta})\,. \tag{3}$$

The log-transformation translates the product of probabilities into a sum of log-probabilities and helps maintain computational precision and numerical stability. Computationally, it often proves convenient to convert the log-likelihood function (3) into the *negative log-likelihood* function $\mathsf{NLL}(\boldsymbol{\theta};\hat{\boldsymbol{P}})$,

$$\mathsf{NLL}(\boldsymbol{\theta};\hat{\boldsymbol{P}}) = -\mathsf{LL}(\boldsymbol{\theta};\hat{\boldsymbol{P}}) = -\textstyle\sum_{i=1}^{\mathrm{n}} \log(\mathsf{p}(\hat{\boldsymbol{P}}_{\mathrm{i}}\,|\,\boldsymbol{\theta}))\,, \tag{4}$$

and minimize the negative log-likelihood NLL rather than maximize the log-likelihood LL.

Maximum likelihood estimate. The parameter values that maximize the log-likelihood function $\mathsf{LL}(\boldsymbol{\theta};\hat{\boldsymbol{P}})$ from equation (3) are the *maximum likelihood estimate* of the true parameter values $\boldsymbol{\theta}$,

$$\boldsymbol{\theta}_{\mathsf{MLE}} = \arg\max_{\boldsymbol{\theta}}(\mathsf{LL}(\boldsymbol{\theta};\hat{\boldsymbol{P}})) = \arg\max_{\boldsymbol{\theta}} \textstyle\sum_{i=1}^{\mathrm{n}} \log(\mathsf{p}(\hat{\boldsymbol{P}}_{\mathrm{i}}\,|\,\boldsymbol{\theta}))\,. \tag{5}$$

Intuitively, the maximum likelihood estimate $\boldsymbol{\theta}_{\mathsf{MLE}}$ defines the set of parameters values that make the observed data, in our case the measured stresses $\hat{\boldsymbol{P}}$, most probable. Maximizing the log-likelihood $\mathsf{LL}(\boldsymbol{\theta};\hat{\boldsymbol{P}})$ from equation (3) is equivalent to minimizing the negative log-likelihood $\mathsf{NLL}(\boldsymbol{\theta};\hat{\boldsymbol{P}})$ from equation (4). This implies that the parameter vector that *maximizes the log-likelihood* $\boldsymbol{\theta}_{\mathsf{MLE}}$ is also the parameter vector that *minimizes the negative-log-likelihood*,

$$\boldsymbol{\theta}_{\mathsf{MLE}} = \arg\min_{\boldsymbol{\theta}}(\mathsf{NLL}(\boldsymbol{\theta};\hat{\boldsymbol{P}})) = \arg\min_{\boldsymbol{\theta}} \textstyle\sum_{i=1}^{\mathrm{n}} -\log(\mathsf{p}(\hat{\boldsymbol{P}}_{\mathrm{i}}\,|\,\boldsymbol{\theta}))\,. \tag{6}$$

Many machine learning tools have built-in algorithms for minimization, and favor the definition (6) over (5).

Priors. The *prior probability distribution* $\mathsf{P}(\boldsymbol{\theta})$ represents our initial belief about the distribution of the parameters before we have observed any data. In our case, the model parameters are the weights of the neural network, $\boldsymbol{\theta} = \{\boldsymbol{w},\boldsymbol{w}^*\}$. We can distinguish two types of network weights, the *external* weights $\boldsymbol{w}$ out of the last hidden layer and the *internal* weights $\boldsymbol{w}^*$ between the hidden layers. For each parameter we would like to infer, we have to select a prior probability distribution $\mathsf{p}_i(w_i, w_i^*)$ and the total prior probability density $\mathsf{P}(\boldsymbol{\theta})$ becomes the product of all $i = 1,\ldots,\mathrm{n}$ individual distributions,

$$\mathsf{P}(\boldsymbol{\theta}) = \textstyle\prod_{i=1}^{\mathrm{n}} \mathsf{p}_i(w_i, w_i^*)\,. \tag{7}$$

To increase the robustness of our model discovery, we only apply prior probability distributions to the external weights $\boldsymbol{w} = \{ w_i \}$, while keeping the internal weights $\boldsymbol{w}^* = \{ w_i^* \}$ deterministic. The choice of the individual distributions p_i is based on our prior domain knowledge or simply on mathematical convenience. For example, we could choose Gaussian distributions with probability densities

$$\mathsf{p}_i = \mathcal{N}(\mu_i, \sigma_i) \qquad \text{with} \qquad \mathsf{p}_i = \frac{1}{\sqrt{2\pi}\,\sigma_i)} \exp\left(-\frac{(w_i - \mu_i)^2}{2\sigma_i^2} \right), \tag{8}$$

with fixed means μ_i and standard deviations σ_i, or uniform distributions $\mathsf{p}_i = 1 \,/\, (w_{i,\max} - w_{i,\min})$ with fixed upper and lower bounds $w_{i,\max}$ and $w_{i,\min}$. In Bayesian statistics, we use the prior $\mathsf{P}(\boldsymbol{\theta})$ to update the beliefs about the parameters $\boldsymbol{\theta}$ after observing the data $\hat{\boldsymbol{P}}(\boldsymbol{F}_i)$ using Bayes' theorem (1). Importantly, while the priors can take various forms, such as normal or uniform, they are *fixed* distributions that represent our *initial* beliefs about the parameters.

Marginal likelihood. The *marginal likelihood* function $\mathsf{P}(\hat{\boldsymbol{P}})$ is also often referred to as evidence. It represents a likelihood function in which the parameter variables are marginalized by integrating over the entire parameter domain,

$$\mathsf{P}(\hat{\boldsymbol{P}}) = \int_{\boldsymbol{\theta}} \mathsf{P}(\hat{\boldsymbol{P}}, \boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\theta}\,. \tag{9}$$

Marginal likelihoods are generally difficult if not impossible to compute. If needed, we could integrate equation (9) using numerical integration schemes such as Gaussian integration or Monte Carlo methods. Fortunately, for most practical purposes, we do not need to know the precise value of the marginal likelihood $\mathsf{P}(\hat{\boldsymbol{P}})$. Instead of evaluating *absolute* probabilities using Bayes' theorem (1), we typically focus on *relative* probabilities using the ratio of two posterior probabilities, $\mathsf{P}_1\,(\,\boldsymbol{\theta}_1 \,|\, \hat{\boldsymbol{P}}\,)/\mathsf{P}_2\,(\,\boldsymbol{\theta}_2 \,|\, \hat{\boldsymbol{P}}\,)$. Since the marginal likelihood is a normalizing constant, it cancels out when computing relative probabilities. For model comparison and parameter estimation tasks, we often reduce Bayes' theorem (1) to the fact that the posterior distribution $\mathsf{P}\,(\,\boldsymbol{\theta} \,|\, \hat{\boldsymbol{P}}\,)$ is proportional to the product of the likelihood and the prior, i.e.,

$$\mathsf{P}\,(\,\boldsymbol{\theta} \,|\, \hat{\boldsymbol{P}}\,) \propto \mathsf{P}\,(\,\hat{\boldsymbol{P}} \,|\, \boldsymbol{\theta}\,)\; \mathsf{P}\,(\,\boldsymbol{\theta}\,)\,. \tag{10}$$

For most practical purposes, knowing this proportionality (10) is sufficient and we do not need to know the exact value of the marginal likelihood $\mathsf{P}\,(\,\hat{\boldsymbol{P}}\,)$.

Posterior. The *posterior probability distribution* $\mathsf{P}(\boldsymbol{\theta} \,|\, \hat{\boldsymbol{P}})$ is the conditional probability of the parameters $\boldsymbol{\theta}$ for the given data $\hat{\boldsymbol{P}}$, the experimentally measured stress. We can calculate the posterior probability from the likelihood $\mathsf{P}(\hat{\boldsymbol{P}} \,|\, \boldsymbol{\theta})$ and our prior knowledge encoded through the prior probability distribution $\mathsf{P}(\boldsymbol{\theta})$, using Bayes' theorem (1),

$$\mathsf{P}\,(\,\boldsymbol{\theta} \,|\, \hat{\boldsymbol{P}}\,) = \frac{\mathsf{P}\,(\,\hat{\boldsymbol{P}} \,|\, \boldsymbol{\theta}\,)\; \mathsf{P}\,(\,\boldsymbol{\theta}\,)}{\mathsf{P}(\,\hat{\boldsymbol{P}}\,)} \propto \mathsf{P}\,(\,\hat{\boldsymbol{P}} \,|\, \boldsymbol{\theta}\,)\; \mathsf{P}\,(\,\boldsymbol{\theta}\,)\,. \tag{11}$$

The parameters $\boldsymbol{\theta}$ that maximize the posterior probability distribution $\mathsf{P}\,(\,\boldsymbol{\theta} \,|\, \hat{\boldsymbol{P}}\,)$ are called the *maximum a posteriori estimate* $\boldsymbol{\theta}_{\mathsf{MAP}}$ of the true parameter values $\boldsymbol{\theta}$, and their values are independent of the marginal likelihood $\mathsf{P}(\,\hat{\boldsymbol{P}}\,)$,

$$\boldsymbol{\theta}_{\mathsf{MAP}} = \arg\max_{\boldsymbol{\theta}}\, (\mathsf{P}\,(\,\boldsymbol{\theta} \,|\, \hat{\boldsymbol{P}}\,)) = \arg\max_{\boldsymbol{\theta}}\, (\mathsf{P}\,(\,\hat{\boldsymbol{P}} \,|\, \boldsymbol{\theta}\,)\; \mathsf{P}\,(\,\boldsymbol{\theta}\,))\,. \tag{12}$$

A common strategy to estimate the posterior distributions (11) and the maximum a posteriori estimate (12) is multi-chain full-batch Hamiltonian Monte Carlo, a highly efficient and well-studied Markov Chain Monte Carlo method. Theoretically, Hamiltonian Monte Carlo is guaranteed to produce samples from the true posterior asymptotically. In practise, applying Hamiltonian Monte Carlo to state-of-the-art neural networks is extremely challenging due to its high computational cost: It can take tens of thousands of training epochs to produce a single sample from the posterior.

Variational Bayesian inference. Variational Bayesian inference has become a popular method to speed-up computation when estimating complex probability distributions in the context of Bayesian statistics. Its underlying idea is to *approximate complex posterior distributions* with a much simpler distribution, the *variational distribution*. Here

we approximate the posterior probability distribution $\mathsf{P}(\boldsymbol{\theta} \,|\, \hat{\boldsymbol{P}})$ of the unknown parameter vector $\boldsymbol{\theta} = \{\boldsymbol{w}, \boldsymbol{w}^*\}$ by a less flexible family of distributions, the variational distribution $\mathsf{Q}(\boldsymbol{\theta}; \boldsymbol{W})$ of the parameter vector $\boldsymbol{W} = \{\boldsymbol{w}, \boldsymbol{w}^*, \boldsymbol{w}_\mu, \boldsymbol{w}_\sigma\}$. This new parameter vector $\boldsymbol{W}$ consists of two types of parameters, the *network weights* $\boldsymbol{w}$ and $\boldsymbol{w}^*$, and the *variational parameters* $\boldsymbol{w}_\mu$ and $\boldsymbol{w}_\sigma$. We follow the standard approach [3] and assume that the variational approximation $\mathsf{Q}(\boldsymbol{\theta}; \boldsymbol{W})$ adopts a parameterized Gaussian distribution,

$$\mathsf{Q}(\boldsymbol{\theta}; \boldsymbol{W}) = \prod_{i=1}^{\mathrm{n}} \mathsf{q}_i\,(\boldsymbol{\theta}; w_i, w_i^*, w_{\mu,i}, w_{\sigma,i})\,, \tag{13}$$

as product of $i = 1, \ldots, \mathrm{n}$ one-dimensional Gaussian distributions q_i. Each individual Gaussian distribution is the product of the normal distribution, $\mathcal{N}(w_{\mu,i}, w_{\sigma,i})$, in terms of the mean and standard deviation $w_{\mu,i}$ and $w_{\sigma,i}$, and the neural network activation function $a_i(w_i,\, w_i^*)$, in terms of the external and internal weights w_i and w_i^*,

$$\mathsf{q}_i = \mathcal{N}(w_{\mu,i}, w_{\sigma,i})\; a_i\,(w_i,\, w_i^*)\,. \tag{14}$$

Importantly, instead of using *fixed* means and standard deviations μ_i and σ_i as in equation (8), variational inferences treats the means and standard deviations $w_{\mu,i}$ and $w_{\sigma,i}$ as *trainable* variational parameters. Yet, here, instead of using the standard deviation $w_{\sigma,i}$ directly, we subject it to the exponential linear unit function $\mathsf{elu}(w_{\sigma,i})$, with $\mathsf{elu}(\circ) = (\circ)$ for non-negative arguments $(\circ) \geq 0$, and $\mathsf{elu}(\circ) = \alpha(\exp((\circ) - 1)$ for negative arguments $(\circ) < 0$. Using this modified distribution, $\mathsf{q}_i = \mathcal{N}(w_{\mu,i}, \mathsf{elu}(w_{\sigma,i}))\; a_i\,(w_i,\, w_i^*)$, instead of equation (14) manages negative values by pushing them closer to zero and speeds up learning by bringing the normal gradient closer to the unit natural gradient [4]. Now, the objective is to find the optimal parameters $\boldsymbol{W}$, such that the variational approximation $\mathsf{Q}(\boldsymbol{\theta}; \boldsymbol{W})$ is as close to the true posterior distribution $\mathsf{P}(\boldsymbol{\theta} \,|\, \hat{\boldsymbol{P}})$ as possible. We measure the divergence between $\mathsf{Q}(\boldsymbol{\theta}; \boldsymbol{W})$ and $\mathsf{P}(\boldsymbol{\theta} \,|\, \hat{\boldsymbol{P}})$ using the *Kullbach-Leibler divergence*,

$$\mathsf{KL}\,[\,\mathsf{Q}(\boldsymbol{W}) \,\|\, \mathsf{P}(\boldsymbol{\theta} \,|\, \hat{\boldsymbol{P}})\,] = \int \mathsf{Q}(\boldsymbol{W})\, \log \frac{\mathsf{Q}(\boldsymbol{W})}{\mathsf{P}(\boldsymbol{\theta} \,|\, \hat{\boldsymbol{P}})}\, \mathrm{d}\boldsymbol{\theta}\,. \tag{15}$$

With the definition of the conditional probability, $\mathsf{P}(\boldsymbol{\theta} \,|\, \hat{\boldsymbol{P}}) = \mathsf{P}(\boldsymbol{\theta}, \hat{\boldsymbol{P}})/\mathsf{P}(\hat{\boldsymbol{P}})$, and the calculus rules of the logarithm, $\log(A \cdot B) = \log(A) + \log(B)$ and $\log(A/B) = -\log(B/A)$, this definition becomes

$$\mathsf{KL}\,[\,\mathsf{Q}(\boldsymbol{W}) \,\|\, \mathsf{P}(\boldsymbol{\theta} \,|\, \hat{\boldsymbol{P}})\,] = \int \mathsf{Q}(\boldsymbol{W}) \log\mathsf{P}(\hat{\boldsymbol{P}})\, \mathrm{d}\boldsymbol{\theta} - \int \mathsf{Q}(\boldsymbol{W}) \log \frac{\mathsf{P}(\boldsymbol{\theta}, \hat{\boldsymbol{P}})}{\mathsf{Q}(\boldsymbol{W})}\, \mathrm{d}\boldsymbol{\theta}\,. \tag{16}$$

Since the probability of the data $\mathsf{P}(\hat{\boldsymbol{P}})$ is independent of the parameters $\boldsymbol{\theta}$, the first integral simplifies to $\int \mathsf{Q}(\boldsymbol{W}) \log(\mathsf{P}(\hat{\boldsymbol{P}}))\; \mathrm{d}\boldsymbol{\theta} = \log(\mathsf{P}(\hat{\boldsymbol{P}}))\; \int \mathsf{Q}(\boldsymbol{W})\, \mathrm{d}\boldsymbol{\theta}$, where the integral for all probability densities $\mathsf{Q}(\boldsymbol{W})$ is identical to one, $\int \mathsf{Q}(\boldsymbol{W})\, \mathrm{d}\boldsymbol{\theta} = 1$, and the Kullbach-Leibler divergence (15) is identical to the following expression,

$$\mathsf{KL}\,[\,\mathsf{Q}(\boldsymbol{W}) \,\|\, \mathsf{P}(\boldsymbol{\theta} \,|\, \hat{\boldsymbol{P}})\,] = \log(\mathsf{P}(\hat{\boldsymbol{P}}))\; - \int \mathsf{Q}(\boldsymbol{W})\, \log \frac{\mathsf{P}(\boldsymbol{\theta}, \hat{\boldsymbol{P}})}{\mathsf{Q}(\boldsymbol{W})}\, \mathrm{d}\boldsymbol{\theta}\,. \tag{17}$$

Minimum Kullbach-Leibler divergence. The objective of the variational inference is to find the parameters $\boldsymbol{W}$ that minimize the Kullbach-Leibler divergence (17). Since the probability of the data $\mathsf{P}(\hat{\boldsymbol{P}})$ is independent of the parameters $\boldsymbol{W}$, we only need to minimize the second term,

$$\boldsymbol{W}^* = \arg\min_{\boldsymbol{W}} \left(- \int \mathsf{Q}(\boldsymbol{W})\, \log \frac{\mathsf{P}(\boldsymbol{\theta}, \hat{\boldsymbol{P}})}{\mathsf{Q}(\boldsymbol{W})}\, \mathrm{d}\boldsymbol{\theta} \right). \tag{18}$$

We now eliminate the unknown posterior probability using the definition of the conditional probability, $\mathsf{P}(\boldsymbol{\theta}, \hat{\boldsymbol{P}}) = \mathsf{P}(\boldsymbol{\theta} \,|\, \hat{\boldsymbol{P}}) \cdot \mathsf{P}(\hat{\boldsymbol{P}})$, and apply the calculus rules of the logarithm,

$$\boldsymbol{W}^* = \arg\min_{\boldsymbol{W}} \left(\int \mathsf{Q}(\boldsymbol{W})\, (-\log(\mathsf{P}(\,\hat{\boldsymbol{P}} \,|\, \boldsymbol{\theta}\,)))\, \mathrm{d}\boldsymbol{\theta} + \int \mathsf{Q}(\boldsymbol{W})\, \log \frac{\mathsf{Q}(\boldsymbol{W})}{\mathsf{P}(\boldsymbol{\theta})}\, \mathrm{d}\boldsymbol{\theta} \right). \tag{19}$$

The first term is the expected negative log-likelihood, $\mathsf{NLL}(\boldsymbol{\theta}; \hat{\boldsymbol{P}}) = -\log(\mathsf{P}(\,\hat{\boldsymbol{P}} \,|\, \boldsymbol{\theta}))$, given the parameters $\boldsymbol{\theta}$ distributed according to the variational distribution Q, and the second term is the Kullbach-Leibler divergence, $\mathsf{KL}\,[\,\mathsf{Q}(\boldsymbol{W}) \,\|\, \mathsf{P}(\boldsymbol{\theta})\,]$,

$$\boldsymbol{W}^* = \arg\min_{\boldsymbol{W}} \left(\mathbb{E}_{\boldsymbol{\theta} \sim \mathsf{Q}}[\, -\log(\mathsf{P}(\hat{\boldsymbol{P}} \,|\, \boldsymbol{\theta}))\,] + \mathsf{KL}\,[\,\mathsf{Q}(\boldsymbol{W}) \,\|\, \mathsf{P}(\boldsymbol{\theta})\,] \right). \tag{20}$$

We now formulate the loss function for our Bayesian constitutive neural network motivated by equation (20).

Loss function. We train our Bayesian constitutive neural network by minimizing a three-term loss function L to learn the network weights $\boldsymbol{w}$ and $\boldsymbol{w}^*$ and the variational parameters $\boldsymbol{w}_\mu$ and $\boldsymbol{w}_\sigma$ [3], i.e.,

$$L = L_{\mathsf{NLL}} + L_{\mathsf{KL}} + L_1 \rightarrow \min_{\boldsymbol{W}} \,. \tag{21}$$

The first term represents the expected negative log-likelihood NNL according to equation (4) given the parameters $\boldsymbol{\theta}$ with the variational distribution Q,

$$L_{\mathsf{NLL}} = \mathbb{E}_{\boldsymbol{\theta}\sim\mathsf{Q}}\,[\,-\log(\mathsf{P}(\hat{\boldsymbol{P}}\,|\,\boldsymbol{\theta}))\,]\,. \tag{22}$$

The second term is the Kullback-Leibler divergence between the variational approximation $\mathsf{Q}(\boldsymbol{\theta};\boldsymbol{W})$ that approximates the parameters $\boldsymbol{\theta}$ using $\boldsymbol{W}$ and the prior $\mathsf{P}(\boldsymbol{\theta})$. By minimizing this second term, variational learning efficiently approximates the prior distribution $\mathsf{P}(\boldsymbol{\theta})$ when the exact distribution is unknown,

$$L_{\mathsf{KL}} = \mathsf{KL}\,[\,\mathsf{Q}(\boldsymbol{\theta};\boldsymbol{W})\,\|\,\mathsf{P}(\boldsymbol{\theta}|\hat{\boldsymbol{P}})\,]\,. \tag{23}$$

The third term is the L_1 regularization that sparisfies the model by penalizing the α-weighted L_1 norm of the external weights $\boldsymbol{w}$ [41],

$$L_1 = \alpha\,\|\,\boldsymbol{w}\,\|_1 = \alpha\,\textstyle\sum_{i=1}^{n_w}|\,w_i\,|\,. \tag{24}$$

In summary, the loss function (21) is a sum of a data-dependent part associated with the likelihood cost, a prior dependent part associated with the complexity cost, and a regularization-dependent part associated with the sparsity cost [3]. We minimize the loss function to train the network using the ADAM optimizer, a robust adaptive algorithm for gradient-based first-order optimization.

In the following, we demonstrate how to design Bayesian constitutive neural networks for *isotropic* and *transversely isotropic* incompressible hyperelastic materials. For the isotropic case, we discover the best model, parameters, and posterior distributions to explain *synthetic* stress-stretch data of a Mooney-Rivlin material in uniaxial tension, uniaxial compression, equibiaxial tension, and pure shear, perturbed by aleatoric noise. For the transversely isotropic case, we discover the best model, parameters, and posterior distributions to explain *experimental* stress-stretch data of healthy and diseased abdominal aortic tissue from the medial layer and from both media and adventitia layers combined when tested in equibiaxial tension.

3. Isotropic Bayesian constitutive neural networks

The *isotropic* Bayesian constitutive neural network takes the deformation gradient $\boldsymbol{F}$, the gradient of the deformation map $\boldsymbol{\varphi}$ with respect to the coordinates of the undeformed sample $\boldsymbol{X}$, as input,

$$\boldsymbol{F} = \nabla_{\boldsymbol{X}}\boldsymbol{\varphi}\,, \tag{25}$$

and calculates the first and second invariants,

$$I_1 = [\,\boldsymbol{F}^{\mathrm{t}}\cdot\boldsymbol{F}\,]:\boldsymbol{I} \qquad \text{and} \qquad I_2 = \frac{1}{2}\,[I_1^2 - [\,\boldsymbol{F}^{\mathrm{t}}\cdot\boldsymbol{F}\,]:[\,\boldsymbol{F}^{\mathrm{t}}\cdot\boldsymbol{F}\,]]\,. \tag{26}$$

For *perfectly incompressible* materials, the third invariant remains constant and identical to one, $I_3 = J^2 = 1$. The network discovers *hyperelastic* material models that satisfy the second law of thermodynamics, which implies that the Piola stress $\boldsymbol{P}$ is the derivative of the free energy $\psi(\boldsymbol{F})$ with respect to the deformation gradient $\boldsymbol{F}$ modified by a pressure term, $-p\,\boldsymbol{F}^{-\mathrm{t}}$,

$$\boldsymbol{P} = \frac{\partial\psi}{\partial\boldsymbol{F}} - p\,\boldsymbol{F}^{-\mathrm{t}} = \frac{\partial\psi}{\partial I_1}\frac{\partial I_1}{\partial\boldsymbol{F}} + \frac{\partial\psi}{\partial I_2}\frac{\partial I_2}{\partial\boldsymbol{F}} - p\,\boldsymbol{F}^{-\mathrm{t}}\,, \tag{27}$$

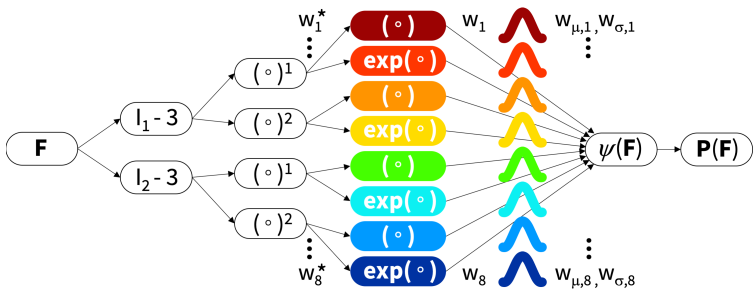

Figure 1: **Isotropic Bayesian constitutive neural network.** The network has two hidden layers to discover the free-energy function $\psi(I_1, I_2)$ as a function of the invariants of the deformation gradient $\boldsymbol{F}$ using eight terms. The first layer generates powers $(\circ)$ and $(\circ)^2$ of the network input, the second layer applies the identity $(\circ)$ and exponential function $(\exp(\circ))$ to these powers, and the network output is the sum of these eight terms, weighted by their probability densities $\mathcal{N}(\boldsymbol{w}_\mu, \boldsymbol{w}_\sigma)$. During training, the network learns the network weights $\boldsymbol{w}$ and $\boldsymbol{w}^*$ and the variational parameters $\boldsymbol{w}_\mu$ and $\boldsymbol{w}_\sigma$.

where the hydrostatic pressure, $p = -\frac{1}{3}\,\boldsymbol{P} : \boldsymbol{F}$, acts as a Lagrange multiplier that we determine from the boundary conditions. We discover the free-energy function ψ using a Bayesian constitutive neural network that takes the deformation gradient $\boldsymbol{F}$ as input and approximates the free-energy function $\psi(\boldsymbol{F})$ as the sum of eight probability-weighted terms. Figure 1 illustrates our neural network with two hidden layers and eight nodes [21]. The first layer generates powers $(\circ)$ and $(\circ)^2$ of the network input, the two invariants I_1 and I_2. The second layer applies the identity, $(\circ)$ and the exponential function $(\exp(\circ))$ to these powers. The network output is the sum of these eight terms, weighted by their probability densities $\mathcal{N}(w_\mu, w_\sigma)$. The free-energy function of this networks takes the following explicit form,

$$\begin{aligned}\psi = {} & \mathcal{N}(w_{\mu,1}, w_{\sigma,1})\; w_1\, w_1^*\; [I_1 - 3] \;\; + \mathcal{N}(w_{\mu,2}, w_{\sigma,2})\; w_2\; [\exp(w_2^*\; [I_1 - 3]\;\;) - 1] \\ & + \mathcal{N}(w_{\mu,3}, w_{\sigma,3})\; w_3\, w_3^*\; [I_1 - 3]^2 + \mathcal{N}(w_{\mu,4}, w_{\sigma,4})\; w_4\; [\exp(w_4^*\; [I_1 - 3]^2) - 1] \\ & + \mathcal{N}(w_{\mu,5}, w_{\sigma,5})\; w_5\, w_5^*\; [I_2 - 3] \;\; + \mathcal{N}(w_{\mu,6}, w_{\sigma,6})\; w_6\; [\exp(w_6^*\; [I_2 - 3]\;\;) - 1] \\ & + \mathcal{N}(w_{\mu,7}, w_{\sigma,7})\; w_7\, w_7^*\; [I_2 - 3]^2 + \mathcal{N}(w_{\mu,8}, w_{\sigma,8})\; w_8\; [\exp(w_8^*\; [I_2 - 3]^2) - 1]\,,\end{aligned} \tag{28}$$

corrected by the pressure term, $\psi = \psi - p\,[J - 1]$. To complete the definition of the Piola stress in equation (27), we take its derivatives with respect to the two invariants,

$$\begin{aligned}\frac{\partial\psi}{\partial I_1} = {} & \mathcal{N}(w_{\mu,1}, w_{\sigma,1})\; w_1\, w_1^* \;\; + \mathcal{N}(w_{\mu,2}, w_{\sigma,2})\; w_2\, w_2^*\; \exp(\, w_2^*\; [\, I_1 - 3\,]\;\;) \\ & + 2\,[\, I_1 - 3\,][\; \mathcal{N}(w_{\mu,3}, w_{\sigma,3})\; w_3\, w_3^* \;\; + \mathcal{N}(w_{\mu,4}, w_{\sigma,4})\; w_4\, w_4^*\; \exp(\, w_4^*\; [\, I_1 - 3\,]^2)\,, \\ \frac{\partial\psi}{\partial I_2} = {} & \mathcal{N}(w_{\mu,5}, w_{\sigma,5})\; w_5\, w_5^* \;\; + \mathcal{N}(w_{\mu,6}, w_{\sigma,6})\; w_6\, w_6^*\; \exp(\, w_6^*\; [\, I_2 - 3\,]\;\;) \\ & + 2\,[\, I_2 - 3\,][\; \mathcal{N}(w_{\mu,7}, w_{\sigma,7})\; w_7\, w_7^* \;\; + \mathcal{N}(w_{\mu,8}, w_{\sigma,8})\; w_8\, w_8^*\; \exp(\, w_8^*\; [\, I_2 - 3\,]^2)\,.\end{aligned} \tag{29}$$

The network has two sets of network weights, $\boldsymbol{w}$ and $\boldsymbol{w}^*$, associated the eight terms of the free-energy function, and two sets of variational parameters, $\boldsymbol{w}_\mu$ and $\boldsymbol{w}_\sigma$, associated with the means and standard deviations of these eight terms. We learn these 32 weights by minimizing the three-term loss function $L = L_{\text{NLL}} + L_{\text{KL}} + L_1$ in equation (21).We consider data from homogeneous uniaxial compression, uniaxial tension, equibiaxial tension, and pure shear test.

For the cases of *uniaxial compression*, and *uniaxial tension*, with a stretch λ in the 1-direction, such that $I_1 = \lambda^2 + 2\lambda^{-1}$ and $I_2 = 2\lambda + \lambda^{-2}$ and $\mathbf{F} = \text{diag}\,\{\,\lambda, \lambda^{-1/2}, \lambda^{-1/2}\,\}$ and $\mathbf{P} = \text{diag}\,\{\,P_{11}, 0, 0\,\}$, the stress-stretch relation for isotropic materials [22] is

$$P_{11} = 2\left[\frac{\partial\psi}{\partial I_1} + \frac{1}{\lambda}\frac{\partial\psi}{\partial I_2}\right]\left[\lambda - \frac{1}{\lambda^2}\right]. \tag{30}$$

For the case of *equibiaxial tension*, with a stretch λ in the 1- and 2-directions, such that $I_1 = 2\lambda^2 + \lambda^{-4}$ and $I_2 = \lambda^4 + 2\lambda^{-2}$ and $\mathbf{F} = \text{diag}\,\{\,\lambda, \lambda, \lambda^{-2}\,\}$ and $\mathbf{P} = \text{diag}\,\{\,P_{11}, P_{22}, 0\,\}$, the stress-stretch relation for isotropic materials [21] is

$$P_{11} = 2\left[\frac{\partial\psi}{\partial I_1} + \lambda^2\frac{\partial\psi}{\partial I_2}\right]\left[\lambda - \frac{1}{\lambda^5}\right] = P_{22}\,. \tag{31}$$

For the case of *pure shear* of a long rectangular specimen stretched with λ along its short axis in the 1-direction, and no deformation along it long axis in the 2-direction, such that $I_1 = I_2 = \lambda^2 + 1 + \lambda^{-2}$ and $\mathbf{F} = \text{diag}\{\lambda, 1, \lambda^{-1}\}$ and $\mathbf{P} = \text{diag}\{P_{11}, P_{22}, 0\}$, the stress-stretch relations for isotropic materials [21] are

$$P_{11} = 2\left[\frac{\partial\psi}{\partial I_1} + \frac{\partial\psi}{\partial I_2}\right]\left[\lambda - \frac{1}{\lambda^3}\right] \qquad \text{and} \qquad P_{22} = 2\left[\frac{\partial\psi}{\partial I_1} + \lambda^2\frac{\partial\psi}{\partial I_2}\right]\left[1 - \frac{1}{\lambda^2}\right]. \tag{32}$$

We explore the performance of the isotropic Bayesian neural network from Figure 1 on the basis of synthetic stress-stretch data of a Mooney-Rivlin material [29, 38] perturbed by aleatoric noise. The strain energy function of the Mooney-Rivlin model is $\psi = \frac{1}{2}\mu_1[I_1 - 3] + \frac{1}{2}\mu_2[I_2 - 3]$, where μ_1 and μ_2 denote the two model parameters. Following equations (30), (31) and (32), we obtain the explicit stress-stretch relationships for uniaxial compression and tension, $P_{11} = [\mu_1 + \lambda^{-1}\mu_2][\lambda - \lambda^{-2}]$, for equibiaxial tension, $P_{11} = [\mu_1 + \lambda^2\mu_2][\lambda - \lambda^{-5}]$, and for pure shear, $P_{11} = [\mu_1 + \mu_2][\lambda - \lambda^{-3}]$. To generate the synthetic data, we apply an additive Gaussian aleatoric noise to the stress, $\hat{P}_{11} = P_{11} + \mathcal{N}(\mu, \sigma)$. We select a zero mean, $\mu = 0$, and a standard deviation that scales linearly with the absolute stress, $\sigma = \sigma_0 |P_{11}|$, with an aleatoric scaling coefficient σ_0. We sample stress-stretch data pairs $\{\lambda, \hat{P}_{11}\}$ for all load cases, apply Bayesian model discovery, and evaluate model robustness. To explore the predictive potential of the Bayesian network, we compare two cases, training on *all* data and training on *incomplete* data while testing on the remaining data.

Bayesian networks successfully rediscover the initial model, even in the presence of noise. Figure 2 shows the discovered model for the synthetic data of the Mooney-Rivlin model, $\psi = \frac{1}{2}\mu_1[I_1 - 3] + \frac{1}{2}\mu_2[I_2 - 3]$, with true parameters $\mu_1 = 1\,\text{kPa}$ and $\mu_2 = 1\,\text{kPa}$, an aleatoric scaling coefficient $\sigma_0 = 0.15$, and an L_1 regularization parameter $\alpha = 0.001$, trained on *all* data in the stretch ranges of $\lambda = [0.8, \ldots, 1.0]$ for uniaxial compression, and $\lambda = [1.0, \ldots, 1.3]$ for uniaxial tension, equibiaxial tension, and pure shear. Figure 3 highlights the discovered isotropic model parameters along with their posterior distributions. Both figures confirm that our Bayesian constitutive neural network successfully identifies the mean stress response and the uncertainty in the data. First and foremost, out of $2^8 = 256$ possible combinations of terms, the network robustly rediscovers the Mooney Rivlin model, $\psi = \frac{1}{2}\mu_1[I_1 - 3] + \frac{1}{2}\mu_2[I_2 - 3]$, with mean network weights of $w_1 = 0.782\,\text{kPa}$ and $w_5 = 0.525\,\text{kPa}$, which translate into Mooney Rivlin parameters of $\mu_1 = 2\,w_1 = 1.564\,\text{kPa}$ and $\mu_2 = 2\,w_5 = 1.050\,\text{kPa}$, two meaningful parameters with physical units and a clear physical interpretation [29, 38]. In addition, the network also discovers the posterior distributions of the weights along with their standard deviations, $w_1 = 0.782 \pm 0.02\,\text{kPa}$ and $w_5 = 0.525 \pm 0.01\,\text{kPa}$. Most importantly, the network autonomously trains all inactivate model terms to zero [26], $w_2 = w_3 = w_4 = w_6 = w_7 = w_8 = 0\,\text{kPa}$, which is not straightforward in conventional Bayesian neural networks [12]. Taken together, our Bayesian constitutive neural networks robustly and repeatably discover models, parameters, and uncertainty from synthetic data, even in the presence of noise.

Bayesian networks robustly discover uncertainties in non-training regimes. Figure 4 shows the discovered model for the synthetic data of the Mooney-Rivlin model, $\psi = \frac{1}{2}\mu_1[I_1 - 3] + \frac{1}{2}\mu_2[I_2 - 3]$, with true parameters $\mu_1 = 1\,\text{kPa}$ and $\mu_2 = 1\,\text{kPa}$, an aleatoric scaling coefficient $\sigma_0 = 0.15$, and an L_1 regularization parameter $\alpha = 0.001$, trained on *incomplete* data in the stretch ranges of $\lambda = [0.8, \ldots, 1.0]$ for uniaxial compression, and $\lambda = [1.0, \ldots, 1.15]$ for uniaxial tension, equibiaxial tension, and pure shear, and tested in the unseen stretch rage of $\lambda = [1.15, \ldots, 1.3]$. Figure 5 highlights the discovered isotropic model parameters along with their posterior distributions. Similar to the previous example, our Bayesian constitutive neural network successfully discovers the mean stress response and the uncertainty in the data. However, the direct comparison of the uncertainties in the stress predictions in the blue-shaded areas in Figures 2 and 4 confirms our intuition that training on *all* data in Figure 2 results in smaller uncertainties than training on *incomplete* data in Figure 4. Interestingly, the extrapolation into the unseen regime of $1.15 < \lambda \leq 1.3$ is better in uniaxial tension and pure shear than in equibiaxial tension, where the uncertainties increase dramatically as the stretch increases. Nonetheless, the network is still able to discover the initial Mooney Rivlin model [29, 38], $\psi = \frac{1}{2}\mu_1[I_1 - 3] + \frac{1}{2}a[\exp(w_2^*[I_2 - 3]) - 1]/b + \frac{1}{2}\mu_2[I_2 - 3]$, but now with an additional exponential linear first invariant Demiray-type term [5], $\frac{1}{2}a[\exp(w_2^*[I_2 - 3]) - 1]/b$. The discovered mean network weights of $w_1 = 0.824\,\text{kPa}$, $w_2 = 0.054\,\text{kPa}$ and $w_5 = 0.468\,\text{kPa}$ translate into Mooney Rivlin stiffnesses of $\mu_1 = 2\,w_1 = 1.648\,\text{kPa}$ and $\mu_2 = 2\,w_5 = 0.936\,\text{kPa}$, a stiffness like parameter of $a = 2\,w_2\,w_2 = 1.42\,\text{kPa}$, and a nonlinearity parameter of $b = w_2^* = 2.58$, four meaningful parameters with physical units and a clear physical interpretation. In addition, the network also discovers the posterior distributions of the weights along with their

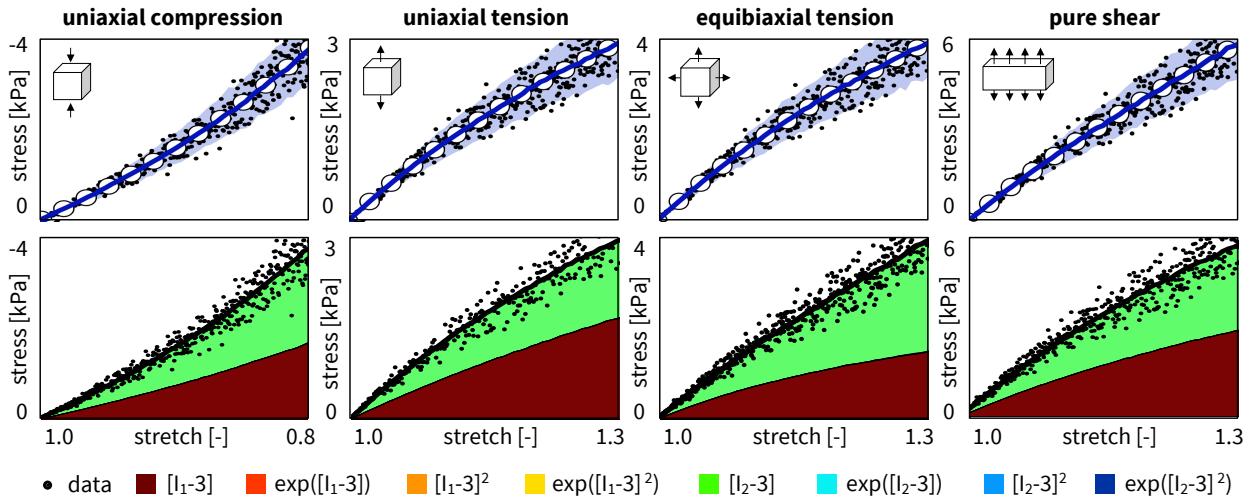

Figure 2: **Synthetic data and discovered isotropic model for training on all data.** Nominal stresses P as functions of the stretches λ for the isotropic, perfectly incompressible Bayesian constitutive neural network with two hidden layers, eight nodes, and eight prior distributions from Figure 1. We used all compressive data up to $\lambda = 0.8$, and all tensile, biaxial, and shear data up to $\lambda = 1.3$ for training. Dots illustrate the synthetic uniaxial compression, uniaxial tension, equibiaxial tension, and pure shear data of the Mooney Rivlin model with Gaussian noise; solid blue curves and blue-shaded areas indicate the mean predicted stresses ± standard deviations; color-coded areas highlight the eight possible contributions to the discovered stress function.

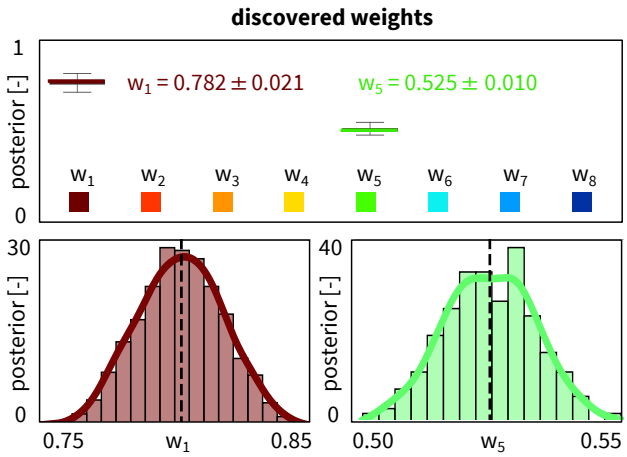

Figure 3: **Posterior distributions of discovered isotropic model parameters for training on all data.** When using all data for training, the isotropic, perfectly incompressible Bayesian constitutive neural network robustly rediscovers the Gaussian noise perturbed Mooney Rivlin model, $\psi = w_1\,[\,I_1 - 3\,] + w_5\,[\,I_2 - 3\,]$, with discovered network weights of $w_1 = 0.782 \pm 0.02$ and $w_5 = 0.525 \pm 0.01$, while all other network weights correctly train to zero.

standard deviations, $w_1 = 0.054 \pm 0.01$ kPa, $w_2 = 0.054 \pm 0.01$ kPa, and $w_5 = 0.468 \pm 0.00$ kPa. We observe that the non-Mooney Rivlin weight $w_2 = 0.054$ is an order of magnitude smaller than the two Mooney Rivlin weights $w_1 = 0.054$ and $w_5 = 0.468$. We conclude that the influence of the additional exponential linear first invariant Demiray-type term [5], $w_2\,[\,\exp(w_2^*[I_2 - 3]) - 1\,]$ is relatively small, and that the discovered model remains dominated by the two Mooney Rivlin terms [29, 38], $w_1\,[\,I_1 - 3\,]$ and $w_5\,[\,I_2 - 3\,]$. The small narrow red band of the exponential linear first invariant contribution to the overall stress in Figure 4 confirms this observation. Importantly, even for incomplete training data, the network autonomously identifies the true non-zero model terms w_1 and w_5, while training all but the additional small w_2 term to zero, $w_3 = w_4 = w_6 = w_7 = w_8 = 0$ kPa. We have confirmed, although not explicitly shown here, that we can explicitly modulate the sparsification of the parameter vector $\boldsymbol{w}$ through the penalty parameter α [11] to fine-tune the trade-off between model accuracy and model interpretability [7]. Especially

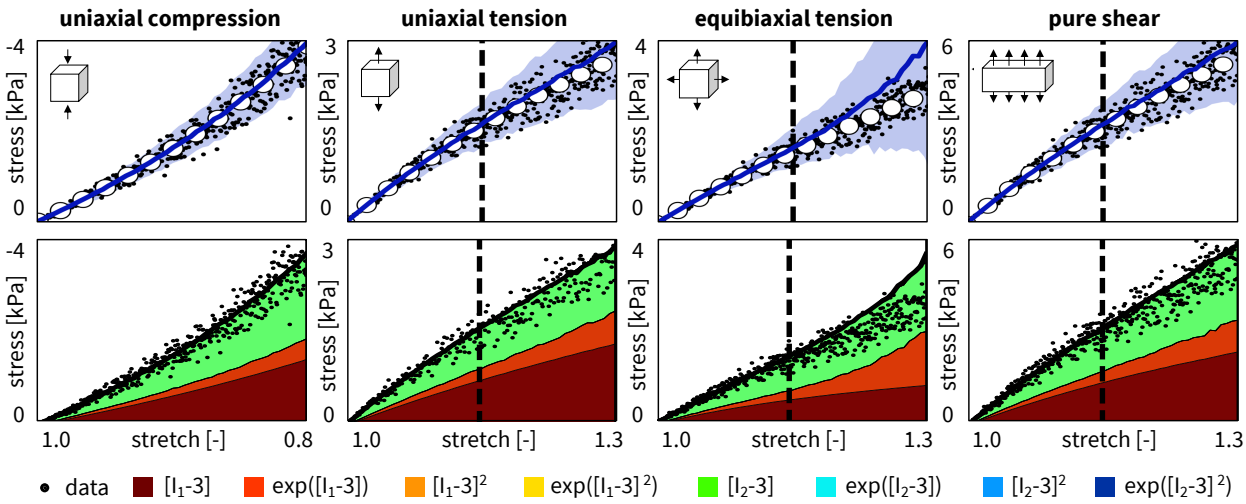

Figure 4: **Synthetic data and discovered isotropic model for training on incomplete data.** Nominal stresses P as functions of the stretches λ for the isotropic, perfectly incompressible Bayesian constitutive neural network with two hidden layers, eight nodes, and eight prior distributions from Figure 1. We used all compressive data up to $\lambda = 0.8$ for training, and tensile, biaxial, and shear data up to $\lambda = 1.15$ for training, and up to $\lambda = 1.3$ for testing. Dots illustrate the synthetic uniaxial compression, uniaxial tension, equibiaxial tension, and pure shear data of the Mooney Rivlin model with Gaussian noise; solid blue curves and blue-shaded areas indicate the mean predicted stresses ± standard deviations; color-coded areas highlight the eight possible contributions to the discovered stress function.

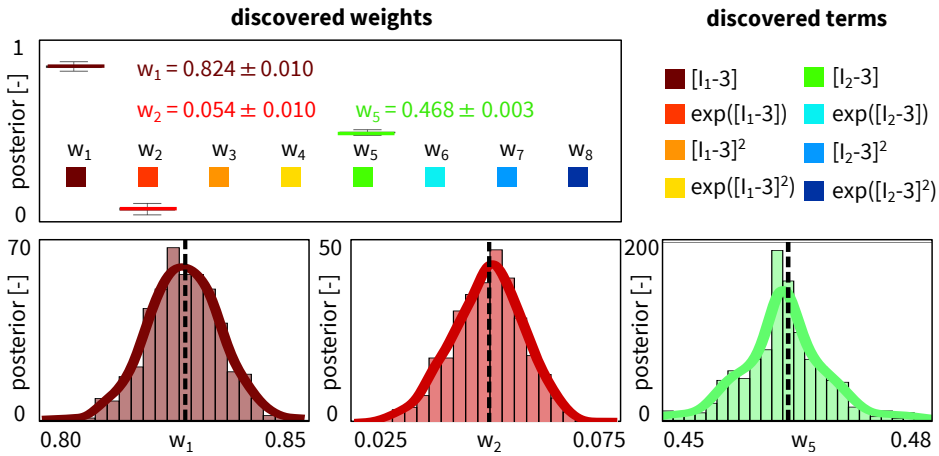

Figure 5: **Posterior distributions of discovered isotropic model parameters for training on incomplete data.** When using incomplete data for training, the isotropic, perfectly incompressible Bayesian constitutive neural network rediscovers the Gaussian noise perturbed Mooney Rivlin model with an additional term, $\psi = w_1\,[\,I_1 - 3\,] + w_2\,[\,\exp(w_2^*[I_2 - 3]) - 1\,], + w_5\,[\,I_2 - 3\,]$, with discovered network weights of $w_1 = 0.824 \pm 0.01$, $w_2 = 0.054 \pm 0.01$ and $w_5 = 0.468 \pm 0.00$, while all other network weights correctly train to zero.

in the context of interpretability, our specific network architecture, with standard external and internal network weights $\boldsymbol{w}$ and $\boldsymbol{w}^*$ and variational parameters $\boldsymbol{w}_\mu$ and $\boldsymbol{w}_\sigma$ only *after* the final hidden layer, turns out to be critical to selectively modulate sparsification. While we did experiment with alternative network architectures in which the variational parameters *replace* the external network weights $\boldsymbol{w}$, or all external and internal weights $\boldsymbol{w}$ and $\boldsymbol{w}^*$ [30], we believe that our current network architecture in Figure 2 provides the most robust parameter sparsification and the most interpretable model discovery. Taken together, our Bayesian constitutive neural network robustly and repeatably discovers predictive models, parameters, and uncertainty from synthetic data, even in the presence of noise and incomplete training data.

4. Transversely isotropic Bayesian constitutive neural networks

The *transversely isotropic* Bayesian constitutive neural network takes the deformation gradient $\boldsymbol{F}$ as input,

$$\boldsymbol{F} = \nabla_X \boldsymbol{\varphi} \,. \tag{33}$$

In addition, its kinematics are characterized through a pronounced direction $\boldsymbol{n}_0$ with unit length $\|\boldsymbol{n}_0\| = 1$ in the undeformed configuration, which map onto the pronounced direction $\boldsymbol{n} = \boldsymbol{F} \cdot \boldsymbol{n}_0$ with stretches $\|\boldsymbol{n}\| = \lambda_n$ in the deformed configuration. We characterize the deformation state through the two isotropic invariants I_1 and I_2, and two anisotropic invariants I_4 and I_5 [39],

$$\begin{aligned} I_1 &= \quad [\boldsymbol{F}^{\mathrm{t}} \cdot \boldsymbol{F}] : \boldsymbol{I} \quad \text{and} \quad I_2 = \tfrac{1}{2}\,[I_1^2 - [\boldsymbol{F}^{\mathrm{t}} \cdot \boldsymbol{F}] : [\boldsymbol{F}^{\mathrm{t}} \cdot \boldsymbol{F}]]\,, \\ I_4 &= \boldsymbol{n}_0 \cdot [\boldsymbol{F}^{\mathrm{t}} \cdot \boldsymbol{F}] \cdot \boldsymbol{n}_0 \quad \text{and} \quad I_5 = \quad \boldsymbol{n}_0 \cdot [\boldsymbol{F}^{\mathrm{t}} \cdot \boldsymbol{F}]^2 \cdot \boldsymbol{n}_0\,. \end{aligned} \tag{34}$$

A *perfectly incompressible* material has a constant Jacobian equal to one, $I_3 = J^2 = 1$. The network discovers *hyperelastic* material models that satisfy the second law of thermodynamics, and their Piola stress $\boldsymbol{P} = \partial\psi(\boldsymbol{F})/\partial\boldsymbol{F}$ is the derivative of the free energy $\psi(\boldsymbol{F})$ with respect to the deformation gradient $\boldsymbol{F}$ modified by a pressure term $-p\,\boldsymbol{F}^{-\mathrm{t}}$,

$$\boldsymbol{P} = \frac{\partial\psi}{\partial\boldsymbol{F}} - p\,\boldsymbol{F}^{-\mathrm{t}} = \frac{\partial\psi}{\partial I_1}\frac{\partial I_1}{\partial\boldsymbol{F}} + \frac{\partial\psi}{\partial I_2}\frac{\partial I_2}{\partial\boldsymbol{F}} + \frac{\partial\psi}{\partial I_4}\frac{\partial I_4}{\partial\boldsymbol{F}} + \frac{\partial\psi}{\partial I_5}\frac{\partial I_5}{\partial\boldsymbol{F}} - p\,\boldsymbol{F}^{-\mathrm{t}}\,, \tag{35}$$

where the hydrostatic pressure, $p = -\frac{1}{3}\,\boldsymbol{P} : \boldsymbol{F}$, acts as a Lagrange multiplier that we determine from the boundary conditions.

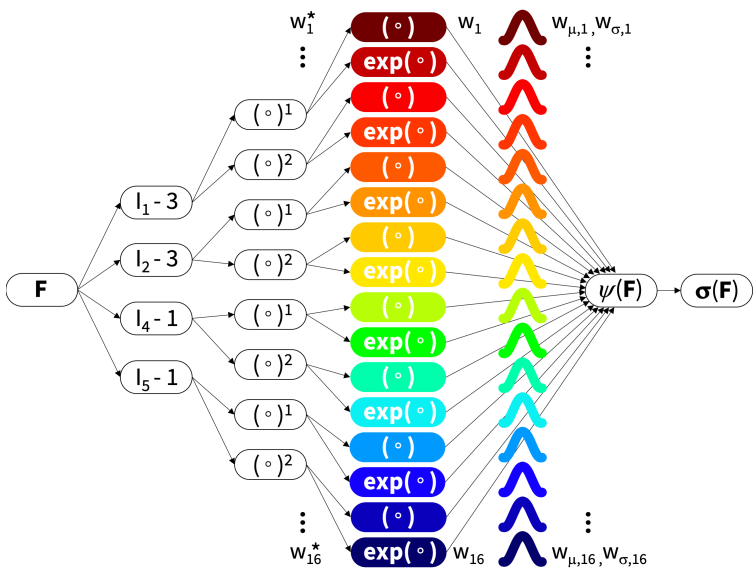

Figure 6: **Transversely isotropic Bayesian constitutive neural network.** The network has two hidden layers to discover the free-energy function $\psi(I_1, I_2, I_4, I_5)$ as a function of the invariants of the deformation gradient $\boldsymbol{F}$ using sixteen terms. The first layer generates powers $(\circ)$ and $(\circ)^2$ of the network input, the second layer applies the identity $(\circ)$ and exponential function $(\exp(\circ))$ to these powers, and the network output is the sum of these sixteen terms, weighted by their probability densities $\mathcal{N}(\boldsymbol{w}_\mu, \boldsymbol{w}_\sigma)$. During training, the network learns the network weights $\boldsymbol{w}$ and $\boldsymbol{w}^*$ and the variational parameters $\boldsymbol{w}_\mu$ and $\boldsymbol{w}_\sigma$.

We discover the free-energy function ψ using a Bayesian constitutive neural network that takes the deformation gradient $\boldsymbol{F}$ as input and approximates the free-energy function $\psi(\boldsymbol{F})$ as the sum of sixteen probability-weighted terms. Figure 6 illustrates our neural network with two hidden layers and sixteen nodes [23]. The first layer generates powers $(\circ)$ and $(\circ)^2$ of the network input, the two invariants I_1 and I_2. The second layer applies the identity, $(\circ)$ and the

exponential function ($\exp(\circ)$) to these powers. The network output is the sum of these sixteen terms, weighted by their probability densities $\mathcal{N}(w_\mu, w_\sigma)$. The free-energy function of this networks takes the following explicit form,

$$
\begin{aligned}
\psi = {} & \mathcal{N}(w_{\mu,1}\ , w_{\sigma,1}\ )\ w_1\ \ w_1^*\ \ [I_1 - 3]\ \ + \mathcal{N}(w_{\mu,2}\ , w_{\sigma,2}\ )\ w_2\ \ [\exp(w_2^*\ \ [I_1 - 3]\ \ ) - 1] \\
& + \mathcal{N}(w_{\mu,3}\ , w_{\sigma,3}\ )\ w_3\ \ w_3^*\ \ [I_1 - 3]^2 + \mathcal{N}(w_{\mu,4}\ , w_{\sigma,4}\ )\ w_4\ \ [\exp(w_4^*\ \ [I_1 - 3]^2) - 1] \\
& + \mathcal{N}(w_{\mu,5}\ , w_{\sigma,5}\ )\ w_5\ \ w_5^*\ \ [I_2 - 3]\ \ + \mathcal{N}(w_{\mu,6}\ , w_{\sigma,6}\ )\ w_6\ \ [\exp(w_6^*\ \ [I_2 - 3]\ \ ) - 1] \\
& + \mathcal{N}(w_{\mu,7}\ , w_{\sigma,7}\ )\ w_7\ \ w_7^*\ \ [I_2 - 3]^2 + \mathcal{N}(w_{\mu,8}\ , w_{\sigma,8}\ )\ w_8\ \ [\exp(w_8^*\ \ [I_2 - 3]^2) - 1] \\
& + \mathcal{N}(w_{\mu,9}\ , w_{\sigma,9}\ )\ w_9\ \ w_9^*\ \ [I_4 - 1]\ \ + \mathcal{N}(w_{\mu,10}, w_{\sigma,10})\ w_{10}\,[\exp(w_{10}^*\,[I_4 - 1]\ \ ) - 1] \\
& + \mathcal{N}(w_{\mu,11}, w_{\sigma,11})\ w_{11}\,w_{11}^*\,[I_4 - 1]^2 + \mathcal{N}(w_{\mu,12}, w_{\sigma,12})\ w_{12}\,[\exp(w_{12}^*\,[I_4 - 1]^2) - 1] \\
& + \mathcal{N}(w_{\mu,13}, w_{\sigma,13})\ w_{13}\,w_{13}^*\,[I_5 - 1]\ \ + \mathcal{N}(w_{\mu,14}, w_{\sigma,14})\ w_{14}\,[\exp(w_{14}^*\,[I_5 - 1]\ \ ) - 1] \\
& + \mathcal{N}(w_{\mu,15}, w_{\sigma,15})\ w_{15}\,w_{15}^*\,[I_5 - 1]^2 + \mathcal{N}(w_{\mu,16}, w_{\sigma,16})\ w_{16}\,[\exp(w_{16}^*\,[I_5 - 1]^2) - 1]\,,
\end{aligned}
\tag{36}
$$

corrected by the pressure term, $\psi = \psi - p\,[J - 1]$. To complete the definition of the Piola stress in equation (35), we take its derivatives with respect to the two invariants,

$$
\begin{aligned}
\frac{\partial\psi}{\partial I_1} = {} & \mathcal{N}(w_{\mu,1}\ , w_{\sigma,1})\ \ w_1\ \ w_1^*\ \ + \mathcal{N}(w_{\mu,2}\ , w_{\sigma,2})\ \ w_2\ \ w_2^*\ \ \exp(\,w_2^*\ \ [\,I_1 - 3\,]\ \ ) \\
& + 2\,[\,I_1 - 3\,][\ \mathcal{N}(w_{\mu,3}\ , w_{\sigma,3})\ \ w_3\ \ w_3^*\ \ + \mathcal{N}(w_{\mu,4}\ , w_{\sigma,4})\ \ w_4\ \ w_4^*\ \ \exp(\,w_4^*\ \ [\,I_1 - 3\,]^2)\,, \\
\frac{\partial\psi}{\partial I_2} = {} & \mathcal{N}(w_{\mu,5}\ , w_{\sigma,5})\ \ w_5\ \ w_5^*\ \ + \mathcal{N}(w_{\mu,6}\ , w_{\sigma,6})\ \ w_6\ \ w_6^*\ \ \exp(\,w_6^*\ \ [\,I_2 - 3\,]\ \ ) \\
& + 2\,[\,I_2 - 3\,][\ \mathcal{N}(w_{\mu,7}\ , w_{\sigma,7})\ \ w_7\ \ w_7^*\ \ + \mathcal{N}(w_{\mu,8}\ , w_{\sigma,8})\ \ w_8\ \ w_8^*\ \ \exp(\,w_8^*\ \ [\,I_2 - 3\,]^2)\,, \\
\frac{\partial\psi}{\partial I_4} = {} & \mathcal{N}(w_{\mu,9}\ , w_{\sigma,9})\ \ w_9\ \ w_9^*\ \ + \mathcal{N}(w_{\mu,10}, w_{\sigma,10})\ w_{10}\,w_{10}^*\ \exp(\,w_{10}^*\ [\,I_4 - 1\,]\ \ ) \\
& + 2\,[\,I_4 - 1\,][\ \mathcal{N}(w_{\mu,11}, w_{\sigma,11})\ w_{11}\,w_{11}^*\ + \mathcal{N}(w_{\mu,12}, w_{\sigma,12})\ w_{12}\,w_{12}^*\ \exp(\,w_{12}^*\ [\,I_4 - 1\,]^2)\,, \\
\frac{\partial\psi}{\partial I_5} = {} & \mathcal{N}(w_{\mu,13}, w_{\sigma,13})\ w_{13}\,w_{13}^*\ + \mathcal{N}(w_{\mu,14}, w_{\sigma,14})\ w_{14}\,w_{14}^*\ \exp(\,w_{14}^*\ [\,I_5 - 1\,]\ \ ) \\
& + 2\,[\,I_5 - 1\,][\ \mathcal{N}(w_{\mu,15}, w_{\sigma,15})\ w_{15}\,w_{15}^*\ + \mathcal{N}(w_{\mu,16}, w_{\sigma,16})\ w_{16}\,w_{16}^*\ \exp(\,w_{16}^*\ [\,I_5 - 1\,]^2)\,.
\end{aligned}
\tag{37}
$$

The network has two sets of network weights, $\boldsymbol{w}$ and $\boldsymbol{w}^*$, associated the sixteen terms of the free-energy function, and two sets of variational parameters, $\boldsymbol{w}_\mu$ and $\boldsymbol{w}_\sigma$, associated with the means and standard deviations of these sixteen terms. We learn these 32 weights by minimizing the three-term loss function $L = L_{\mathsf{NLL}} + L_{\mathsf{KL}} + L_1$ in equation (21).

We consider data from *biaxial extension* tests with stretches λ_1 and λ_2 in the 1- and 2-directions, such that $I_1 = \lambda_1^2 + \lambda_2^2 + (\lambda_1\lambda_2)^{-2}$ and $I_2 = \lambda_1^{-2} + \lambda_2^{-2} + (\lambda_1\lambda_2)^2$ and $I_4 = \lambda_1^2\cos^2\alpha + \lambda_2^2\sin^2\alpha$ and $I_5 = \lambda_1^4\cos^2\alpha + \lambda_2^4\sin^2\alpha$. The deformation gradient is $\mathbf{F} = \mathrm{diag}\,\{\,\lambda_1, \lambda_2, (\lambda_1\lambda_2)^{-1}\,\}$ and the Piola stress is $\mathbf{P} = \mathrm{diag}\,\{\,P_{11}, P_{22}, 0\,\}$. We use the zero-normal-stress condition, $P_{33} = 0$, to determine the pressure, $p = 2(\lambda_1\lambda_2)^{-2}\ \partial\psi/\partial I_1 + 2(\lambda_1^{-2} + \lambda_2^{-2})\ \partial\psi/\partial I_2$ [6]. In what follows, we use data from biaxial extension tests on samples with *two fiber families* with *identical properties* that are mounted symmetrically with respect to the 1- and 2-directions. This implies that we can combine the effects of both fiber families in the fourth and fifth invariants I_4 and I_5 by multiplying the anisotropic stress terms by a factor $n_{\mathrm{f}} = 2$. In addition, we assume that these two fiber families do not interact such that the eighth invariant vanishes identically, $I_8 = 0$ [28]. Equation (35) then provides explicit analytical expressions for the Piola stresses P_1 and P_2 in terms of the stretches λ_1 and λ_2 [27],

$$
\begin{aligned}
P_1 &= 2\left[\lambda_1 - \frac{1}{\lambda_1^3\lambda_2^2}\right]\frac{\partial\psi}{\partial I_1} + 2\left[\lambda_1\lambda_2^2 - \frac{1}{\lambda_1^3}\right]\frac{\partial\psi}{\partial I_2} + 2\,n_{\mathrm{f}}\,\lambda_1\cos^2\alpha\,\frac{\partial\psi}{\partial I_4} + 4\,n_{\mathrm{f}}\,\lambda_1^3\cos^2\alpha\,\frac{\partial\psi}{\partial I_5}\,, \\
P_2 &= 2\left[\lambda_2 - \frac{1}{\lambda_1^2\lambda_2^3}\right]\frac{\partial\psi}{\partial I_1} + 2\left[\lambda_1^2\lambda_2 - \frac{1}{\lambda_2^3}\right]\frac{\partial\psi}{\partial I_2} + 2\,n_{\mathrm{f}}\,\lambda_2\sin^2\alpha\,\frac{\partial\psi}{\partial I_4} + 4\,n_{\mathrm{f}}\,\lambda_2^3\sin^2\alpha\,\frac{\partial\psi}{\partial I_5}\,,
\end{aligned}
\tag{38}
$$

which we translate into the Cauchy stresses σ_1 and σ_2 that are reported in the experiment [31] using $J\,\boldsymbol{\sigma} = \boldsymbol{P}\cdot\boldsymbol{F}^{\mathrm{t}} = \partial\psi/\partial\boldsymbol{F}\cdot\boldsymbol{F}^{\mathrm{t}} - p\,\boldsymbol{I}$,

$$
\begin{aligned}
\sigma_1 &= 2\left[\lambda_1^2 - \frac{1}{\lambda_1^2\lambda_2^2}\right]\frac{\partial\psi}{\partial I_1} + 2\left[\lambda_1^2\lambda_2^2 + \frac{1}{\lambda_1^2}\right]\frac{\partial\psi}{\partial I_2} + 2\,n_{\mathrm{f}}\,\lambda_1^2\cos^2\alpha\,\frac{\partial\psi}{\partial I_4} + 4\,n_{\mathrm{f}}\,\lambda_1^4\cos^2\alpha\,\frac{\partial\psi}{\partial I_5}\,, \\
\sigma_2 &= 2\left[\lambda_2^2 - \frac{1}{\lambda_1^2\lambda_2^2}\right]\frac{\partial\psi}{\partial I_1} + 2\left[\lambda_1^2\lambda_2^2 + \frac{1}{\lambda_2^2}\right]\frac{\partial\psi}{\partial I_2} + 2\,n_{\mathrm{f}}\,\lambda_2^2\sin^2\alpha\,\frac{\partial\psi}{\partial I_4} + 4\,n_{\mathrm{f}}\,\lambda_2^4\sin^2\alpha\,\frac{\partial\psi}{\partial I_5}\,.
\end{aligned}
\tag{39}
$$

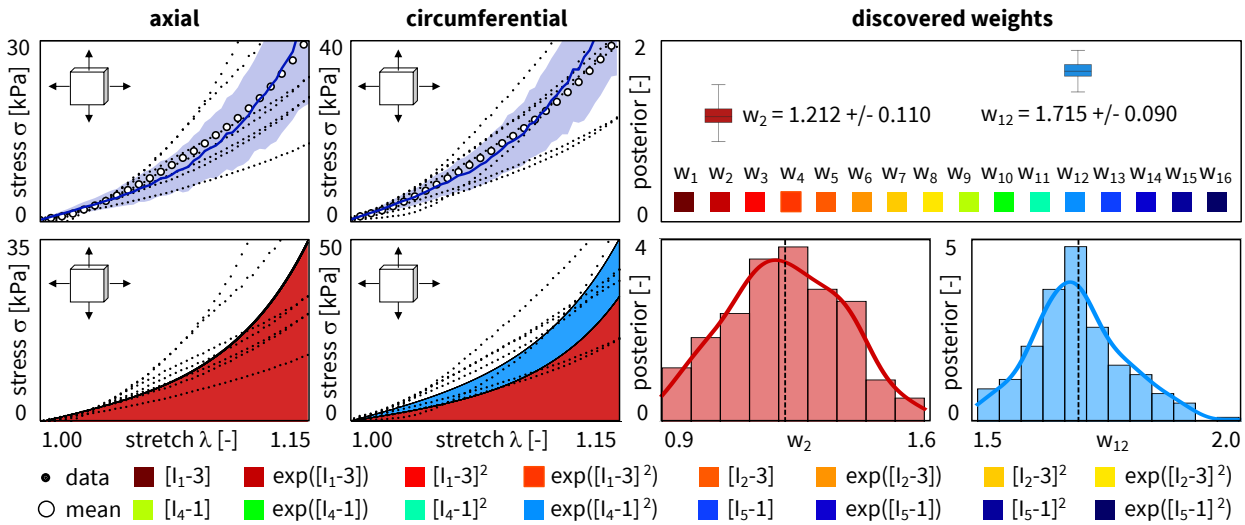

Figure 7: **Biaxial extension data and discovered model, parameters, and uncertainties for healthy medial layer.** True axial and circumferential stresses σ_{axl} and σ_{cir} as functions of the biaxial stretches λ_{axl} and λ_{cir} for the transversely isotropic Bayesian constitutive neural network with two hidden layers, sixteen nodes, and sixteen prior distributions from Figure 6. Dots illustrate the axial and circumferential biaxial extension data of $n = 7$ healthy medial aortic samples; solid blue curves and blue-shaded areas indicate the mean predicted stresses ± standard deviations; color-coded areas highlight the sixteen possible contributions to the discovered stress function. The Bayesian network discovers a two-term model, $\psi = w_2\,[\exp(w_2^*[I_1 - 3]) - 1] + w_{12}\,[\exp(w_{12}^*[I_4 - 1]^2) - 1]$, with network weights of $w_2 = 1.212 \pm 0.11$ and $w_{12} = 1.715 \pm 0.09$, while all other network weights train to zero.

We explore the performance of the transversely isotropic Bayesian neural network from Figure 6 on the basis of real stress-stretch data from human abdominal aortic tissue samples [31, 32]. We consider three data sets of axial and circular stretch stress pairs from the healthy medial layer, the healthy composite aorta, and the aneurysmal composite aorta, in the stretch ranges of $1.00 \leq \lambda_{\text{axl}}, \lambda_{\text{cir}} \leq 1.15$, and compare the discovered models, parameters, and uncertainties.

Bayesian networks robustly discover model, parameters, and uncertainties for the healthy medial layer. Figure 7, right, shows the equibiaxial extension data and the discovered model for the healthy human medial aorta. The training data consist of axial and circumferential stretch stress pairs from $n = 7$ samples of the medial layer with collagen fiber orientations $\boldsymbol{n}_0 = [\cos(\alpha), \sin(\alpha), 0]^{\text{t}}$ for $\alpha = 9.85°$. Notably, the data points of the healthy medial aorta in Figure 7 display a moderate *inter-sample variation*, a pronounced *stretch stiffening*, and a notable *anisotropy*. Interestingly, out of $2^{16} = 65,536$ possible models, the Bayesian network discovers a familiar model, $\psi = \frac{1}{2}\,a_1\,[\exp(b_1[I_1 - 3]) - 1] + \frac{1}{2}\,a_4\,[\exp(b_4[I_4 - 1]^2) - 1]$, that is dominated by the red exponential linear first invariant Demiray term [5], $\frac{1}{2}\,a_1\,[\exp(b_1[I_1 - 3]) - 1]$, and by the light blue exponential quadratic fourth invariant Holzapfel term [13], $\frac{1}{2}\,a_4\,[\exp(b_4[I_4 - 1]^2) - 1]$. Importantly, the network robustly discovers a *sparse* and *interpretable* model [26], with only two terms–an isotropic term associated with the extracellular matrix and an anisotropic term associated with the collagen fibers–while all other network weights autonomously train to zero. Notably, the discovered model agrees well with a previous analysis of human tissue samples from the healthy medial aorta, which ranked this model within the best-in-class two-term models for the healthy media [24], not only for data from equibiaxial testing, but for data from five different biaxial tests combined [34]. The non-zero weights of the model translate into stiffnesses of $a_1 = 1.12$ kPa and $a_4 = 1.72$ kPa and nonlinearity parameters of $b_1 = 5.87$ and $b_4 = 3.96$, four meaningful parameters with a clear physical interpretation. Figure 7, left, highlights the discovered model parameters associated with these two terms, $w_2 = 1.212 \pm 0.11$ kPa and $w_{12} = 1.715 \pm 0.09$ kPa, along with their posterior distributions and standard deviations. Compared to the previous examples based on synthetic data perturbed by controlled Gaussian noise in Figures 3 and 5, the posterior distributions of the weights in Figure 7, right, display significantly larger standard deviations. This agrees with our expectation of large parameter variations across real biomedical data [31], and with the wide inter-sample spread in the raw data in Figure 7, left. Taken together, our Bayesian constitutive neural network

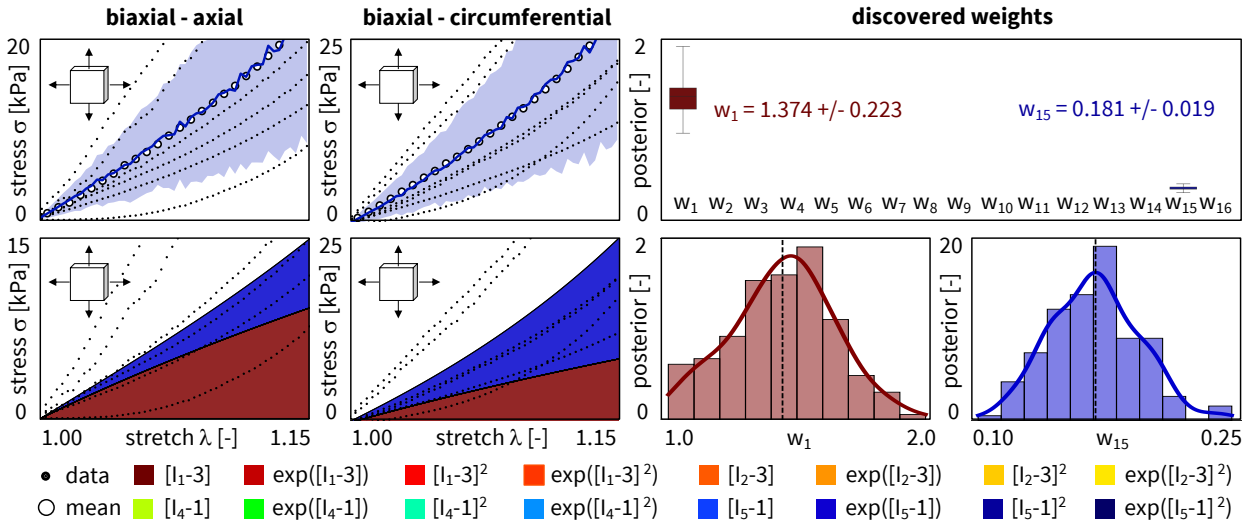

Figure 8: **Biaxial extension data and discovered model, parameters, and uncertainties for healthy composite aorta.** True axial and circumferential stresses σ_{axl} and σ_{cir} as functions of the biaxial stretches λ_{axl} and λ_{cir} for the transversely isotropic Bayesian constitutive neural network with two hidden layers, sixteen nodes, and sixteen prior distributions from Figure 6. Dots illustrate the axial and circumferential biaxial extension data of $n = 6$ healthy composite aortic samples; solid blue curves and blue-shaded areas indicate the mean predicted stresses ± standard deviations; color-coded areas highlight the sixteen possible contributions to the discovered stress function. The Bayesian network discovers a two-term model, $\psi = w_1\,[\,I_1 - 3\,] + w_{15}\,[\,I_5 - 1\,]^2$, with network weights of $w_1 = 1.374 \pm 0.22$ and $w_{15} = 0.181 \pm 0.02$, while all other network weights train to zero.

robustly discovers interpretable transversely isotropic models, parameters, and uncertainties from real world data; it not only learns point values, but distributions of parameters with credible intervals, means, and standard deviations that provide valuable information to communicate our confidence in the model, and improve the model if needed.

Bayesian networks discover a more linear model with more uncertainty for the composite aorta than for the medial layer. Figure 8, left, shows the biaxial extension data and the discovered model for the healthy human composite aorta. The training data consist of axial and circumferential stretch stress pairs from $n = 6$ samples of the composite tissue with collagen fiber orientations $\boldsymbol{n}_0 = [\cos(\alpha), \sin(\alpha), 0\,]^{\text{t}}$ for $\alpha = 29.8°$. Notably, within the tested stretch regime of $1.00 \leq \lambda_{\text{axl}}, \lambda_{\text{cir}} \leq 1.15$, and for the tested equibiaxial state, the data points of the healthy composite aorta in Figure 8 display a significant *inter-sample variation*, but *no stretch stiffening*, and only *moderate anisotropy*. As a result, the discovered model, $\psi = \frac{1}{2}\mu_1\,[\,I_1 - 3\,] + \frac{1}{2}\mu_5\,[\,I_5 - 1\,]^2$, is dominated by the dark red linear first invariant neo Hooke term [42], $\frac{1}{2}\mu_1\,[\,I_1 - 3\,]$, supplemented by the dark blue quadratic fifth invariant term, $\frac{1}{2}\mu_5\,[\,I_5 - 1\,]^2$. Similar to the previous example, the L_1 regularization promotes a *sparse* and *interpretable* model [26], with only two terms, one isotropic and one anisotropic, while all other network weights autonomously train to zero. This includes the weights of the exponential terms that were prominently featured by the medial layer model [34] to account for the nonlinear stretch stiffening in Figure 7, that is not present in the composite tissue samples in Figure 8. The non-zero weights of the discovered composite model translate into the stiffnesses of $\mu_1 = 2.748$ kPa and $\mu_{15} = 0.362$ kPa, two meaningful parameters with a clear physical interpretation. Figure 8, right, highlights the discovered model parameters associated with these two terms, along with their posterior distributions and standard deviations, $w_1 = 1.374 \pm 0.22$ kPa and $w_{15} = 0.181 \pm 0.02$ kPa. Again, we observe larger standard deviations than for the synthetic data in Figures 3 and 5, but also larger standard deviations than for the medial layer in Figure 7, right. Confirming this observation, the model uncertainty for the composite aorta associated with the blue-shaded areas in Figure 8, left, is significantly larger than the model uncertainty for the medial layer associated with the blue-shaded areas in Figure 7, left. Since the composite aorta is made up of two mechanically relevant layers [13], the media and the adventitia, both with distinct collagen fiber orientations [31], a mechanistic microstructural model for the composite aorta would need to include at least two anisotropic terms, one for each layer [35]. Instead, we discover a macroscopic phenomenological model that averages the collagen fiber stiffening and anisotropy into a more linear

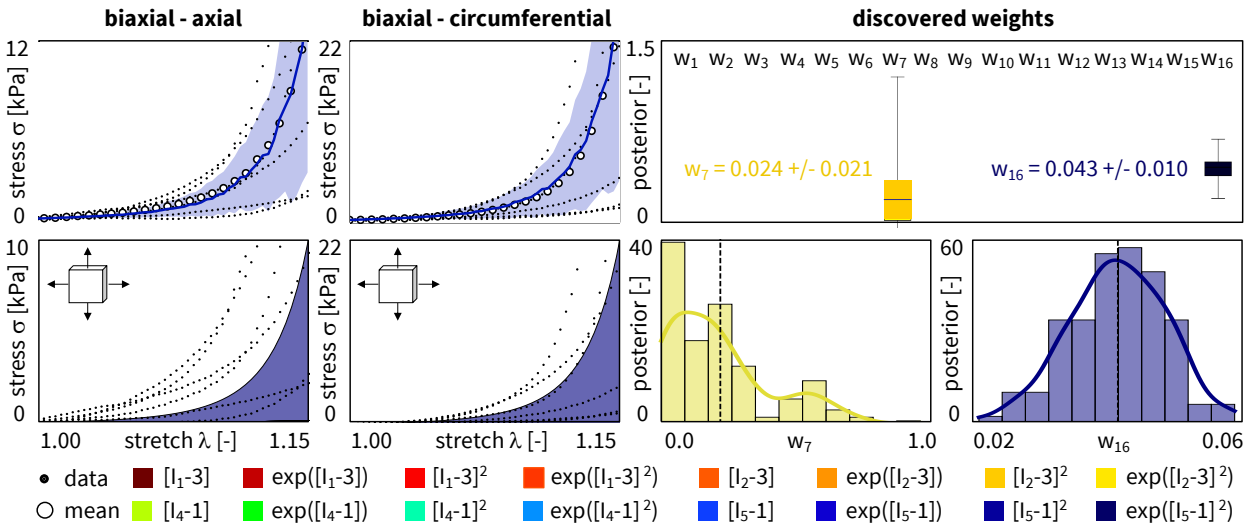

Figure 9: **Biaxial extension data and discovered model, parameters, and uncertainties for aneurysmal composite aorta.** True axial and circumferential stresses σ_{axl} and σ_{cir} as functions of the biaxial stretches λ_{axl} and λ_{cir} for the transversely isotropic, perfectly incompressible Bayesian constitutive neural network with two hidden layers, sixteen nodes, and sixteen prior distributions from Figure 6. Dots illustrate the axial and circumferential biaxial extension data of $n = 6$ aneurysmal aortic samples; solid blue curves and blue-shaded areas indicate the mean predicted stresses ± standard deviations; color-coded areas highlight the sixteen possible contributions to the discovered stress function. The Bayesian network discovers a two-term model, $\psi = w_7\,[\,I_2 - 3\,]^2 + w_{16}\,[\exp(w^*_{16}[\,I_5 - 1\,]^2) - 1]$, with network weights of $w_7 = 0.024 \pm 0.02$ kPa and $w_{16} = 0.042 \pm 0.01$ kPa, while all other network weights train to zero.

and more isotropic model. Notably, our Bayesian network discovers large aleatoric and epistemic uncertainties that alert us of these unreliable predictions and call for model improvement [33]. Taken together, our Bayesian network autonomously discovers distinct characteristic models for different tissue types: a mechanistic microstructural model with exponential stiffening and notable anisotropy for the medial layer and a phenomenological macrostructural model with a linear response and more isotropy for the composite aorta. Naturally, this composite model contains less microstructural information and generates larger uncertainties.

Bayesian networks discover a more exponential model with larger uncertainties for the diseased than for the healthy aorta. Figure 9, left, shows the biaxial extension data and the discovered model for the aneurysmal human composite aorta. The training data consist of axial and circumferential stretch stress pairs from $n = 6$ samples with collagen fiber orientations $\boldsymbol{n}_0 = [\cos(\alpha), \sin(\alpha), 0\,]^{\text{t}}$ for $\alpha = 31.5°$. Similar to the healthy tissue samples in Figure 8, the data points of the diseased tissue samples in Figure 9 display a significant *inter-sample variation*; yet, in contrast to the healthy samples, they also display a pronounced *stretch stiffening*, a notable *anisotropy*. Accordingly, the discovered model, $\psi = \frac{1}{2}\mu\,[\,I_2 - 3\,]^2 + \frac{1}{2}\,a\,\exp(b[\,I_5 - 1\,]^2) - 1)/b$, is dominated by the dark blue exponential quadratic fifth invariant Holzapfel-type term [13], $\frac{1}{2}\,a\,\exp(b[\,I_5 - 1\,]^2) - 1)/b$, supplemented by a small yellow quadratic second invariant term [18], $\frac{1}{2}\mu\,[\,I_2 - 3\,]^2$. Similar to all previous examples, the L_1 regularized network robustly discovers a *sparse* and *interpretable* model with only two terms, one isotropic and one anisotropic, while all other network weights autonomously train to zero [26]. The non-zero weights translate into the model parameters of $\mu = 0.048$ kPa, $a = 2\,w_{16}w^*_{16} = 0.05$ kPa, and $b = w^*_{16} = 5.0$. Figure 9, right, highlights the two discovered parameters, their posterior distributions, and their standard deviations, $w_7 = 0.024 \pm 0.02$ kPa and $w_{16} = 0.042 \pm 0.01$ kPa. In agreement with our intuition, of all three discovered models, for the healthy medial layer, the healthy composite aorta, and the aneurysmal composite aorta, the model for the aneurysmal composite aorta displays the largest degree of *uncertainty*, as we conclude from the large blue-shaded areas in Figure 9, left. This uncertainty is a natural result of the complex but diffuse microstructure of aneurysmal tissue, dominated by straight and thick struts of collagen [31]. Taken together, our Bayesian constitutive neural networks successfully delineate between healthy and diseased tissues and discover different models–linear for the composite healthy and exponentially stiffening for the composite diseased tissue–both with interpretable parameters, credible intervals, means, and standard deviations.

5. Conclusion

The inability to communicate uncertainty and the risk to produce unreliable predictions are serious deficiencies of classical neural networks. This makes them unsuitable for biomedical applications, where data are sparse and vary significantly from one patient to another. To supplement medical decision making by neural network modeling–especially with a view towards human health–it is absolutely critical to know and understand the uncertainties associated with our model predictions. Here we propose an efficient and robust method, regularized variational Bayesian inference, to train constitutive neural networks and discover the model, parameters, and uncertainties that best explain and predict the unique characteristics of biomedical systems. To demonstrate the potential of this approach, we prototype solutions on synthetic data perturbed by aleatoric noise and on real world data from healthy and diseased human arteries. Our results on synthetic data demonstrate that Bayesian constitutive neural networks can successfully rediscover the initial model, even in the presence of noise, and robustly discover uncertainties, even from incomplete training data. Interestingly, we observe larger uncertainties for equibiaxial tests than for tension, compression, and shear tests. These uncertainties provide valuable guidance for model improvement: If we decided to collect more data, additional equibiaxial tests would most efficiently reduce epistemic uncertainties and improve model robustness. Our results on healthy and diseased human arteries demonstrate that Bayesian constitutive neural networks can successfully discriminate between healthy and diseased tissues, robustly discover interpretable models for both, and efficiently quantify uncertainties in model discovery. Notably, model uncertainty is smallest for the healthy medial layer, moderate for the healthy composite aorta, and largest for the diseased aorta. This observation is in line with an increasing microstructural complexity, from the well-organized healthy medial architecture with pronounced collagen fiber orientations to the distorted aneurysmal tissue with dispersed collagen fragments. Importantly, the failure of the Bayesian prediction presents an opportunity to learn: We could expand our current model library and reanalyze the same data with *new models*, or even collect *new data* and improve model performance in regions with high uncertainties. Especially in risk-sensitive areas like medical diagnosis, it is paramount that we can precisely communicate our confidence in our model predictions. Here we have prototyped this approach for healthy and diseased human arteries. We envision that Bayesian model discovery will generalize naturally to other biomedical systems for which real-world data are rare and inter-personal variations are large. Knowing, understanding, and communicating the uncertainties in automated model discovery is a vital step to improve model prediction, enable personalized simulation, and support informed decision making.

Data availability

Our source code, data, and examples will be available at https://github.com/LivingMatterLab/CANNs.

Acknowledgments

This work was supported by the Emmy Noether Grant 533187597 *Computational Soft Material Mechanics Intelligence* to Kevin Linka and by the NSF CMMI Award 2320933 *Automated Model Discovery for Soft Matter* and the ERC Advanced Grant 101141626 *DISCOVER* to Ellen Kuhl.

References

[1] Alber M, Buganza Tepole A, Cannon W, De S, Dura-Bernal S, Garikipati K, Karniadakis GE, Lytton WW, Perdikaris P, Petzold L, Kuhl E (2019) Integrating machine learning and multiscale modeling–perspectives, challenges, and opportunities in the biological, biomedical, and behavioral sciences. NPJ Digital Medicine 2:115.

[2] Bayes T, Price R (1763). An Essay towards Solving a Problem in the Doctrine of Chances. By the Late Rev. Mr. Bayes, F.R.S. communicated by Mr. Price, in a Letter to John Canton, A.M.F.R.S. Philosophical Transactions of the Royal Society of London. 53:370–418.

[3] Blundell C, Cornebise J, Kavukcuoglu K, Wierstra D (2015) Weight uncertainty in neural networks. Proceedings of the 32nd International Conference on Machine Learning, Lille, France, PMLR 37: 1613-1622.

[4] Clevert DA, Unterthiner T, Hochreiter, S (2016). Fast and accurate deep network learning by exponential linear units (elus). International Conference on Learning Representations. 1613-1622.

[5] Demiray H (1972) A note on the elasticity of soft biological tissues. Journal of Biomechanics 5: 309-311.

[6] H. Fehervary, Vander Sloten, N. Famaey (2019) Development of an improved parameter fitting method for planar biaxial testing using rakes. International Journal for Numerical Methods in Biomedical Engineering 35: e3174.

[7] Flaschel M, Kumar S, De Lorenzis L (2023) Automated discovery of generalized standard material models with EUCLID. Computer Methods in Applied Mechanics and Engineering 405: 115867.
[8] Gal Y, Ghahramani Z (2016) Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. Proceedings of the 33rd International Conference on Machine Learning (ICML), 1050-1059.
[9] Goodfellow I, Bengio Y, Courville A (2016) Deep Learning. MIT Press.
[10] Graves A (2011) Practical variational inference for neural networks. Advances in Neural Information Processing Systems, 545-552.
[11] Hastie T, Tibshirani R, Friedman J (2009) The Elements of Statistical Learning. Second Edition; Springer, New York.
[12] Hinton GE, van Camp D (1993) Keeping the neural networks simple by minimizing the description length of the weights. Proceedings of the 6th Annual ACM Conference on Computational Learning Theory, 5-13.
[13] Holzapfel GA, Gasser TC, Ogden RW (2000) A new constitutive framework for arterial wall mechanics and comparative study of material models. Journal of Elasticity 61:1-48.
[14] Holzapfel G (2000) Nonlinear Solid Mechanics: A Continuum Approach to Engineering. John Wiley & Sons, Chichester.
[15] Joshi A, Thakolkaran P, Zheng Y, Escande M, Flaschel M, De Lorenzis L, Kumar S (2022) Bayesian-EUCLID: Discovering hyperelastic material laws with uncertainties. Computer Methods in Applied Mechanics and Engineering 398: 115225.
[16] Kendall A, Gal Y (2017) What uncertainties do we need in Bayesian deep learning for computer vision? Advances in Neural Information Processing Systems, 5574-5584.
[17] Klingma DP, Welling M (2013) Auto-encoding variational Bayes. arXiv preprint arXiv:1312.6114.
[18] Kuhl E, Goriely A (2024) I too love I2: A new class of hyperelastic isotropic incompressible models based solely on the second invariant. Journal of the Mechanics and Physics of Solids 188: 105670.
[19] Linka K, Hillgartner M, Abdolazizi KP, Aydin RC, Itskov M, Cyron CJ (2021) Constitutive artificial neural networks: A fast and general approach to predictive data-driven constitutive modeling by deep learning. Journal of Computational Physics 429: 110010.
[20] Linka K, Schafer A, Meng X, Zou Z, Karniadakis GE, Kuhl E (2022) Bayesian Physics-Informed Neural Networks for real-world nonlinear dynamical systems. Computer Methods in Applied Mechanics and Engineering 402: 115346.
[21] Linka K, Kuhl E (2023) A new family of Constitutive Artificial Neural Networks towards automated model discovery. Computer Methods in Applied Mechanics and Engineering 403: 115731.
[22] Linka K, St Pierre S, Kuhl E (2023) Automated model discovery for human brain using Constitutive Artificial Neural Networks. Acta Biomaterialia 160: 134-151.
[23] Linka K, Buganza Tepole A, Holzapfel GA, Kuhl E (2023) Automated model discovery for skin: Discovering the best model, data, and experiment. Computer Methods in Applied Mechanics and Engineering 410: 116007.
[24] Linka K, Kuhl E (2024) Best-in-class modeling: A novel strategy to discover constitutive models for soft matter systems. Extreme Mechanics Letters 70: 102181.
[25] MacKay DJC (1992) A practical Bayesian framework for backpropagation networks. Neural Computation 4: 448-472.
[26] McCulloch JA, St.Pierre SR, Linka K, Kuhl E (2024) On sparse regression, Lp-regularization, and automated model discovery. International Journal for Numerical Methods in Engineering, 125: e7481.
[27] McCulloch JA, Kuhl E (2024) Automated model discovery for textile structures: The unique mechanical signature of warp knitted fabrics. bioRxiv preprint doi:10.1101/2024.07.26.605392.
[28] Merodio J, Ogden RW (2006) The influence of the invariant I8 on the stress–deformation and ellipticity characteristics of doubly fiber-reinforced non-linearly elastic solids. International Journal of Non-Linear Mechanics 41: 556–563.
[29] Mooney M (1940) A theory of large elastic deformations. Journal of Applied Physics 11: 582-590.
[30] Neal RM (1996) Bayesian Learning for Neural Networks. Springer Science + Business Media, New York.
[31] Niestrawska JA, Viertler C, Regitnig P, Cohnert TU, Sommer G, Holzapfel GA (2015) Microstructure and mechanics of healthy and aneurysmatic abdominal aortas: experimental analysis and modelling. Journal of the Royal Society Interface 13: 20160620.
[32] Niestrawska JA, Haspinger DC, Holzapfel GA (2018) The influence of fiber dispersion on the mechanical response of aortic tissues in health and disease: a computational study. Computer Methods in Biomechanics and Biomedical Engineering 21:99-112.
[33] Oliver A, Shields MD, Graham-Brady L (2021) Bayesian neural networks for uncertainty quantification in data-driven materials modeling. Computer Methods in Applied Mechanics and Engineering 386: 114079.
[34] Peirlinck M, Linka K, Hurtado JA, Holzapfel GA, Kuhl E (2024) Democratizing biomedical simulation through automated model discovery and a universal material subroutine. Computational Mechanics, in press, doi:10.1101/2023.12.06.570487.
[35] Peirlinck M, Hurtado JA, Rausch MK, Buganza Tepole A, Kuhl E (2024) A universal material model subroutine for soft matter systems. Engineering with Compupters 2024; in press, doi:10.48550/arXiv.2404.13144.
[36] Peng GCY, Alber M, Buganza Tepole A, Cannon W, De S, Dura-Bernal S, Garikipati K, Karniadakis GE, Lytton WW, Perdikaris P, Petzold L, Kuhl E (2021) Multiscale modeling meets machine learning: What can we learn? Archive of Computational Methods in Engineering 28:1017-1037.
[37] Raissi M, Perdikaris P, Karniadakis GE (2019) Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational Physics 378:686–707.
[38] Rivlin RS (1948) Large elastic deformations of isotropic materials. IV. Further developments of the general theory. Philosophical Transactions of the Royal Society of London Series A 241: 379–397.
[39] Spencer AJM (1971) Theory of Invariants. In: Eringen AC, Ed., Continuum Physics Vol. 1: 239-353, Academic Press, New York.
[40] Szegedy C, Zaremba W, Sutskever I, Bruna J, Erhan D, Goodfellow I, Fergus R (2014) Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199.
[41] Tibshirani R (1996) Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society B 58: 267-288.
[42] Treloar LRG (1948) Stresses and birefringence in rubber subjected to general homogeneous strain. Proceedings of the Physical Society 60: 135-144.