

RESEARCH ARTICLE

WILEY

Physics-constrained symbolic model discovery for polyconvex incompressible hyperelastic materials

Bahador Bahmani | WaiChing Sun[✉]

Department of Civil Engineering and Engineering Mechanics, Columbia University, New York, New York, USA

Correspondence

WaiChing Sun, Department of Civil Engineering and Engineering Mechanics, Columbia University, 614 SW Mudd, Mail Code: 4709, New York, NY, 10027, USA.
Email: wsun@columbia.edu

Funding information

Air Force Office of Scientific Research, Grant/Award Numbers: FA9550-21-1-0391, FA9550-21-1-0027; National Science Foundation, Grant/Award Number: CMMI-1846875; MURI, Grant/Award Number: FA9550-19-1-0318

Abstract

We present a machine learning framework capable of consistently inferring mathematical expressions of hyperelastic energy functionals for incompressible materials from sparse experimental data and physical laws. To achieve this goal, we propose a polyconvex neural additive model (PNAM) that enables us to express the hyperelastic model in a learnable feature space while enforcing polyconvexity. An upshot of this feature space obtained via the PNAM is that (1) it is spanned by a set of univariate basis functions that can be re-parametrized with a more complex mathematical form, and (2) the resultant elasticity model is guaranteed to fulfill the polyconvexity, which ensures that the acoustic tensor remains elliptic for any deformation. To further improve the interpretability, we use genetic programming to convert each univariate basis into a compact mathematical expression. The resultant multi-variable mathematical models obtained from this proposed framework are not only more interpretable but are also proven to fulfill physical laws. By controlling the compactness of the learned symbolic form, the machine learning-generated mathematical model also requires fewer arithmetic operations than its deep neural network counterparts during deployment. This latter attribute is crucial for scaling large-scale simulations where the constitutive responses of every integration point must be updated within each incremental time step. We compare our proposed model discovery framework against other state-of-the-art alternatives to assess the robustness and efficiency of the training algorithms and examine the trade-off between interpretability, accuracy, and precision of the learned symbolic hyperelastic models obtained from different approaches. Our numerical results suggest that our approach extrapolates well outside the training data regime due to the precise incorporation of physics-based knowledge.

KEYWORDS

hyperelasticity, interpretable model, polyconvex neural additive model, symbolic regression

1 | INTRODUCTION

Engineering analysis and computer simulations of solid behaviors often involve solvers that predict admissible solutions that fulfill a set of constraints. As explained in Kirchdoerfer and Ortiz,¹ these constraints can be (1) balance principles, which are often regarded as ground truth, and (2) constitutive laws, of which the legitimacy of employing such

supplemental constraints depends on the specific situation upon which the model is used.² For instance, while vehicle crash simulations for safety analysis^{3,4} and those for computer animation and gaming⁵ can both benefit from improving high fidelity, the demands for robustness, accuracy, efficiency, and risk tolerance in predictions are significantly different for these two applications. This difference often leads to different modeling choices, ranging from simplified models that offer speed and robustness, often at the expense of accuracy, to highly sophisticated models or multiscale constitutive updates from representative elementary volumes that exhibit higher fidelity, precision, and accuracy, even at the expense of efficiency.⁶

For soft materials that remain elastic while undergoing large isochoric deformation, modeling frameworks, such as hypoelasticity (cf. Truesdell,⁷ Green,⁸ and Freed et al.⁹) and hyperelasticity (cf. Ogden,¹⁰ Holzapfel et al.,¹¹ Mihai et al.,¹² Mooney,¹³ and Rivlin and Saunders¹⁴) are some of the more popular choices for these different applications. In the former framework, one may directly establish a relationship between a pair of stress and strain measures. For instance, one may compose an isotropic function that maps the left Cauchy–Green tensor $\mathbf{b}$ to the Cauchy stress for isotropic elastic materials.⁷ Hyperelastic models, on the other hand, provide an alternative strategy where one considers that the deformation increases the Helmholtz free energy, and the stress and its corresponding tangent stiffness are the Jacobian and the Hessian of this free energy with respect to the deformation measure. This latter approach provides a convenient and flexible way to fulfill thermodynamics consistency, enforce different symmetries, and, if desired, ensure the stability of the material models by enforcing properties (e.g., growth condition, convexity, polyconvexity) on the elastic energy functionals. As such, prior knowledge of the material behaviors, such as the lattice structure of a crystal (for symmetries) and the shape memory effect of certain alloys (for multiple potential wells), can be easily incorporated into the modeling process. If this prior knowledge can be incorporated by deducing the specific *form* of the energy functional, then the last remaining task to complete the model is to find the actual *parametrization* of the model that fulfills all constraints of both experimental data and prior physical knowledge.

In addition to hand-crafted mathematical expressions, a variety of alternatives, such as artificial neural networks (ANNs),^{15–19} Gaussian processes,^{20,21} and symbolic regression (SR),²² are often used to generate closure of the constitutive laws. In the former case, feedforward ANNs may provide the expressivity (the ability to fit any complex data) necessary to yield precise models.^{23,24} However, underfitting, overfitting, the lack of interpretability, and the incompatibility with physics constraints could all negatively impact the quality of the learned models. While the underfitting and overfitting issues can be circumvented with hyperparameter tuning, both the lack of interpretability and the incompatibility with known physics constraints are issues that make the resultant models not feasible for the intended engineering applications, especially of high consequences, such as patient-specific simulations or design of structural components for civil infrastructure systems. For example, Shen et al.²⁵ and Liang and Chandrashekhara²⁶ proposed training a hyperelastic energy functional for elastomeric foams. Liang and Chandrashekhara²⁶ use strain invariants as inputs for an ANN and conducted training based on the calculated stress as labeled data for supervised learning. Meanwhile, Le et al.²⁷ introduce neural network hyperelastic models to upscale constitutive responses for representative elementary volumes where the energy, stress, and stiffness are obtained from ANNs. There has been a rapidly growing body of work on hyperelastic models parametrized by ANNs trained using Sobolev norms,^{15,28,29} neural ordinary differential equations¹⁹ as well as multi-objective optimization problems that attempt to fulfill all data and physics constraints.^{18,30}

This article aims to formulate a feasible machine learning framework that can consistently generate hyperelastic models whose properties can be easily interpreted and fulfill all known physics constraints. A critical technical barrier we would like to overcome is the difficulty of determining the expression tree in the SR that simultaneously satisfies the polyconvexity of the learned model. While there have been works on generating polyconvex neural network models via neural networks, to the authors' best knowledge, this contribution is the first attempt to develop a machine learning algorithm to generate a polyconvex hyperelastic model expressed via mathematical expressions. To achieve this objective, we use a parameterization strategy similar to the neural additive model³¹ and the recently proposed quadratic extension,³² of which the feature space is spanned by univariate functions obtained from neural networks while improving the machine learning algorithm by introducing physics constraints and polyconvexity to ensure desirable properties of the learned hyperelastic models. To improve interpretability without comprising expressivity and accuracy, we introduce an additional step where we use SR on the feature space such that it can be approximated by basis functions expressed analytically. As polyconvexity can be guaranteed by energy functionals written in the specific additive form (cf. Hartmann and Neff³³), this feature enables us to express the resultant polyconvex hyperelastic model as a function of the features of the strain invariants.

1.1 | Reviews of physics constraints for elasticity models

Previous machine learning models have been trained with prior physical knowledge incorporated as constraints (e.g., Teichert et al.,³⁴ Liu et al.,¹⁶ Masi et al.,³⁵ Tac et al.,¹⁹ and Vlassis et al.²⁹). The basic strategy is similar to those used in physics-informed neural network paradigms considered seminal by many for solving partial differential equations where a variety of loss functions are employed to ensure that the learned solution satisfies physics constraints.^{36,37} However, as the training is designed to minimize, not eliminate, discrepancies, there is a possibility of violating these constraints when dealing with data not used during training, especially in the extrapolation regime.³⁸ Furthermore, enforcing multiple constraints into the loss function may lead to a multi-objective optimization problem where gradient conflicts among different objectives may further complicate the search for global optima,^{39,40} which is an NP-hard problem.⁴¹

Several neural network architectures have been proposed to explicitly incorporate all or a subset of physical laws by design for constitutive modeling.^{19,42–47} The by-design strategy can be selecting the optimal parameterization of input variables (e.g., using strain invariants instead of the strain tensor for isotropic materials) or modifying neural network architectures to preserve symmetry, invariance, and equivariance. By fulfilling the proposed physics constraints by design, the learned models that inherently fulfill the physics constraints are more robust, especially in the data-limited regime. In fact, this incorporation of physics constraints is in line with the history of hand-crafted hyperelastic models in which material symmetry has already been heavily leveraged to yield a specific form of mathematical expressions that reduces the number of independent variables, enforce symmetry and thermodynamics constraints,^{10,11,48} and induce desirable properties, such as polyconvexity^{33,48–50} and quasiconvexity.^{51,52} In this article, we will adopt this latter by-design strategy while leveraging the power of the neural additive model and symbolic regression to further improve the model obtained from the machine learning algorithm.

1.2 | Reviews on interpretable machine learning constitutive laws

Learnable parameters, such as weights and biases, parametrize the learned function obtained from training neural networks. As such, the current trend of increasingly deep and large neural networks often leads to significant challenges in interpreting and examining the *global* property of machine learning models.^{53,54} For instance, while it is possible to use a sampling technique to test the robustness of the learned model for a set of strain inputs against constraints, satisfying the physics constraints for a subset of data points only estimates the population loss. Model accuracy shown in the sampling test is only a necessary but not sufficient condition for generalizability.

An obvious strategy to circumvent this issue is to derive alternative parameterization that may lead to more compact mathematical expressions where analyses (such as calculating the acoustic wave speed and detecting the loss of ellipticity) typically performed on hand-crafted models can be conducted. In contrast to ANN-based methods, SR methods are free-form approaches where the equation form is discovered in a data-driven manner using gradient-free methods like genetic programming. The application of SR algorithms in data-driven mechanics has proven effective in discovering yield functions for plasticity.^{55–57} While initial attempts do not enforce physics knowledge, data augmentation via physical intuitions has shown improvements in the SR performance.⁵⁵ In a recent work by Abdusalamov et al.,²² a SR method for discovering hyperelastic materials is introduced, directly utilizing energy functionals. This approach offers advantages in terms of thermodynamics consistency. However, it can be computationally expensive due to the requirement for symbolic gradient calculations during optimization iterations.

The advantage of SR machine learning methods is their ability to provide explicit equation forms, which are often simpler than neural network operations. However, various challenges have hindered their popularity compared to ANN-based methods in mechanics. First, their lack of scalability, especially in multidimensional data settings, is attributed to the combinatorial nature of their search space. Second, incorporating mechanistic constraints, especially those related to gradient operations like ellipticity, is not a straightforward task in these algorithms. This difficulty arises from their use of gradient-free optimizers and the costly process of symbolic gradient calculations, in contrast to the efficiency of automatic differentiation methods used in ANN-based methods.

Sparse regression within a predefined library of modes⁵⁸ offers a potential method to bridge the gap between scalability and interpretability in model identification. Flaschel et al.⁵⁹ and Wang et al.⁶⁰ develop a material discovery formulation from a predefined library of material models. Their approach indirectly discovers a material model from displacement and force data over the boundary of the material sample. Similarly, Linka and Kuhl⁶¹ employ a library of modes inspired by classical constitutive models and prior physical knowledge to directly learn energy functionals from strain and stress data.

Each mode's contribution to the final prediction is trained using a gradient-based optimizer with automatic differentiation as the backbone algorithm. Nevertheless, such predefined modes may introduce significant bias in the modeling and might restrict the learning of complex modes not present in the library. For instance, Wang et al.⁶² parameterize the stress tensor as a polynomial function of the strain tensor. However, since physics constraints, such as material symmetry and thermodynamics constraints, are not explicitly enforced in the formulation, the model is not guaranteed to be compatible with these constraints.

Remark 1 (Alternative approaches for elasticity problems). The model-free or distance minimization method¹ is extended to finite deformation,^{63,64} eliminating the need for any model assumptions. Another model-free approach known as What-You-Prescribe-Is-What-You-Get (WYPIWYG)⁶⁵ is also employed. Following the WYPIWYG idea, methods based on spline shape functions are also introduced.^{66–68} Some studies formulate the learning of constitutive laws as a manifold learning problem, searching for the response surface rather than using conventional surrogate models for input-to-output mapping.^{69–73}

1.3 | Notations and organization of the remaining article

The remaining content of this article is organized as follows. In Section 2, we review crucial elements needed to construct our model structure, adhering to physical knowledge and mechanical properties of incompressible hyperelastic materials. These properties include isotropy, material objectivity, polyconvexity, and coercivity. Our model discovery method is outlined in Section 3, where we summarize our two-step approach. We then describe the specific structure of the neural network and SR algorithms used to ensure polyconvexity of the final discovered energy functional. In Section 4, we provide reduced forms of the proposed formulation to handle common experimental setups, which will be useful for calibrating the model from experimental data. To demonstrate the effectiveness of our framework, we find two symbolic models for real and synthetic data in Section 5. Additionally, we provide a discussion on the formal analysis of the discovered models in Section 6.

As for notations and symbols, bold-faced and blackboard bold-faced letters denote tensors (including vectors which are rank-one tensors); the symbol “ $\cdot$ ” denotes a single contraction of adjacent indices of two tensors (e.g., $\mathbf{a} \cdot \mathbf{b} = a_i b_i$ or $\mathbf{c} \cdot \mathbf{d} = c_{ij} d_{jk}$); the symbol “ $:$ ” denotes a double contraction of adjacent indices of tensors of rank two or higher (e.g., $\mathbb{C} : \boldsymbol{\varepsilon} = C_{ijkl} \varepsilon_{kl}$); the symbol “ $\otimes$ ” denotes a juxtaposition of two vectors (e.g., $\mathbf{a} \otimes \mathbf{b} = a_i b_j$) or two symmetric second-order tensors [e.g., $(\boldsymbol{\alpha} \otimes \boldsymbol{\beta})_{ijkl} = \alpha_{ij} \beta_{kl}$]. We also define identity tensors: $\mathbf{I} = \delta_{ij}$ and $\mathbb{I} = (\delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk})/2$, where δ_{ij} is the Kronecker delta. As for sign conventions, unless specified, tensile stress and dilative pressure are considered positive.

2 | HYPERELASTICITY FORMULATION

In this section, we establish the theoretical foundation upon which we construct our modeling structure, ensuring the incorporation of physical knowledge in the model. First, we delve into the kinematics of finite strain elasticity. Next, we review essential conditions for incorporating physically or empirically inspired constraints, such as polyconvexity. Finally, we derive the most general form of the energy functional, introducing unknown functions that will be parametrized by appropriate hypothesis classes in the following section.

2.1 | Kinematics of finite deformation

For completeness, we briefly review the kinematics of a continuum, which is the input of a path-independent elastic energy functional. Recall that the motion of a material point at the reference configuration can be described by the vector field $\boldsymbol{\phi}(\mathbf{X}) : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ which moves points $\mathbf{X} = X_I \mathbf{E}_I$ in the reference configuration Ω_0 (i.e., $\mathbf{X} \in \Omega_0$) to locations $\mathbf{x} = x_i \mathbf{e}_i$ in the current configuration Ω . The deformation gradient tensor $\mathbf{F}$, the primary measure of deformation, is the tangent operator of the motion $\boldsymbol{\phi}$, that is,

$$\mathbf{F} = \nabla^{\mathbf{X}} \boldsymbol{\phi}; \quad F_{IJ} = \frac{\partial \phi_i}{\partial X_J}. \quad (1)$$

The stretch vector λ^N along the direction of the unit vector $\mathbf{N}$ at $\mathbf{X} \in \Omega_0$ is defined as,

$$\lambda^N = \mathbf{F}\mathbf{N}; \quad \lambda_i^N = F_{iJ}N_J. \quad (2)$$

For practical reasons, one may prefer to use the right Cauchy–Green tensor $\mathbf{C}$ as the deformation measure,

$$\mathbf{C} = \mathbf{F}^T \mathbf{F}; \quad C_{IJ} = F_{iI}F_{iJ}. \quad (3)$$

The right Cauchy–Green tensor is symmetric and positive-definite which is more favorable for numerical calculations. Moreover, it is fully described with respect to the reference coordinate system which may ease analytical derivations by avoiding conversion between spatial and material coordinate systems. The first three principal invariants of the right Cauchy–Green tensor are calculated as follows,

$$I_1 = \text{tr}(\mathbf{C}) = C_{II}, \quad (4)$$

$$I_2 = \text{tr}(\text{adj}(\mathbf{C})) = \frac{1}{2}(I_1^2 - \text{tr}(\mathbf{C}^2)) = \frac{1}{2}(C_{II}C_{JJ} - C_{IJ}C_{JI}), \quad (5)$$

$$I_3 = \det(\mathbf{C}) = J^2, \quad (6)$$

where J is the Jacobian of the deformation gradient, that is, $J = \det(\mathbf{F})$, and $\text{tr}(\cdot)$ and $\det(\cdot)$ are trace and determinant operators, respectively. In these relations, adjugate operator is defined as $\text{adj}(\mathbf{C}) = \text{cof}(\mathbf{C})^T = \det(\mathbf{C})\mathbf{C}^{-1}$, where $\text{cof}(\cdot)$ is the cofactor operator. The importance of the tensor representation based on its invariants will be clarified later. For isotropic hyperelastic materials, these three invariants are sufficient to predict the elastic stored energy and the corresponding stress measure due to coaxiality.

2.2 | Physics constraints for isotropic elastic materials

Presumably, one may, for instance, develop constitutive theories by establishing relations between the deformation gradient and the first Piola–Kirchhoff stress $\mathbf{P}(\mathbf{F})$. However, caution must be exercised to avoid violating physical principles such as thermodynamic consistency.⁷⁴ Here, our focus is on the modeling of Green-elastic (hyperelastic) materials that postulate the existence of the Helmholtz free energy.

2.2.1 | Thermodynamic consistency for Green-elastic materials

Assuming that a material produces no entropy locally,⁷⁵ then the material is perfectly elastic. In this case, the second law of thermodynamics, which requires nonnegative internal dissipation, is fulfilled by the existence of the Helmholtz free energy, that is,

$$\mathcal{D} = \mathbf{P} : \dot{\mathbf{F}} - \dot{W}(\mathbf{F}) = \mathbf{P} : \dot{\mathbf{F}} - \frac{\partial W(\mathbf{F})}{\partial \mathbf{F}} : \dot{\mathbf{F}} \geq 0, \quad (7)$$

where W is the Helmholtz free energy, and by definition, since the dissipation is always zero for perfectly elastic materials, we have,

$$\mathbf{P} = \frac{\partial W(\mathbf{F})}{\partial \mathbf{F}}. \quad (8)$$

2.2.2 | Objectivity and frame indifference

The free energy functional W must be invariant with respect to any rigid rotation of the reference coordinate system. This is equivalent to saying,

$$W(\mathbf{F}) = W(\mathbf{QF}) \quad \forall \mathbf{Q} \in \text{SO}(3), \quad (9)$$

where $\mathbf{Q}$ is any arbitrary rotation tensor belonging to the special orthogonal group $\text{SO}(3)$, that is, $\mathbf{Q}^T \mathbf{Q} = \mathbf{I}$. This requirement can be satisfied if one defines the strain energy functional solely based on the right Cauchy–Green deformation tensor, that is, $\bar{W}(\mathbf{C}) = W(\mathbf{F})$, which leads to the following relation,

$$\exists \bar{W}(\mathbf{C}) \text{ s.t. } \mathbf{P} = \frac{\partial \bar{W}(\mathbf{C})}{\partial \mathbf{F}} = 2\mathbf{F} \frac{\partial \bar{W}(\mathbf{C})}{\partial \mathbf{C}} \Rightarrow \text{frame indifference.} \quad (10)$$

2.2.3 | Isotropy condition

An isotropic material exhibits the same strain-stress response under a symmetry transformation, that is,

$$W(\mathbf{F}) = W(\mathbf{FQ}^T) \quad \forall \mathbf{Q} \in \text{SO}(3). \quad (11)$$

For the free energy written in terms of $\mathbf{C}$, the isotropy of the constitutive responses implies that,

$$\bar{W}(\mathbf{C}) = \bar{W}(\mathbf{QCQ}^T) \quad \forall \mathbf{Q} \in \text{SO}(3) \Rightarrow \text{isotropic material.} \quad (12)$$

From the representation theorem for invariants^{76,77} one may show that this constraint is satisfied if the free energy is expressed as a function of only the principal invariants, that is,

$$\exists \psi(I_1(\mathbf{C}), I_2(\mathbf{C}), I_3(\mathbf{C})) \text{ s.t. } \mathbf{S} = 2 \frac{\partial \psi(I_1, I_2, I_3)}{\partial \mathbf{C}} \Rightarrow \text{Isotropic material,} \quad (13)$$

$$\frac{\partial \psi(I_1, I_2, I_3)}{\partial \mathbf{C}} = \left(\frac{\partial \psi}{\partial I_1} + I_1 \frac{\partial \psi}{\partial I_2} \right) \mathbf{I} - \frac{\partial \psi}{\partial I_2} \mathbf{C} + I_3 \frac{\partial \psi}{\partial I_3} \mathbf{C}^{-1}, \quad (14)$$

where $\mathbf{S}$ is the second Piola–Kirchhoff stress and $\mathbf{P} = \mathbf{FS}$.

2.2.4 | Incompressibility condition

In this article, we limit our focus to deducing the mathematical expression of elastic stored energy functionals for incompressible materials. A Material is considered incompressible when it only deforms in an isochoric manner, that is, the $\det(\mathbf{F}) = J = 1$. For practical purposes, the constitutive responses of many solids that exhibit significant isochoric deformation with negligible volumetric deformation (e.g., rubber), as well as liquids in room temperature (e.g., water), are idealized as incompressible.^{2,10,77} In these cases, one may introduce a scalar variable p (hydrostatic pressure) that serves as a Lagrange multiplier to account for the energy required to maintain this incompressibility constraint. Under this condition, the second Piola–Kirchhoff stress can be written as,

$$\psi = \psi^{\text{uc}}(I_1, I_2) - \frac{1}{2} p U(I_3) \Rightarrow \text{incompressible and isotropic material,} \quad (15)$$

$$\mathbf{S} = 2 \frac{\partial \psi^{\text{uc}}(I_1, I_2)}{\partial \mathbf{C}} - p \frac{\partial U}{\partial \mathbf{C}}, \quad (16)$$

$$\frac{\partial \psi^{\text{uc}}(I_1, I_2)}{\partial \mathbf{C}} = \left(\frac{\partial \psi^{\text{uc}}}{\partial I_1} + I_1 \frac{\partial \psi^{\text{uc}}}{\partial I_2} \right) \mathbf{I} - \frac{\partial \psi^{\text{uc}}}{\partial I_2} \mathbf{C}, \quad (17)$$

where $U(I_3)$ is the energy contribution to penalize the incompressibility constraint, that is, $|I_3| \rightarrow 1$.

There are various expressions for $U(I_3)$ such as $U(I_3) = I_3 - 1$ and $(J - 1)^2/2$ to enforce incompressibility.³³ In our case, we use $U(I_3) = I_3 - 1$ such that $\partial U / \partial \mathbf{C} = I_3 \mathbf{C}^{-1}$.⁷⁷

2.2.5 | Solution existence and uniqueness: Polyconvexity condition

In this section, we will show that the combination of convex functions of principal invariants is a subclass of polyconvex functions with respect to the deformation gradient, which leads to the existence of solutions for the elasticity boundary value problem.

Let us consider the functional $I(\mathbf{u})$ defined below,

$$I(\mathbf{u}) = \int_{\Omega_0} W(\nabla^X \mathbf{u}) d\Omega, \quad (18)$$

where $\mathbf{u}(\mathbf{X}) : \Omega_0 \rightarrow \mathbb{R}^3$ is the displacement vector field defined over the open set $\Omega_0 \subset \mathbb{R}^3$. The stationary points of this functional satisfy the equilibrium equations of nonlinear elasticity for a homogeneous body under zero body forces. However, an arbitrary free energy functional may not guarantee the existence of minimizers. Convexity of the free energy with respect to the deformation gradient guarantees this existence and uniqueness.⁷⁸ However, the uniqueness of the solution is too restrictive and not physical in bifurcation scenarios such as buckling. A less restrictive condition is polyconvexity, which is a sufficient condition for the global existence of the solution.⁷⁹

Definition 1 (Convexity). A function $f(\mathbf{x}) : \mathbb{D} \rightarrow \mathbb{R}$ is called convex with respect to its input argument when $f(\lambda \mathbf{x} + (1 - \lambda)\mathbf{y}) \leq \lambda f(\mathbf{x}) + (1 - \lambda)f(\mathbf{y})$ for any arbitrary $\mathbf{x}, \mathbf{y} \in \mathbb{D}$ and $0 < \lambda < 1$. The convexity condition becomes strict when the equality part becomes inadmissible. Here, $\mathbb{D}$ resembles the function domain, that is, $\mathbb{D} = \text{dom}(f)$, which is, in this article, a subset of real-valued vectors or tensors of rank two.

Definition 2 (Rank-one convexity and ellipticity). A twice differentiable free energy $W(\mathbf{F}) = \psi(\mathbf{C})$ leads to an elliptic system iff (if and only if) the Legendre–Hadamard or rank-one condition holds^{80–82}:

$$(\mathbf{M} \otimes \mathbf{m}) : \frac{\partial^2 W}{\partial \mathbf{F} \partial \mathbf{F}} : (\mathbf{M} \otimes \mathbf{m}) \geq 0 \quad \forall \mathbf{M}, \mathbf{m} \in \mathbb{R}^3, \quad (19)$$

where $\mathbf{M}$ and $\mathbf{m}$ are arbitrary vectors in the reference and current coordinate systems, respectively. The ellipticity (strongly elliptic) condition is satisfied if the inequality is strictly positive. Rank-one convexity is a sufficient condition for material stability and the well-posedness of the elasticity boundary value problem. Notice that the ellipticity requirement does not necessarily result in convexity of the strain energy functional, so multiple minimizers can be found.

Definition 3 (Quasiconvexity). Morrey's quasiconvexity condition reads,

$$\int_D W(\mathbf{F}_0 + \nabla^X \mathbf{v}(\mathbf{x})) d\mathbf{x} \geq |D| W(\mathbf{F}_0), \quad (20)$$

for all bounded open sets $D \in \mathbb{R}^3$ and all vector fields $\mathbf{v}(\mathbf{x})$. The quasiconvexity condition along with certain growth and continuity conditions result in the existence of minimizers.⁸³

Definition 4 (Polyconvexity). The free energy $\psi(\mathbf{F})$ is polyconvex iff it is convex with respect to $(\mathbf{F}, \text{adj } \mathbf{F}, \det \mathbf{F})$:

$$\exists \text{ convex } f \text{ s.t. } \psi(\mathbf{F}) = f(\mathbf{F}, \text{adj } \mathbf{F}, \det \mathbf{F}), \quad \forall \mathbf{F} \in \mathbb{R}^{3 \times 3} \Leftrightarrow \text{polyconvex } \psi(\mathbf{F}). \quad (21)$$

Polyconvexity results in quasiconvexity but not necessarily convexity.

Notice that all rank-one convexity, quasiconvexity, and polyconvexity conditions are sufficient conditions for the existence of minimizers. However, working with polyconvexity is preferable since, compared to the others, it is more straightforward to be applied as it has locality without any dependence on arbitrary objects (i.e., $\mathbf{m}$ and $\mathbf{M}$) other than the deformation gradients. In summary, the sequence of conditions presented below illustrates a progression where a condition on the right results from a condition on the left, with arrows indicating these implications. For example, quasiconvexity leads to rank-one convexity:

$$\boxed{\text{polyconvexity}} \rightarrow \text{quasiconvexity} \rightarrow \text{rank-one convexity} \rightarrow \text{existence of minimizers}. \quad (22)$$

Lemma 1. A subclass of polyconvex functions can be constructed by convex functions $f_1(\mathbf{F})$, $f_2(\text{adj}\mathbf{F})$, $f_3(\det \mathbf{F})$ in the additive fashion $\psi = f_1(\mathbf{F}) + f_2(\text{adj}\mathbf{F}) + f_3(\det \mathbf{F})$.

Proof. For brevity, we refer readers to Schröder and Neff⁴⁸ for details. ■

Lemma 2. If a tensor-valued function $h(\mathbf{A}) : \mathbb{R}^{3 \times 3} \rightarrow \mathbb{R}$ and a scalar-valued function $g(a) : \mathbb{R} \rightarrow \mathbb{R}$ are convex and $g(\cdot)$ is a non-decreasing function, then $g \circ h$ is convex.

Proof. For two arbitrary tensors $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{3 \times 3}$

$$(g \circ h)(\lambda \mathbf{A} + (1 - \lambda) \mathbf{B}) = g(h(\lambda \mathbf{A} + (1 - \lambda) \mathbf{B})), \quad (23)$$

$$\leq g(\lambda h(\mathbf{A}) + (1 - \lambda) h(\mathbf{B})) \quad \text{using convexity of } h \text{ and } g \text{ non-decreasing}, \quad (24)$$

$$\leq \lambda g(h(\mathbf{A})) + (1 - \lambda) g(h(\mathbf{B})) \quad \text{using convexity of } g. \quad (25)$$

Lemma 3. If $\psi_1(I_1)$, $\psi_2(I_2)$, and $\psi_3(I_3)$ are convex and non-decreasing functions, then $\psi = \psi_1(I_1) + \psi_2(I_2) + \psi_3(I_3)$ is polyconvex in $\mathbf{F}$.

Proof. $I_1(\mathbf{F})$, $I_2(\text{adj}\mathbf{F})$, and $I_3(\det \mathbf{F})$ are convex functions (see Appendix A for details). Based on the composition lemma $\psi_1(\mathbf{F})$, $\psi_2(\text{adj}\mathbf{F})$, and $\psi_3(\det \mathbf{F})$ are convex as well. Hence, ψ is polyconvex in $\mathbf{F}$. ■

In this work, the polyconvexity constraint is guaranteed by choosing parameterized functions for $\psi_1(I_1)$ and $\psi_2(I_2)$ which are convex and nonnegative; this choice will be clarified later. We use an additive structure in the unconstrained part of the free energy in Equation (15), that is, $\psi^{\text{uc}}(I_1, I_2) = \psi_1(I_1) + \psi_2(I_2)$.

2.2.6 | Coercivity (growth) conditions

As mentioned earlier, certain coercivity (growth) conditions along with polyconvexity result in the existence of global minimizers. These conditions ensure that infinite strains result in unbounded stresses. The growth condition is satisfied if,

$$\psi(\mathbf{F}) \geq \alpha (||\mathbf{F}||^p + ||\text{adj}\mathbf{F}||^q + (\det \mathbf{F})^r) + \beta, \quad (26)$$

for $\alpha > 0$, $\beta > 0$, $p \geq 2$, $q \geq \frac{p}{p-1}$, and $r > 1$. The matrix norm is $||\mathbf{F}||^2 = \text{tr}(\mathbf{F}^T \mathbf{F})$. For incompressible materials, this inequality can be simplified to,

$$\psi(\mathbf{F}) \geq \alpha (I_1^p + I_2^q) + \alpha + \beta. \quad (27)$$

In this work, we do not explicitly enforce the coercivity condition, however, we will check the admissibility of the discovered equations according to the coercivity condition.

Remark 2. Compressing material towards zero volume should require an infinite amount of energy. Thus one needs to enforce $\psi \rightarrow \infty$ when $J \rightarrow 0^+$ if the material is compressible.

2.2.7 | Stress free and positivity conditions

The free energy functional should remain nonnegative for all possible deformations. For practical purposes, it is often desirable to assume that the initial reference configuration does not exhibit any residual stress when $\mathbf{C} = \mathbf{F} = \mathbf{I}$ (cf. Hoger⁸⁴). These two conditions can be expressed as,

$$W(\mathbf{F}) \geq 0, \quad (28)$$

$$W(\mathbf{F})|_{\mathbf{F}=\mathbf{I}} = 0, \quad \mathbf{S}(\mathbf{F})|_{\mathbf{F}=\mathbf{I}} = \mathbf{0}. \quad (29)$$

The nonnegative condition can be satisfied by selecting parameterizations that lead to nonnegative free energy functionals. These parameterizations will be elaborated further in latter sections.

Adopting a similar cancellation/counterbalancing strategy used in As'ad et al.⁴⁵ and Chen and Guilleminot,⁴⁶ we formulate the energy functional such that a counterbalance term is added to ensure that the stress-free condition corresponds to the undeformed reference configuration by construction. This is done as follows. Consider an energy functional $\psi(I_1, I_2)$ that does not necessarily meet the stress-free condition in its undeformed state. According to Equation (14), the non-zero stress state is as follows,

$$\left(\frac{\partial \psi}{\partial I_1}(I_1 = 3, I_2 = 3) + 2 \frac{\partial \psi}{\partial I_2}(I_1 = 3, I_2 = 3) \right) \mathbf{I}. \quad (30)$$

By introducing an extra energy component to the energy functional, which operates independently of $\psi(I_1, I_2)$, we can generate an equivalent stress value with an opposite sign. This approach effectively neutralizes the non-zero stress initially observed. For the incompressible scenario, an effective approach is to formulate the additional energy contribution solely as a function of I_3 , which remains unaffected by the deformation state (namely, I_1 and I_2). A practical example of this is setting $\psi_{p_0} = \alpha_0(I_3 - 1)$, where the parameter α_0 is adjustable. This adjustment ensures that the total energy functional, represented by $\psi(I_1, I_2) - \psi_{p_0}$, equals zero in the undeformed state:

$$\alpha_0 = \frac{\partial \psi}{\partial I_1}(I_1 = 3, I_2 = 3) + 2 \frac{\partial \psi}{\partial I_2}(I_1 = 3, I_2 = 3). \quad (31)$$

2.2.8 | Final form of hyperelastic energy functional for model discovery

To fulfill the physical constraints and assumptions we listed in the previous sections, we limit the mathematical expression of the learned model to be in the following form,

$$\psi^{\text{final}}(I_1, I_2, I_3) = \underbrace{\psi_1(I_1) + \psi_2(I_2)}_{\psi(I_1, I_2)} - \underbrace{\frac{1}{2}p(I_3 - 1)}_{\text{incompressibility}} - \underbrace{\alpha_0(I_3 - 1)}_{\psi_{p_0} \text{ zero stress}} - \underbrace{(\psi_1(3) + \psi_2(3))}_{\psi_0 \text{ zero energy}}. \quad (32)$$

In particular, we hypothesize that the additive decomposition of the unconstrained energy functional, $\psi(I_1, I_2) = \psi_1(I_1) + \psi_2(I_2)$ is valid. As discussed in Agarwal et al.,³¹ restricting the energy functional to take the form of a linear combination of input variables may reduce the expressivity. This assumption, however, can also reduce the difficulty of the symbolic regression by reducing the dimensionality of the learned functions. This trade-off will be further demonstrated in the numerical examples.

As mentioned previously, by restricting the learned model to take the form of Equation (32), one may enforce the zero-stress condition by calculating the parameter α_0 such that the four terms in Equation (32) canceling out each other at the reference configuration, that is,

$$\alpha_0 = \left. \frac{d\psi_1}{dI_1} \right|_{I_1=3} + 2 \left. \frac{d\psi_2}{dI_2} \right|_{I_2=3}. \quad (33)$$

One can straightforwardly show that the constructed strain energy satisfies the zero-stress condition at the undeformed state, that is, when $I_1 = I_2 = 3$, $I_3 = 1$, and $p = 0$.

We will formulate the supervised learning problem such that the resultant expression is compatible with the form in Equation (32). As we will discuss in next section, this setting limits the expression trees included in the combinatorial optimization and hence reduces the difficulty of the SR problem.

3 | INTERPRETABLE MODEL DISCOVERY COMPATIBLE WITH PHYSICS CONSTRAINTS

Deriving a physics-constrained model consistent with the proposed form in Equation (32) directly through a single SR algorithm remains challenging for the following reasons. First, it is challenging to straightforwardly restrict each expression tree to yield a convex function over the entire data range and beyond. Ensuring convexity can be complex and may lead to sub-optimal results. Second, as illustrated in Figure 1, the stress objective function requires access to the gradients of the energy functional. This requirement can significantly increase the computational time of SR, especially when dealing with deep expression trees with many terms. Finally, there is a high likelihood of encountering expression trees with unstable functions, thus further complicating the SR training process.

To overcome these well-known challenge for SR, we modify our recently developed interpretable data-driven approach for uncovering yield surfaces³² such that the learning process now takes into account physics constraints. Our divide (step 1 in Figure 1) and conquer (step 2 in Figure 1) strategy still aims to merge the scalability of training neural-network models with the interpretability offered by SR. However, instead of deducing a model consistent with just the data, we establish a theoretical link between this computational approach and the classical discovery of a polyconvex energy functional for isotropic hyperelasticity, which features an additive structure, as stated in Lemma 3.

3.1 | Polyconvex neural additive model

Here, we introduce the polyconvex neural additive model (PNAM). Each shape function $\psi_i(I_i)$ in the PNAM is parameterized by a separate neural network which must be convex, positive, and non-decreasing. A particular class of neural networks known as input-convex neural networks (ICNNs) designed by Amos et al.⁸⁵ fulfills all these properties.

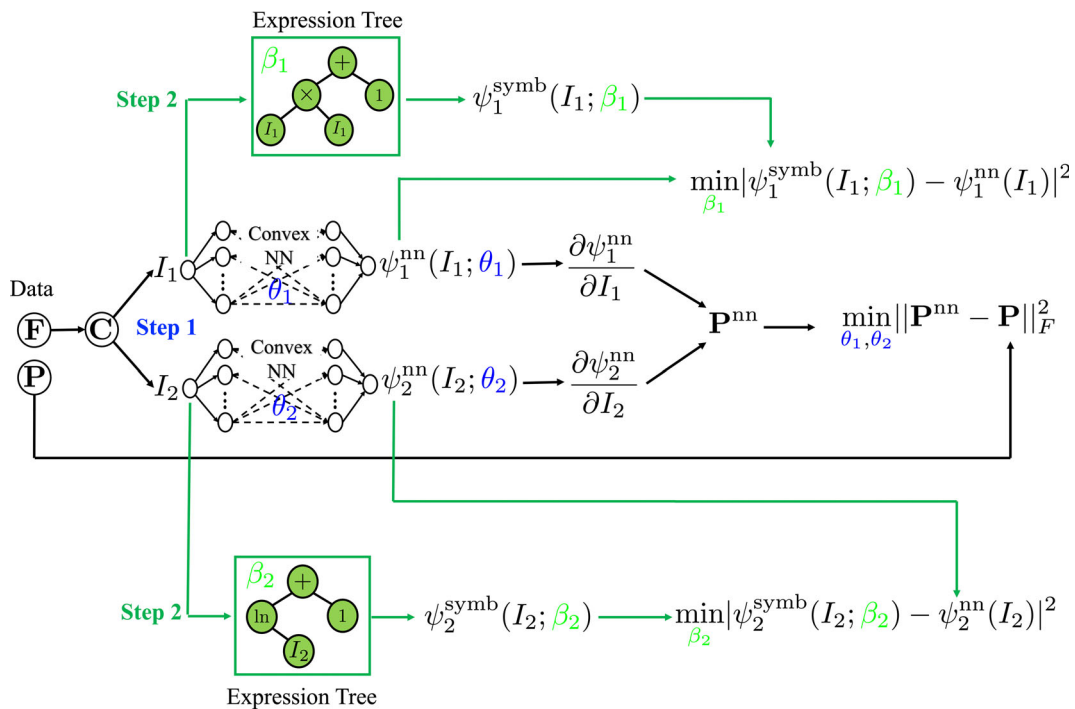


FIGURE 1 Proposed interpretable hyperelastic model discovery via a two-step approach. In **step 1**, a PNAM is trained using a **scalable** gradient-based optimizer with multivariate hyperelasticity data. In **step 2**, an evolutionary-based optimizer finds **interpretable** symbolic equations by conducting SR over univariate functions in parallel. The data are assumed to be available in the form strain-stress pair (F, P) . Trainable neural network parameters are shown in blue, that is, θ_1, θ_2 . The unknown structure of the expression trees are shown in green, that is, β_1, β_2 , which are found by an evolutionary-based algorithm.

The forward operations to obtain $\psi^{\text{ICNN}}(I; \theta)$ in an ICNN are as follows,

$$\mathbf{h}_1 = g_0(I\mathbf{v}^0 + \mathbf{b}^0); \quad \mathbf{v}^0, \mathbf{b}^0, \mathbf{h}^1 \in \mathbb{R}^{l_1}, \quad (34)$$

$$\mathbf{h}_{j+1} = g_j(\mathbf{W}^j \mathbf{h}_j + I\mathbf{v}^j + \mathbf{b}^j); \quad \mathbf{v}^j, \mathbf{b}^j, \mathbf{h}_j \in \mathbb{R}^{l_{j+1}}, \mathbf{W}^j \in \mathbb{R}^{+l_j \times l_{j+1}}, 1 \leq j < L, \quad (35)$$

$$\psi^{\text{ICNN}} = h_L, \quad (36)$$

where, as shown in Figure 2, L is the number of hidden layers, g_j are activation functions for each layer, h_j are the hidden representation of data at each layer, and $\mathbf{W}^j$, $\mathbf{b}^j$, and $\mathbf{v}^j$ are unknown, trainable parameters of the neural network.

Proposition 1. *The function $\psi^{\text{ICNN}}(I; \theta)$ is convex with respect to the input I if all $\mathbf{W}^j$ are nonnegative, and all activation functions g_j are convex.*

Proof. For the sake of completeness, we provide a straightforward proof. The first step in Equation (34) establishes the convexity of each component in the vector $\mathbf{h}_1$ (see Lemmas 9 and 2). To verify the proposition, we must demonstrate the convexity of each component of $\mathbf{h}_{j+1}$ with respect to I for $j \geq 1$. Equation (35) confirms this result, as each component of the resulting vector $\mathbf{W}^j \mathbf{h}_j$ is convex (see Lemma 4). ■

Lemma 4. *A weighted sum of convex functions $h_i(x)$ with nonnegative weights w_i results in a convex function in $x \in \mathbb{R}$.*

Proof. Let us define $g(x) = w_i h_i(x)$ with constants $w_i \in \mathbb{R}^+$ and convex functions h_i , for any arbitrary x_1, x_2 , and λ we have, with summation over i ,

$$g(\lambda x_1 + (1 - \lambda)x_2) = w_i h_i(\lambda x_1 + (1 - \lambda)x_2), \quad (37)$$

$$\leq w_i [\lambda h_i(x_1) + (1 - \lambda)h_i(x_2)] \quad \text{via convexity of } h_i \text{ and positivity of } w_i, \quad (38)$$

$$= \lambda w_i h_i(x_1) + (1 - \lambda)w_i h_i(x_2), \quad (39)$$

$$= \lambda g(x_1) + (1 - \lambda)g(x_2). \quad (40)$$

■

Remark 3. In this study, we utilize the softplus and softplus2 activation functions, both of which are convex, smooth, and non-decreasing. These functions are defined as follows,

$$\text{softplus}(x) = \ln(1 + \exp(x)), \quad (41)$$

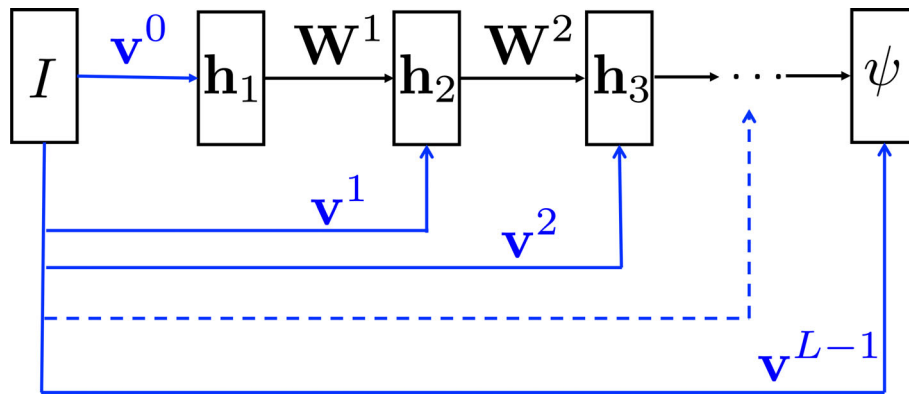


FIGURE 2 Input-convex neural network which is guaranteed to be convex with respect to the input I , non-decreasing, and nonnegative by construction.

$$\text{softplus2}(x) = \text{softplus}(x)^2. \quad (42)$$

Previous works^{18,45,86} demonstrate the application of these activation functions in the context of ICNN modeling of hyperelastic energy functionals.

3.2 | Objective function for neural network training

Each shape function $\psi_i(I_i)$ in Equation (32) is parametrized by a different ICNN $\psi_i^{\text{ICNN}}(I_i; \theta^i)$ where each θ^i concatenates all trainable parameters associated with the i -th ICNN. In this work, we assume we only have access to strain and stress measurements stored in a dataset $\mathcal{D} = \{\mathbf{F}^k, \mathbf{P}^k\}_{k=1}^{N_{\text{data}}}$ for N_{data} number of measurements. Notice that in realistic experiments, access to values of the energy functional or its Hessian is not possible although one may obtain them from numerical simulations.

To calibrate parameters $\theta = \{\theta^i\}_{i=1}^2$, we minimize the following loss function,

$$\theta = \arg \min_{\theta} \sum_{k=1}^{N_{\text{data}}} \left\| \frac{\partial \psi^{\text{tot}}(I_1, I_2; \theta)}{\partial \mathbf{F}} \Big|_{\mathbf{F}^k} - \mathbf{P}^k \right\|_F^2, \quad (43)$$

where $\|\cdot\|_F$ is the Frobenius norm for a second-order tensor. We use the ADAM algorithm⁸⁷ to minimize this single objective function.

3.3 | Symbolic regression

Symbolic regression aims to identify a mathematical expression that optimally fits a provided dataset without predefining the expression's form. The set of possible expressions is typically defined by establishing a range of mathematical operators, functions, variables, and constants, which are then efficiently represented using binary trees (refer to Figure 3). Genetic programming is a widely adopted stochastic optimization technique for exploring the combinatorial space encompassing all conceivable mathematical expressions.^{88–90} Moreover, recent advancements have been made utilizing deep reinforcement learning methods, offering alternative and efficient means for discrete searches within the domain of tree data structures.^{91,92}

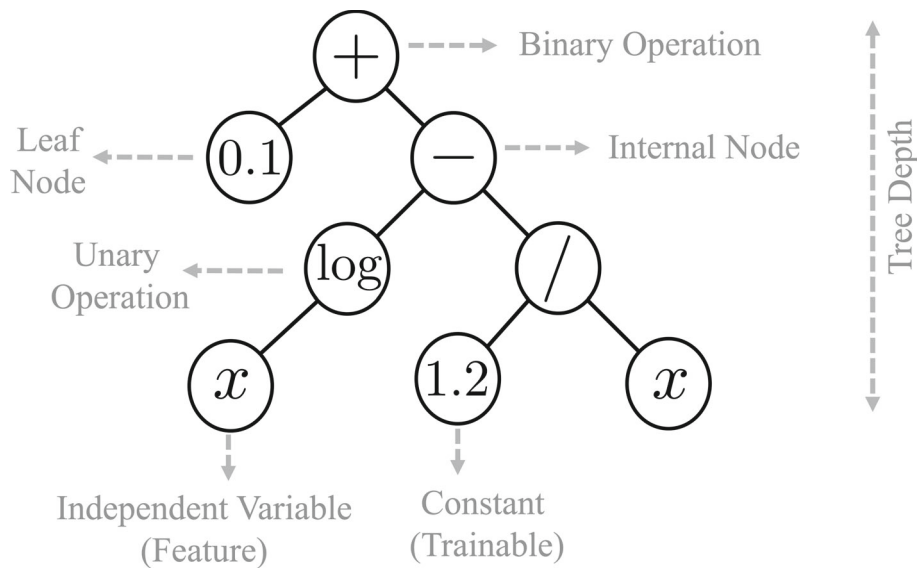


FIGURE 3 Binary expression tree for equation representation. The program $\log(x) - \frac{1.2}{x} + 0.1$ is represented by an expression tree with a depth 3 and size of 8 (the total number of nodes).

Genetic programming employs a population of candidate solutions, randomly generated at the algorithm's start, to represent the space of potential expressions. Each individual candidate solution is described as a binary expression tree (refer to Figure 3), where the leaves symbolize input variables or constants, and the internal nodes denote mathematical operations or functions. The algorithm evaluates the fitness of each candidate solution by comparing its output values to the target values. In this study, we use the mean square error (MSE) as the fitness measure, that is,

$$\text{MSE}(y^{\text{model}}, y^{\text{data}}) = \frac{1}{N_{\text{data}}} \sum_{i=1}^{N_{\text{data}}} (y_i^{\text{model}} - y_i^{\text{data}})^2. \quad (44)$$

The genetic programming algorithm utilizes an iterative process that evolves the population of candidate solutions through three main operations: selection, crossover, and mutation. This process draws inspiration from natural selection. In the selection step, higher-performing individuals from the current population are chosen based on their fitness scores. Crossover (shown in Figure 4A) combines the genetic information of two individuals to generate offspring with characteristics inherited from both parents. On the other hand, mutation (depicted in Figure 4B) involves randomly altering some genetic components of the individual, thereby introducing new variations into the population.

The genetic programming algorithm creates improved candidate solutions iteratively until it finds a satisfactory mathematical expression that fits the data well. During SR optimization, there is a tradeoff between model complexity and expressivity. Striking a balance between the complexity and expressivity of the discovered analytical expression is vital to enhance interpretability and mitigate overfitting. Users can choose the best equation for their requirements, considering desired accuracy and simplicity.

Definition 5 (Complexity score in SR). The assessment of a symbolic equation's complexity is often qualitative due to the lack of a universally agreed-upon definition. In this study, we adopt the complexity measure proposed in Cranmer,⁹³ which uses the number of nodes in the expression tree as the complexity score. While assigning varying weights to different node types is feasible, such as considering $\exp(\cdot)$ as more complex than $+$, we refrain from incorporating such weightings in our analysis.

Symbolic equations used during inference have a smaller memory footprint compared to neural networks, making them more portable. Research has shown that equations discovered by SR generalize well beyond the training data support.⁹⁴ However, when dealing with multi-dimensional vector-valued or tensor-valued functions, SR becomes notably

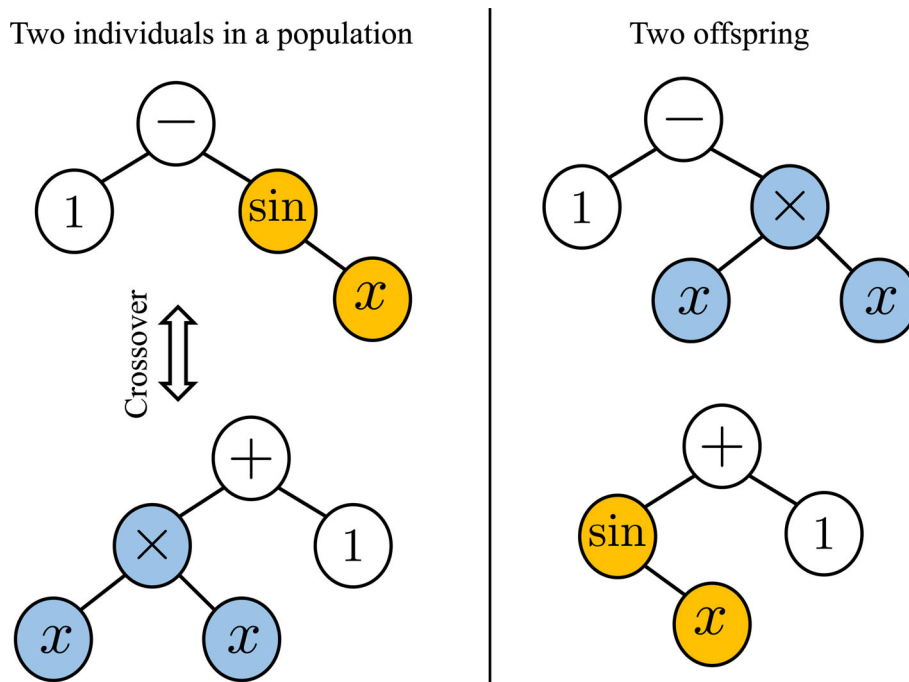


FIGURE 4 (A) Crossover and (B) mutation operations in an evolutionary-based SR algorithm.

more challenging due to the complex combinatorial nature of the optimization problem involved in searching for the appropriate mathematical expression.⁹⁵ Our proposed method overcomes this concern, as mentioned earlier, by extracting a symbolic equation for each single-variable to single-variable ICNN submodel. This feature facilitates parallel computing, allowing SR algorithms to run simultaneously without any issues.

4 | REDUCED FORMS FOR EXPERIMENTAL CALIBRATION

Motivated by real experimental setups, we calibrate model parameters for uniaxial extension (UE), equibiaxial extension (EBE), and pure shear (PS) loading conditions in the incompressible limit. In this section, we summarize reduced forms of equations for a general incompressible, isotropic free energy functional $\psi(I_1, I_2)$ associated with each of these boundary conditions.

4.1 | Uniaxial extension

The easiest experimental setup is UE where only one side of the cubic sample is deformed along its normal axis, for example, the X -axis as shown in Figure 5A. Based on this condition, we only control the stretch $\lambda_X = \lambda$, but, due to the symmetry induced by isotropy, the other two stretches are equal, and, hence the incompressibility condition leads to $\lambda_Y = \lambda_Z = \lambda^{-0.5}$. Under these boundary conditions, only P_{11} becomes non-zero which can be found in Equations (16) and (32). In summary, the following reduced form is obtained for the UE loading condition,

$$\mathbf{F} = \text{diag}(\lambda, \lambda^{-0.5}, \lambda^{-0.5}), \quad (45)$$

$$\mathbf{P} = \text{diag}(P_1, 0, 0), \quad (46)$$

$$P_1 = 2 \left(\frac{\partial \psi}{\partial I_1} + \frac{1}{\lambda} \frac{\partial \psi}{\partial I_2} \right) \left(\lambda - \frac{1}{\lambda^2} \right). \quad (47)$$

4.2 | Equibiaxial extension

EBE is similar to UE where the extension loading is applied over two faces of the cubic sample as depicted in Figure 5B. Based on this condition, we control the stretches $\lambda_X = \lambda_Y = \lambda$, while the other stretch is constrained to $\lambda_Z = \lambda^{-2}$, due

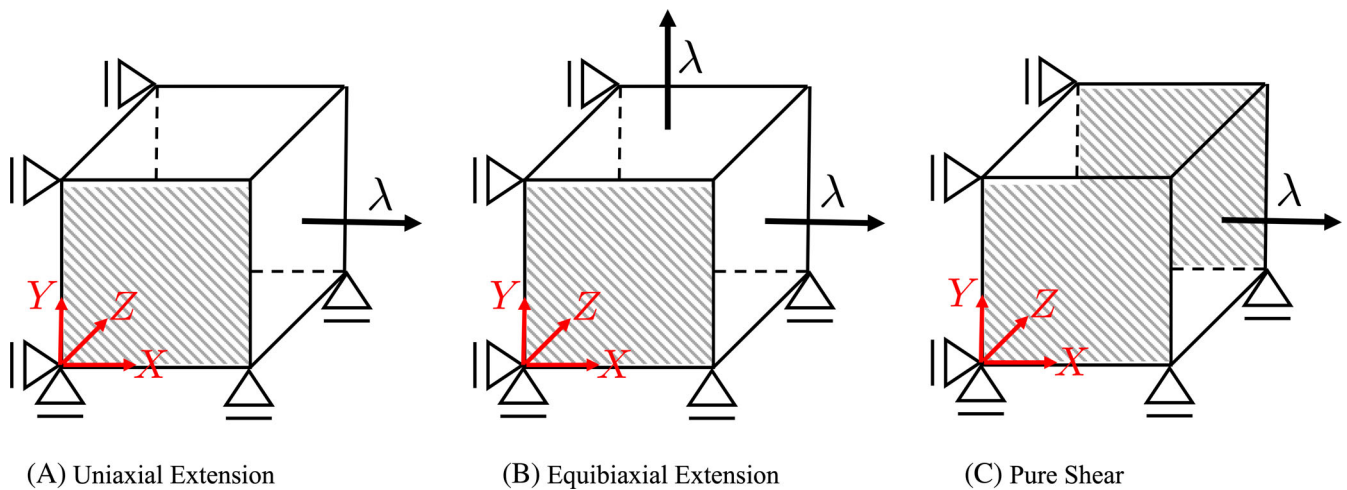


FIGURE 5 Different loading conditions for model calibration and training.

to incompressibility. Under these boundary conditions, only P_{11} and P_{22} become non-zero and the same because of the symmetry induced by isotropy. In summary, the following reduced form is obtained for EBE loading condition,

$$\mathbf{F} = \text{diag}(\lambda, \lambda, \lambda^{-2}), \quad (48)$$

$$\mathbf{P} = \text{diag}(P_1, P_1, 0), \quad (49)$$

$$P_1 = 2 \left(\frac{\partial \psi}{\partial I_1} + \lambda^2 \frac{\partial \psi}{\partial I_2} \right) \left(\lambda - \frac{1}{\lambda^5} \right). \quad (50)$$

4.3 | Pure shear

PS loading condition can be achieved by restricting UE from any movement in the Z-axis, that is, $\lambda_Z = 1$ (see Figure 5C.) We control the stretch $\lambda_X = \lambda$, while the other stretch is constrained to $\lambda_Y = \lambda^{-1}$. Under these boundary conditions, only P_{11} and P_{33} become non-zero. In summary, the following reduced form is obtained for PS loading condition,

$$\mathbf{F} = \text{diag}(\lambda, \lambda^{-1}, 1), \quad (51)$$

$$\mathbf{P} = \text{diag}(P_1, 0, P_3), \quad (52)$$

$$P_1 = 2 \left(\frac{\partial \psi}{\partial I_1} + \frac{\partial \psi}{\partial I_2} \right) \left(\lambda - \frac{1}{\lambda^3} \right), \quad (53)$$

$$P_3 = 2 \left(\frac{\partial \psi}{\partial I_1} + \lambda^2 \frac{\partial \psi}{\partial I_2} \right) \left(1 - \frac{1}{\lambda^2} \right). \quad (54)$$

5 | NUMERICAL EXAMPLES

We demonstrate the application of our proposed method in discovering hyperelastic constitutive models through two numerical examples. First, in Section 5.1, we utilize experimental data from the literature to identify a material model. We compare the simplicity and accuracy of our discovered model with recent efforts in the field. Additionally, we assess and discuss the extrapolation capabilities of our method compared to a standard neural network approach.

In the second problem, in Section 5.2, we showcase the method's potential for modeling more complex materials. To showcase this ability, we generate virtual experimental data using finite element simulations for a randomly synthesized particle-reinforced composite.

Remark 4. Our framework is implemented using PyTorch⁹⁶ and PySR⁹³ open-source packages for neural network and SR training, respectively.

5.1 | Treloar's data for vulcanized rubber

In this example, we demonstrate the application of the proposed scheme in finding a hyperelastic energy functional for the well-known Treloar's experimental data for vulcanized rubber.⁹⁷ We then compare the accuracy and simplicity of the discovered model with a reference model. Additionally, we highlight the significance of physics augmentation and interpretability for achieving better generalization compared to black-box neural network models.

We use 90% of Treloar's data to train the model discovery algorithm and the remaining 10% to validate. Training-related hyperparameters and setups are discussed in Appendix B.1. The proposed algorithm discovers the following equation,

$$\psi = c_{11}I_1 + c_{12} \exp(c_{13}I_1) + c_{21}I_2 + c_0, \quad (55)$$

where $c_{11} = 0.1502$, $c_{12} = 0.0771$, $c_{13} = 0.0665$, $c_{21} = 0.0035$, and $c_0 = 261.5397$. In Table 1, we compare its accuracy with another equation found by a recently introduced SR algorithm in Abdusalamov et al.,²²

$$\psi_{\text{ref}} = \sqrt{0.93296 \exp(0.08031I_1) + \sqrt{I_1 - 0.080316} + (0.0232113I_1 + 0.021633)I_1}. \quad (56)$$

In this table, the R^2 -score is chosen as the measure of accuracy over the entire data. The R^2 -score between predicted values $\{y_i\}_{i=1}^{N_{\text{data}}}$ and ground-truth values $\{y_i^*\}_{i=1}^{N_{\text{data}}}$ is defined as follows,

$$R^2(y^*, y) = 1 - \frac{\sum_{i=1}^{N_{\text{data}}} (y_i^* - y_i)^2}{\sum_{i=1}^{N_{\text{data}}} (y_i^* - \text{mean}(y^*))^2}. \quad (57)$$

Based on this measure, our discovered equation shows improved performance in the EBE dataset, with slight enhancements observed in the UE and PS datasets as well. Considering the depth of the expression tree and square root operators as more complex, our discovered equation is qualitatively simpler. Additionally, in our work, we use 90% of the data to find the equation, whereas the reference utilizes the entire dataset.

The trained ICNN model $\psi_1(I_1)$ for the PNAM solution $\psi(I_1, I_2) = \psi_1(I_1) + \psi_2(I_2)$ is depicted in Figure 6 by the orange curve. The result for $\psi_2(I_2)$ is omitted for brevity as it exhibits characteristics of a linear function, making it straightforward to handle. As mentioned earlier, in step 2 of the proposed method, we apply the SR algorithm to extract an equivalent symbolic equation for the learned model. The SR training process takes less than 3 min on a MacBook Pro laptop with a quad-core Intel Core i5 processor running at 1.4 GHz and using 16 GB of RAM. The SR algorithm lists several candidates ranked by their accuracy/simplicity, with more accurate candidates being less simple in forms.

In Figure 7, we present accuracy versus complexity plots for both the function $\psi_1(I_1)$ and its gradient $\psi_1'(I_1)$. While the gradient information is not employed in the fitness measure, we recommend considering this information for model

TABLE 1 Comparing R^2 -scores of our discovered equation and the reference equation for Treloar's data.

Equation	Uniaxial extension	Equibiaxial extension	Pure shear
This work (Equation 55)	0.9983	0.9904	0.9963
Reference (Equation 56)	0.9979	0.8692	0.9887

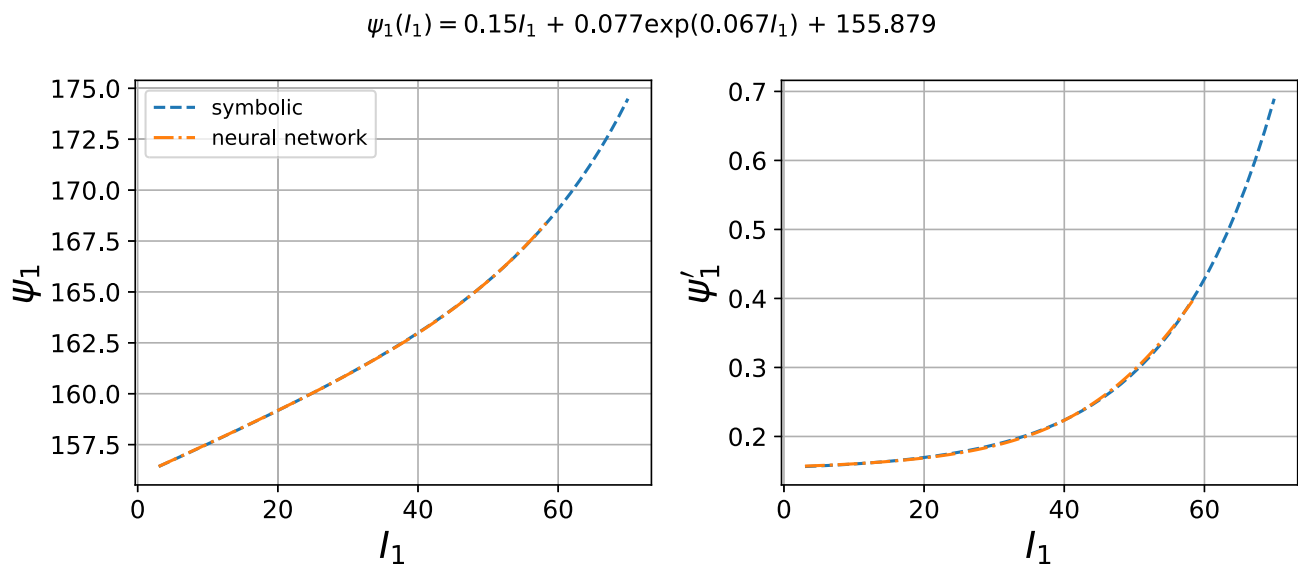


FIGURE 6 Discovered symbolic function from the trained neural network representation. The discovered function and its gradient are shown in left and right plots, respectively. Only data within the training range (orange curves) are used for the SR.

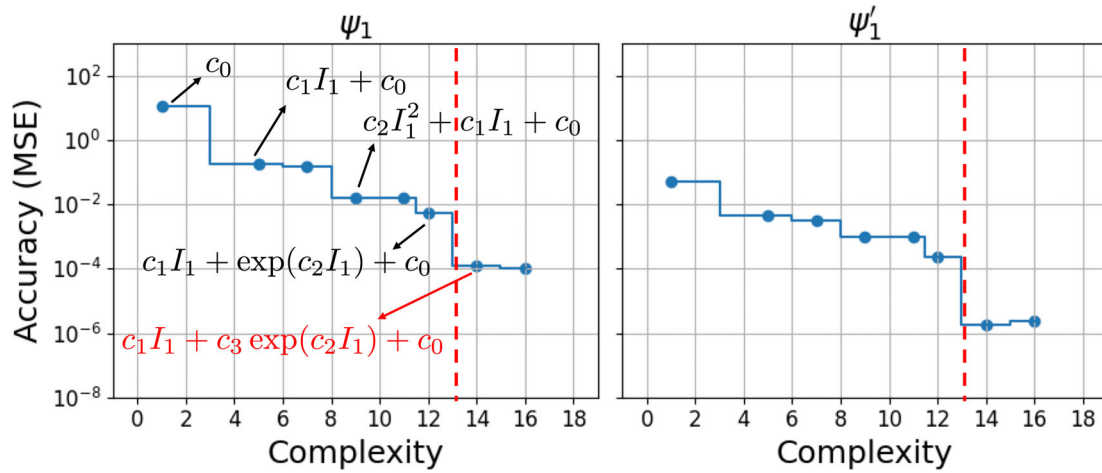


FIGURE 7 Complexity (parsimony) versus accuracy of the discovered function and its gradient for Treloar's data. More accurate equations tend to be more complex (i.e., contain more terms). Achieving a balance between accuracy and simplicity is essential for both the discovered function and its gradient.

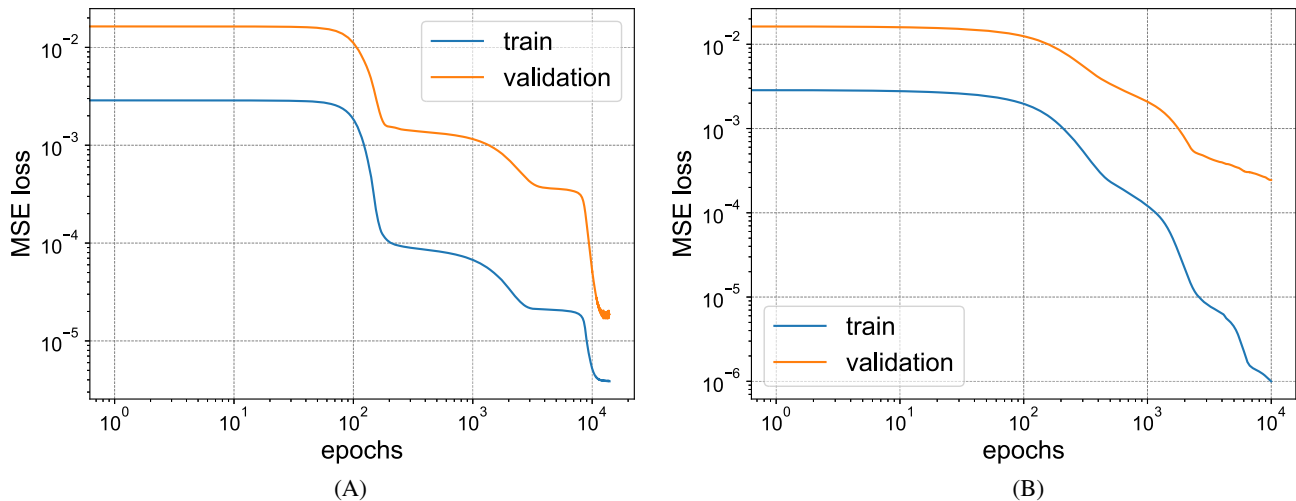


FIGURE 8 Training and validation losses for model calibration in step 1 where a neural network solution is found. 90% of the data is utilized for training, and 10% is utilized for validation. (A) PNAM. (B) Vanilla ANN.

selection during the post-processing step among all available candidates. This consideration is crucial because the accuracy of the gradient directly impacts the final stress predictions. We employ these selection criteria based on two factors: candidates must (1) have errors below a specified threshold and (2) exhibit an abrupt accuracy improvement compared to their simpler counterparts (see the red vertical line in Figure 7). A comparison between the selected equation and the ICNN model is illustrated in Figure 6, with the symbolic equation depicted by the blue curve. A good agreement between the ICNN model and the symbolic equation is observed. Moreover, the behavior of the symbolic function beyond the ICNN training data appears to be reasonable.

We further compare the proposed scheme with the solution obtained from a vanilla ANN. In the case of the vanilla ANN, we substitute $\psi(I_1, I_2)$ in Equation (32) with a conventional multilayer perceptron (MLP), accepting both I_1 and I_2 in its input layer. The model accuracy during the optimization iterations in step 1 for the PNAM and vanilla ANN is illustrated in Figure 8 for both the training and validation data. While the final loss value of the training curve for the vanilla ANN is lower than that for the PNAM, the larger discrepancy between the validation and training curves for the vanilla ANN indicates overfitting. This overfitting may lead to less reliable predictions in the extrapolation regime.

To assess both models' capabilities, we compare their R^2 -scores for all three datasets, as shown in Figure 9. Both models exhibit satisfactory R^2 -scores across all available data. The PNAM demonstrates better performance for UE and EBE, while the vanilla ANN shows slightly better accuracy in the case of the PS dataset. However, it is essential to note that we observe non-physical behavior for the vanilla ANN in the case of the EBE dataset when moving beyond the training data regime.

In order to highlight the extrapolation power resulting from physics augmentation, simplicity, and interpretability of the proposed scheme, we compare predictions of the symbolic equation discovered in this work (Equation 55), the one found in the reference (Equation 56), and the vanilla ANN in Figure 10, where we extend far beyond the training data ranges. Given that we lack access to real data at these ranges, we cannot assess the accuracy of the symbolic equations.

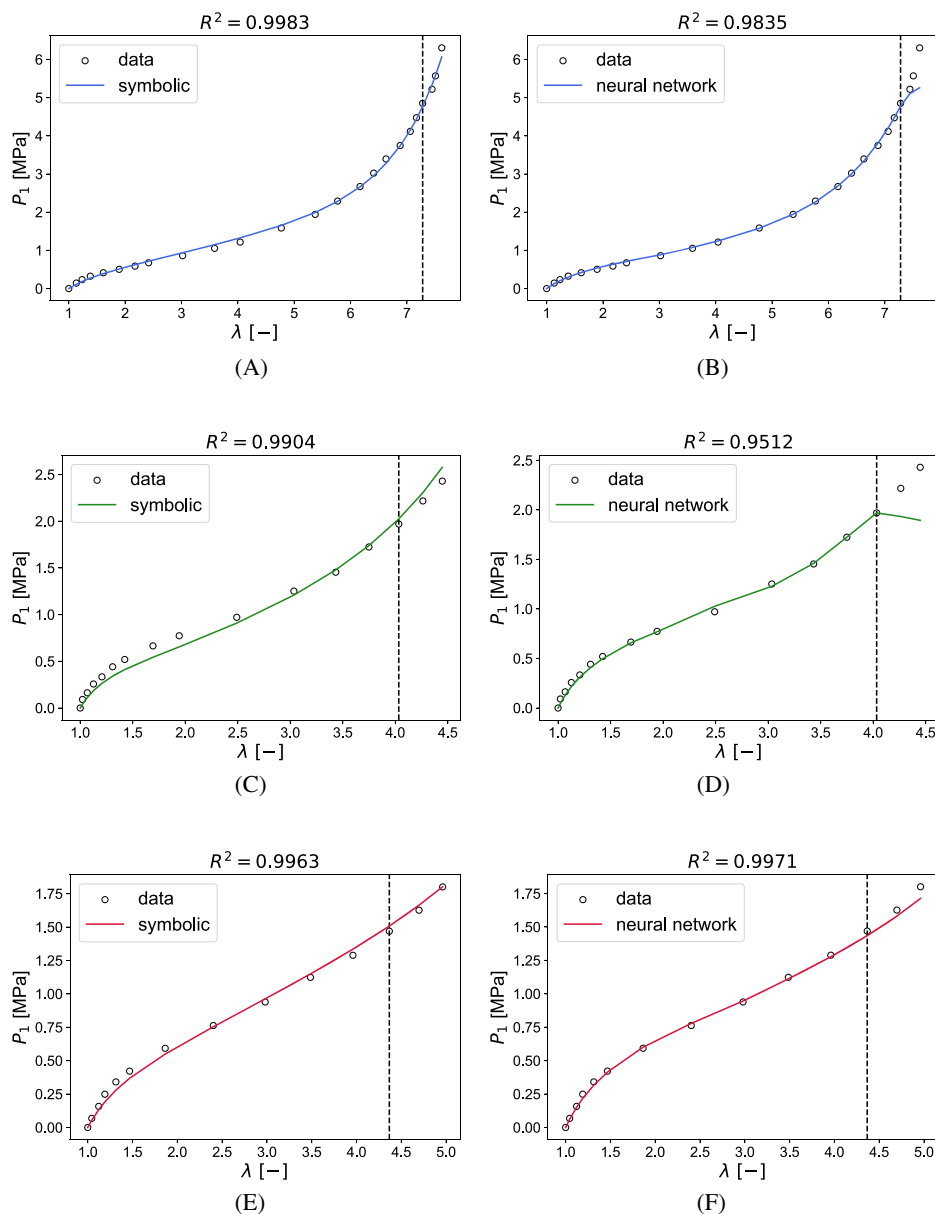


FIGURE 9 Predictions of the symbolic equation discovered from the PNAM and a vanilla ANN for Treloar's data. The dashed vertical line shows the distinction between training and validation data, where validation data are to the right of the line. The R^2 -score indicates the accuracy measure of the entire data (training and validation) for each experimental loading condition. Training is conducted on all three loading conditions at the same time. (A) UE symbolic. (B) UE vanilla ANN. (C) EBE symbolic. (D) EBE vanilla ANN. (E) PS symbolic. (F) PS vanilla ANN.

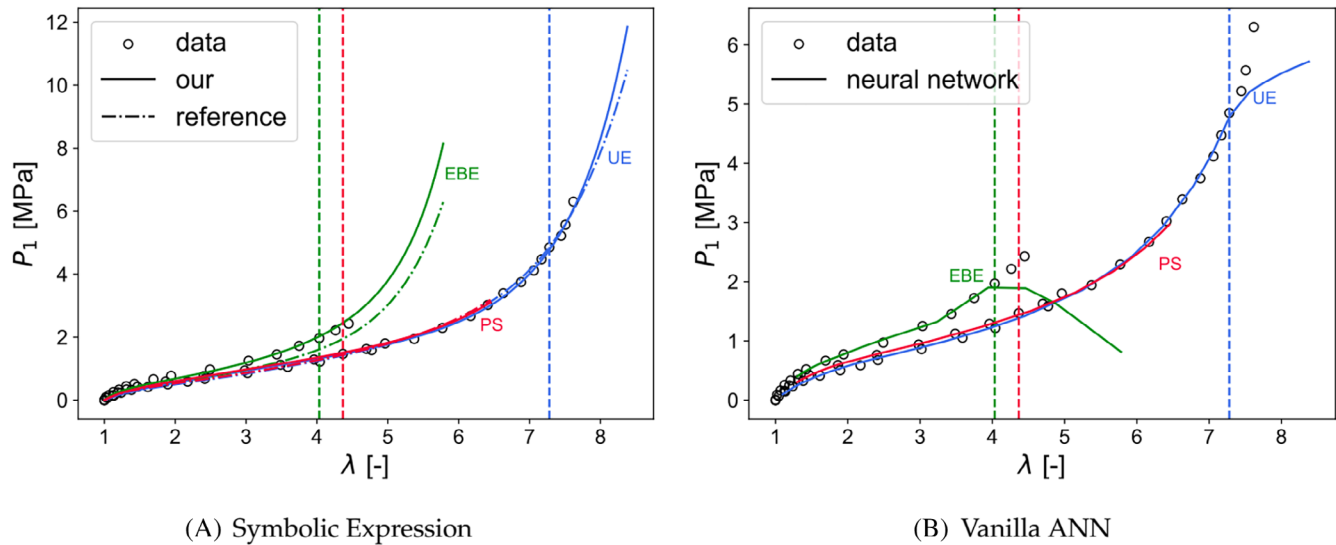


FIGURE 10 Extrapolation capabilities of (A) symbolic equations, (B) the vanilla ANN for Treloar's data. In (A), we compare results obtained via Equation (55) and the reference Equation (56). The dashed vertical lines indicate the distinction between training and validation data, where validation data are to the right of the lines. Our discovered equation extrapolates well and shows slightly better accuracy than the reference's equation. Predictions of the vanilla artificial neural network are not accurate beyond the training data regimes.

However, both equations exhibit reasonable behavior. On the other hand, the response of the vanilla ANN is notably unacceptable, especially under the EBE loading condition.

5.2 | Particle-reinforced composite

In this example, we synthesize data through numerical experimentation on a fabricated particle-reinforced composite. This demonstration highlights the method's efficacy in handling more complex material behaviors. A cubic sample of the composite material with a length of 1 cm is generated by random placements of hard particles in a soft matrix as shown in Figure 11. Neo-Hookean¹⁰ and the Exp-Log model⁹⁸ are assumed for the particles and the matrix, respectively,

$$\psi_{\text{NH}} = \frac{1}{2}\mu(I_1 - 3), \quad (58)$$

$$\psi_{\text{exp-log}} = A \left[\frac{1}{a} \exp(a(I_1 - 3)) + b(I_1 - 1)(1 - \log(I_1 - 2)) - \frac{1}{a} - b \right], \quad (59)$$

where material parameters $\mu = 0.5$ MPa, $A = 0.195$ MPa, $a = 0.018$, and $b = 0.33$; in the limit of small deformation A is analogous to the shear modulus μ in the Neo-Hookean model. The uniaxial behavior of these constituents are shown in Figure 11B.

We subject the composite to a uniaxial extension loading condition using a displacement control FEM simulator. During this process, we collect 64 data points with a displacement increment of 0.8 mm by recording the average forces calculated over the sample surface where the loads are applied; forces are then used to compute stresses. The generated dataset for training the models is depicted in Figure 12.

For training purposes, we utilize 75% of the data, represented by green circle points in Figure 12. The remaining portion of the data, which entirely lies in the extrapolation regime, is used for validation. Our method discovers the following equation,

$$\psi = c_{11}I_1 + c_{12}I_1^4 + \exp(c_{13}I_1) + c_{21}I_2 + c_{22}I_2^2 \ln(I_2) + c_0, \quad (60)$$

where $c_{11} = 0.06948$, $c_{12} = 1.82532 \times 10^{-6}$, $c_{13} = -0.05967$, $c_{21} = 0.91519$, $c_{22} = 0.0069682$, and $c_0 = 15.87993107$.

R^2 -scores of this symbolic equation and the PNAM from which it is discovered are almost the same, with the PNAM

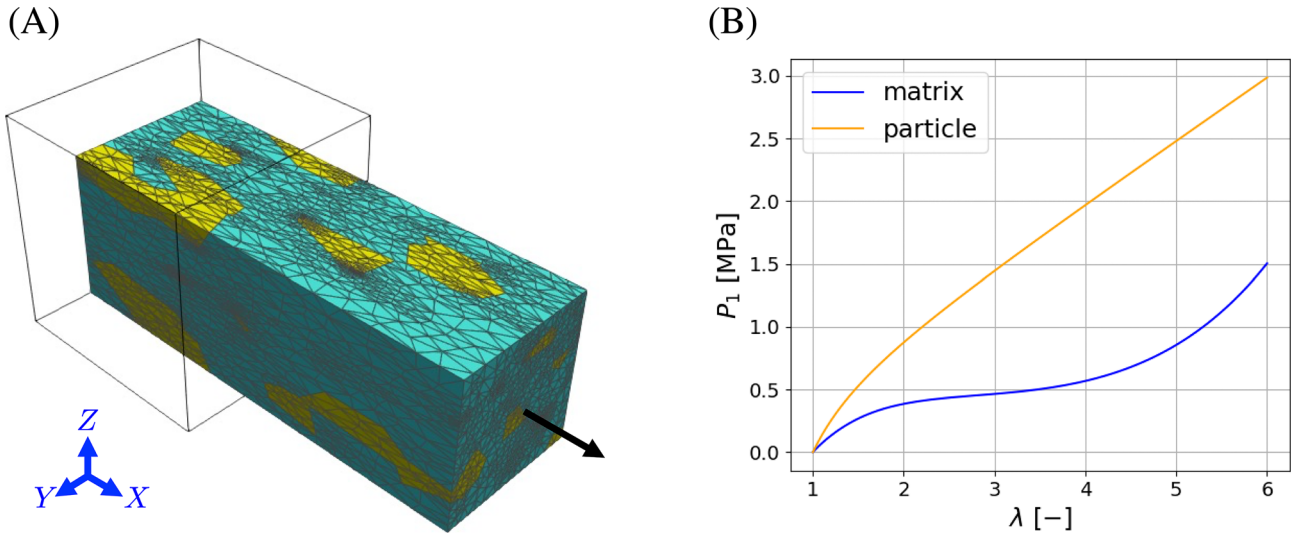


FIGURE 11 (A) Particle-reinforced composite at 100% longitudinal strain, and (B) uniaxial behavior of each constituent. In (A), random particles with a total volume fraction of 0.22 are shown in yellow.

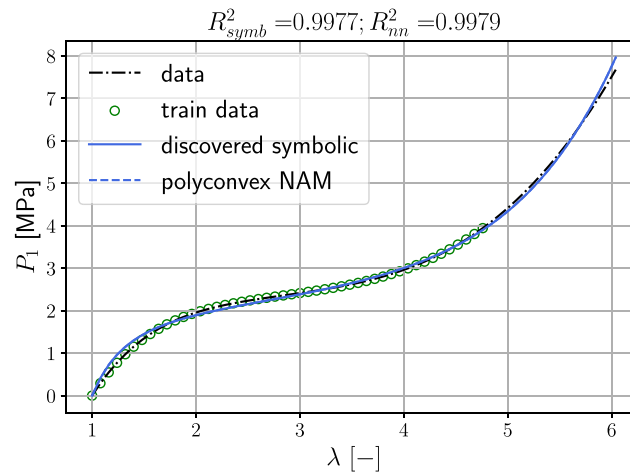


FIGURE 12 Comparison between the discovered symbolic energy functional and the learned PNAM for particle-reinforced composite data.

demonstrating slightly better performance, as shown at the top of Figure 12. Additionally, the comparison between the symbolic representation of each of the learned ICNN models $\psi_i(I_i)$ is presented in Figures 13 and 14. The results indicate good agreement between the neural networks and symbolic equations for both the function values and their gradients.

Remark 5. In this example, the inclusions are randomly distributed, without a particular geometric pattern or shape, and their volume fraction is relatively small (less than 30%). Therefore, at the RVE level and under these circumstances, the anisotropic behavior is not significant, allowing us to approximate it with an isotropic model.⁹⁹ A similar approach to the random generation of inhomogeneities is utilized in other research for the data generation of isotropic models.¹⁰⁰

5.2.1 | Remarks on model execution time

Another advantage of obtaining symbolic equations is their computational efficiency when compared to their neural network-based counterparts. In Figure 15, we compare the execution times for inference between the PNAM and SR. As

$$\psi_1(I_1) = 1.82 \times 10^{-6} I_1^4 + 0.069 I_1 - 0.594 + \exp(-0.06 I_1)$$

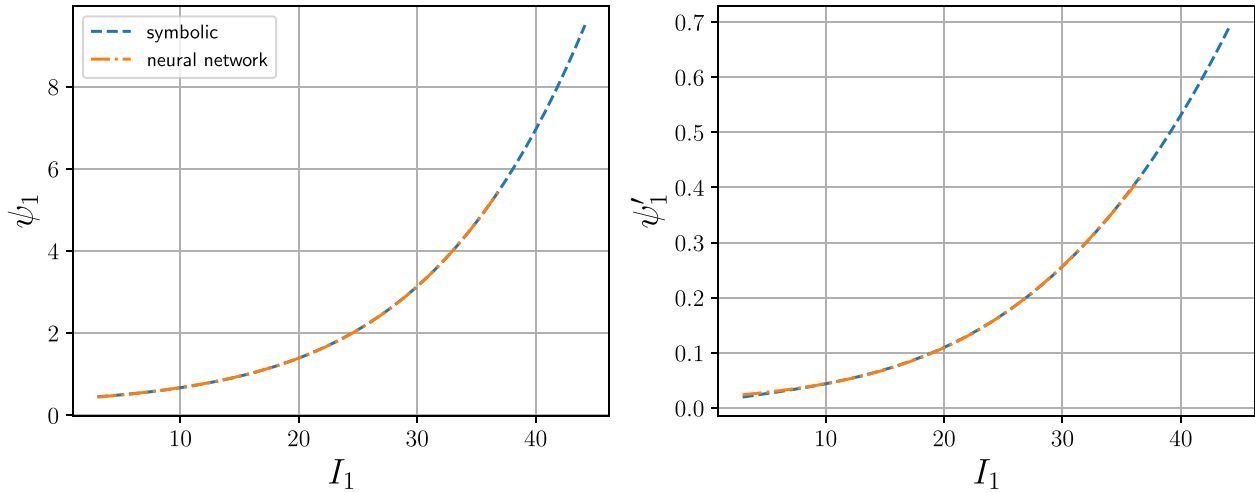


FIGURE 13 Discovered symbolic function from the trained neural network representation. The discovered function and its gradient are shown in left and right plots, respectively. Only data in the training range (orange curves) are used for SR.

$$\psi_2(I_2) = I_2(0.007 I_2 \ln(I_2) + 0.915) + 16.474$$

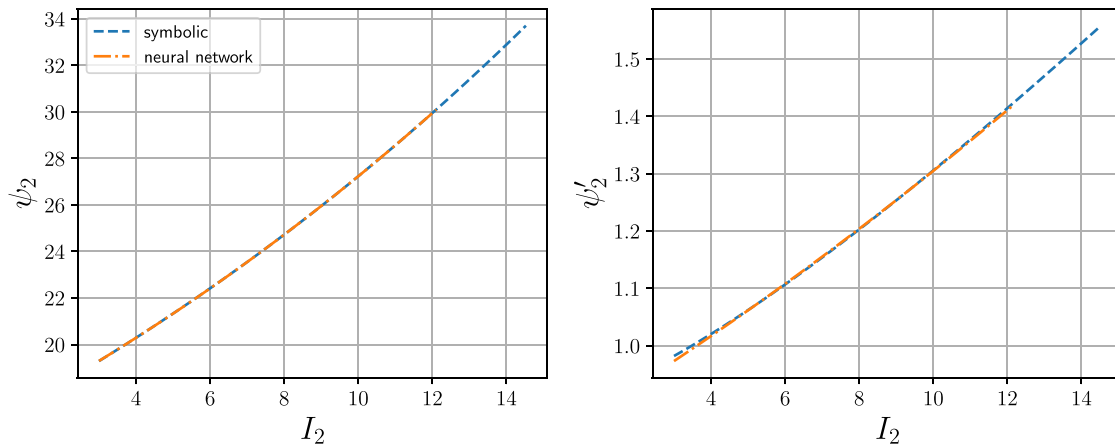


FIGURE 14 Discovered symbolic function from the trained neural network representation. The discovered function and its gradient are shown in left and right plots, respectively. Only data in the training range (orange curves) are used for SR.

anticipated, the runtime for symbolic calculations is shorter than that of neural network forward operations, primarily due to the reduced number of floating-point operations in the symbolic equation. While automatic differentiation provides implementation flexibility, it introduces additional overhead due to the construction of the computational graph, which can be avoided if symbolic operations are effectively managed manually.

6 | DISCUSSION AND ANALYSIS

Thanks to the symbolic forms, classical analysis can be applied to investigate the entire range of the deformation gradient, determining when polyconvexity and growth conditions are violated. In our specific formulation, we can assess polyconvexity by examining the convexity and non-decreasing conditions of shape functions $\psi_i(I_i)$. To verify coercivity, we analyze the energy functional's growth behavior as the invariants approach infinity, that is, $I_1, I_2 \rightarrow \infty$. It is important to note that our goal is not to find the tightest bounds on the coefficients but rather to ensure that the derived equations satisfy the required conditions.

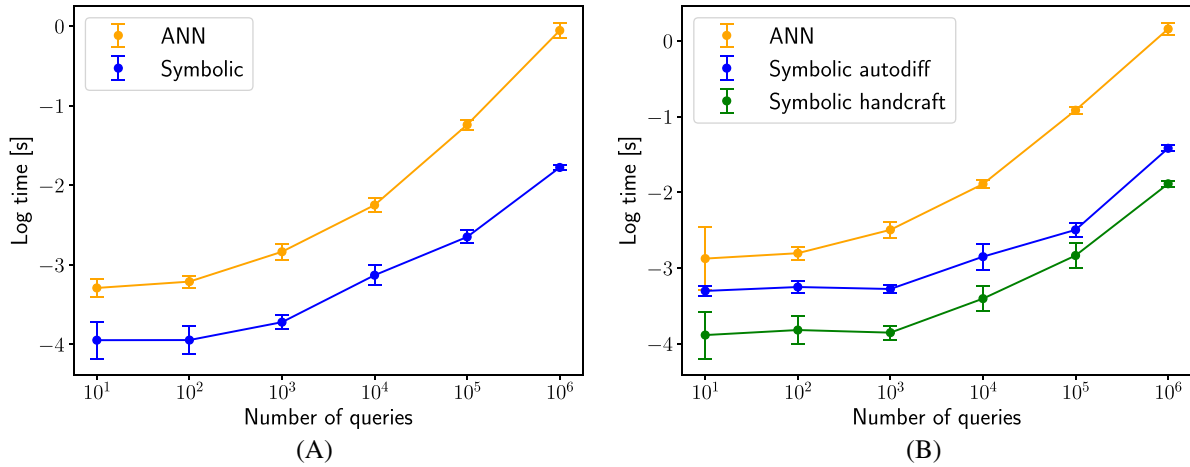


FIGURE 15 Comparing execution times between an artificial neural network and a symbolic strain energy functional. Shown are mean and standard deviation trends for (A) the strain energy and (B) its gradient with respect to the invariants. The gradient of a symbolic equation can be hand-crafted or obtained through automatic differentiation, as showcased in (B). The horizontal axis represents the number of query points in each experiment. Each experiment is replicated for 20 times with distinct random seeds.

6.1 | Treloar machine learning model

The convexity $\psi_2(I_2) = c_{21}I_2$ is evident. The non-decreasing condition requires $c_{21} \geq 0$. For $\psi_1(I_1) = c_{11}I_1 + c_{12} \exp(c_{13}I_2)$ to be convex, it is required that $c_{12} \geq 0$, which can be confirmed by checking the positivity of its second-order derivative. This function will be non-decreasing if its first derivative is nonnegative, which can be easily determined when $c_{11} \geq 0$. By employing the Taylor series expansion for the exponential function and considering any positive integer p , we obtain the following inequality,

$$\psi(I_1, I_2) \geq \frac{c_{12}^p}{p!} I_1^p + c_{21}I_2. \quad (61)$$

To ensure asymptotic satisfaction of Equation (27), we can set $\alpha = \min(c_{12}^p/p!, c_{21})$ with $c_{21} > 0$ and $q = 1$, where p is chosen to be a large value approaching infinity. However, it is essential to note that even for a very large p , the condition $q \geq p/(p-1)$ is marginally violated, and the fulfillment of conditions $c_{21} > 0$ and $c_{12} > 0$ may not guarantee coercivity based on the presented justification.

6.2 | Particle-reinforced composite machine learning model

The convexity of $\psi_1(I_1) = c_{11}I_1 + c_{12}I_1^4 + \exp(c_{13}I_1)$ is ensured if its second-order gradient is nonnegative, that is, $12c_{12}I_1^2 + c_{13}^2 \exp(c_{13}I_1) \geq 0$. Thus, it requires only that $c_{12} \geq 0$ to satisfy convexity. To evaluate the non-decreasing condition, we consider $\psi_1'(I_1) = c_{11} + 4c_{12}I_1^3 + c_{13} \exp(c_{13}I_1) \geq 0$. First, note that $I_1 = \lambda_1^2 + \lambda_2^2 + \lambda_3^2 > 0$. For very large values of $I_1 \rightarrow +\infty$, the exponential term will dominate, implying $c_{13} \geq 0$. Under the incompressibility condition ($\lambda_3 = 1/(\lambda_1\lambda_2)$), it can be shown that $I_1 \geq 2$ serves as a lower bound. Therefore, it suffices to have $c_{11} \geq -(32c_{12} + c_{13} \exp(2c_{13}))$.

Similarly, for $\psi_2(I_2) = c_{21}I_2 + c_{22}I_2^2 \ln(I_2)$, the convexity requirement is $c_{22}(2 \ln(I_2) + 3) \geq 0$, which is satisfied if $c_{22} \geq 0$. For the non-decreasing condition, considering $I_2 = \lambda_1^2\lambda_2^2 + \lambda_1^2\lambda_3^2 + \lambda_2^2\lambda_3^2 \geq 0$, one can determine that the sufficient condition is $c_{21} \geq 0$.

By utilizing the Taylor series expansion for the exponential function, we have,

$$\psi_1(I_1) \geq c_{11}I_1 + c_{12}I_1^4 + \frac{c_{13}^4}{4!} I_1^4 \geq \left(c_{12} + \frac{c_{13}^4}{4!} \right) I_1^4. \quad (62)$$

Furthermore, leveraging the fact that $\ln(x) \geq \frac{x-1}{x}$ for positive x (as $x \ln(x) - x + 1$ has a positive slope), we can write,

$$\psi_2(I_2) \geq c_{22}I_2^2 + (c_{21} - c_{22})I_2 \geq \hat{c}_{22}I_2^2, \quad (63)$$

where $0 \leq \hat{c}_{22} \leq c_{22}$ accounts for the possibility of $c_{21} < c_{22}$.

Based on these bounds, if we choose $\alpha = \min(c_{12} + c_{13}^4/4!, \hat{c}_{22})$ and set $p = 4 \geq 2$ and $q = 2 \geq p/(p-1)$, the inequality in Equation (27) is satisfied, provided that $\alpha > 0$. Consequently, the sufficient conditions to ensure coercivity are $c_{22} > 0$ and $c_{12} > 0$.

Notice that, within the training data regime of symbolic regression, and even slightly beyond that, the discovered symbolic function is non-decreasing; see the derivative function in the right Figure 13, which is positive. However, based on the above discussion, the equation found through symbolic regression violates the non-decreasing condition for $I_1 \rightarrow +\infty$ since $c_{13} < 0$. The learned neural network function $\psi_1(I_1)$ does not suffer from this issue because it satisfies this condition by construction. Since such a constraint has not been encoded during the search in symbolic regression, we may violate this condition, particularly beyond the training regime of symbolic regression. To mitigate such issues at the symbolic regression discovery step, the simplest way would be to generate data from the learned neural network with a wider coverage of I_1 and denser data sampling to train the symbolic regression. A more rigorous approach would be to limit the symbolic regression to search only in the space of all convex and non-decreasing functions. However, incorporating this second restriction is a challenging task for symbolic regression as a discrete search algorithm and requires further research in this area.

7 | CONCLUSION

In this work, we present an efficient machine learning framework capable of directly inferring the mathematical expressions of multivariate elastic stored energy functionals from data that fulfill physics principles (e.g., polyconvexity, material frame indifference) without sacrificing expressivity. A salient feature of this proposed model is the introduction of input convex neural networks for constructing the feature space. This treatment enables us to retain the convexity of the feature mapping that ensures polyconvexity of the learned neural network hyperelastic models. By expressing polyconvex hyperelastic models in the feature space, we can then apply SR on each of the feature space basis. This setting enables us to overcome the challenging curse-of-high-dimensionality issue well known in the SR literature while deducing interpretable mathematical models that can be easily comprehended, analyzed, and implemented. The compactness of the mathematical models also leads to computationally efficient constitute updates that require less arithmetic operations than those of deep neural networks and hence are well-suited for large-scale physics simulations where repeated executions of the models must be carried out in a large number of integration points.

ACKNOWLEDGMENTS

The authors are supported by the National Science Foundation under Grant contract CMMI-1846875, and the Dynamic Materials and Interactions Program from the Air Force Office of Scientific Research under Grant contracts FA9550-21-1-0391 and FA9550-21-1-0027, and the MURI Grant No. FA9550-19-1-0318. These supports are gratefully acknowledged. The authors would like to thank Nhon N. Phan for their valuable feedback and suggestions, which have improved the readability of the manuscript. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing official policies, either expressed or implied, of the sponsors, including the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

DATA AVAILABILITY STATEMENT

Data supporting the findings of this study will be available at Mendeley Data upon the acceptance of this manuscript. Codes used to generate the model is available on GitHub (<https://github.com/bbhm-90/IsoHyperSymb>).

ORCID

WaiChing Sun  <https://orcid.org/0000-0002-3078-5086>

REFERENCES

- Kirchdoerfer T, Ortiz M. Data-driven computational mechanics. *Comput Methods Appl Mech Eng*. 2016;304:81-101.
- Boyce MC, Arruda EM. Constitutive models of rubber elasticity: a review. *Rubber Chem Technol*. 2000;73(3):504-523.
- Yves Le Guennec J-P, Brunet F-ZD, Chau M, Tourbier Y. A parametric and non-intrusive reduced order model of car crash simulation. *Comput Methods Appl Mech Eng*. 2018;338:186-207.
- Barbani D, Baldanzini N, Pierini M. Development and validation of an FE model for motorcycle-car crash test simulations. *Int J Crashworthiness*. 2014;19(3):244-263.
- Grinspun E, Hirani AN, Desbrun M, Schröder P. Discrete shells. *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Citeseer; 2003:62-67.
- Dafalias YF. Modelling cyclic plasticity; simple versus sophistication. *Mechanics of Engineering Materials*. John Wiley & Sons; 1984:153-178.
- Truesdell C. Hypo-elasticity. *J Ration Mech Anal*. 1955;4:83-1020.
- Green AE. Hypo-elasticity and plasticity. II. *J Ration Mech Anal*. 1956;5(5):725-734.
- Freed AD, Einstein DR, Sacks MS. Hypoelastic soft tissues: part II: in-plane biaxial experiments. *Acta Mech*. 2010;213(1-2):205-222.
- Ogden RW. *Non-linear Elastic Deformations*. Courier Corporation; 1997.
- Holzappel GA, Gasser TC, Ogden RW. A new constitutive framework for arterial wall mechanics and a comparative study of material models. *J Elast Phys Sci Solid*. 2000;61:1-48.
- Mihai LA, Budday S, Holzappel GA, Kuhl E, Goriely A. A family of hyperelastic models for human brain tissue. *J Mech Phys Solids*. 2017;106:60-79.
- Mooney M. A theory of large elastic deformation. *J Appl Phys*. 1940;11(9):582-592.
- Rivlin RS, Saunders D. Large elastic deformations of isotropic materials VII. Experiments on the deformation of rubber. *Philos Trans R Soc Lond Ser A Math Phys Sci*. 1951;243(865):251-288.
- Vlassis NN, Ma R, Sun WC. Geometric deep learning for computational mechanics part i: anisotropic hyperelasticity. *Comput Methods Appl Mech Eng*. 2020;371:113299.
- Liu M, Liang L, Sun W. A generic physics-informed neural network-based constitutive model for soft biological tissues. *Comput Methods Appl Mech Eng*. 2020;372:113402.
- Thakolkaran P, Joshi A, Zheng Y, Flaschel M, De Lorenzis L, Kumar S. NN-EUCLID: deep-learning hyperelasticity without stress data. *J Mech Phys Solids*. 2022;169:105076.
- Klein DK, Fernández M, Martin RJ, Neff P, Weeger O. Polyconvex anisotropic hyperelasticity with neural networks. *J Mech Phys Solids*. 2022;159:104703.
- Tac V, Costabal FS, Tepole AB. Data-driven tissue mechanics with polyconvex neural ordinary differential equations. *Comput Methods Appl Mech Eng*. 2022;398:115248.
- Frankel AL, Jones RE, Swiler LP. Tensor basis gaussian process models of hyperelastic materials. *J Mach Learn Model Comput*. 2020;1(1):1-17.
- Fuhg JN, Bouklas N. On physics-informed data-driven isotropic and anisotropic constitutive models through probabilistic machine learning and space-filling sampling. *Comput Methods Appl Mech Eng*. 2022;394:114915.
- Abdusalamov R, Hillgärtner M, Itskov M. Automatic generation of interpretable hyperelastic material models by symbolic regression. *Int J Numer Methods Eng*. 2023;124(9):2093-2104.
- Hornik K, Stinchcombe M, White H. Multilayer feedforward networks are universal approximators. *Neural Netw*. 1989;2(5):359-366.
- Hsu D, Sanford CH, Servedio R, Vlatakis-Gkaragkounis EV. On the approximation power of two-layer networks of random ReLUs. *Conference on Learning Theory*. PMLR; 2021:2423-2461.
- Yuelin Shen K, Chandrashekhara WFB, Oliver LR. Neural network based constitutive model for rubber material. *Rubber Chem Technol*. 2004;77(2):257-277.
- Liang G, Chandrashekhara K. Neural network based constitutive model for elastomeric foams. *Eng Struct*. 2008;30(7):2002-2011.
- Le BA, Yvonnet J, He Q-C. Computational homogenization of nonlinear elastic materials using neural networks. *Int J Numer Methods Eng*. 2015;104(12):1061-1084.
- Vlassis NN, Sun WC. Sobolev training of thermodynamic-informed neural networks for interpretable elasto-plasticity models with level set hardening. *Comput Methods Appl Mech Eng*. 2021;377:113695.
- Vlassis NN, Zhao P, Ma R, Sewell T, Sun WC. Molecular dynamics inferred transfer learning models for finite-strain hyperelasticity of monoclinic crystals: Sobolev training and validations against physical constraints. *Int J Numer Methods Eng*. 2022;123(17):3922-3949.
- Fernández M, Fritzen F, Weeger O. Material modeling for parametric, anisotropic finite strain hyperelasticity based on machine learning with application in optimization of metamaterials. *Int J Numer Methods Eng*. 2022;123(2):577-609.
- Agarwal R, Melnick L, Frosst N, et al. Neural additive models: interpretable machine learning with neural nets. 35th h Conference on Advances in Neural Information Processing Systems. 2021:4699-4711.
- Bahmani B, Suh HS, Sun WC. Discovering interpretable elastoplasticity models via the neural polynomial method enabled symbolic regressions. arXiv preprint arXiv:2307.13149, 2023.
- Hartmann S, Neff P. Polyconvexity of generalized polynomial-type hyperelastic strain energy functions for near-incompressibility. *Int J Solids Struct*. 2003;40(11):2767-2791.

34. Teichert GH, Natarajan AR, Van der Ven A, Garikipati K. Machine learning materials physics: Integrable deep neural networks enable scale bridging by learning free energy functions. *Comput Methods Appl Mech Eng*. 2019;353:201-216.
35. Masi F, Stefanou I, Vannucci P, Maffi-Berthier V. Thermodynamics-based artificial neural networks for constitutive modeling. *J Mech Phys Solids*. 2021;147:104277.
36. Raissi M, Perdikaris P, Karniadakis GE. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J Comput Phys*. 2019;378:686-707.
37. Lagaris IE, Likas A, Fotiadis DI. Artificial neural networks for solving ordinary and partial differential equations. *IEEE Trans Neural Netw*. 1998;9(5):987-1000.
38. Bronstein MM, Bruna J, Cohen T, Velicković P. Geometric deep learning: grids, groups, graphs, geodesics, and gauges. arXiv preprint arXiv:2104.13478, 2021.
39. Tianhe Y, Kumar S, Gupta A, Levine S, Hausman K, Finn C. Gradient surgery for multi-task learning. *Proceedings of the 34th International Conference on Neural Information Processing Systems*. Curran Associates Inc.; 2020:5824-5836.
40. Bahmani B, Sun WC. Training multi-objective/multi-task collocation physics-informed neural network with student/teachers transfer learnings. arXiv preprint arXiv:2107.11496, 2021.
41. Jin C, Kakade SM, Netrapalli P. Provable efficient online matrix completion via non-convex stochastic gradient descent. 30th Conference on Neural Information Processing Systems; 2016.
42. Heider Y, Wang K, Sun WC. So (3)-invariance of informed-graph-based deep neural network for anisotropic elastoplastic materials. *Comput Methods Appl Mech Eng*. 2020;363:112875.
43. Kailai X, Huang DZ, Darve E. Learning constitutive relations using symmetric positive definite neural networks. *J Comput Phys*. 2021;428:110072.
44. Linka K, Hillgärtner M, Abdolazizi KP, Aydin RC, Itskov M, Cyron CJ. Constitutive artificial neural networks: a fast and general approach to predictive data-driven constitutive modeling by deep learning. *J Comput Phys*. 2021;429:110010.
45. As'ad F, Avery P, Farhat C. A mechanics-informed artificial neural network approach in data-driven constitutive modeling. *Int J Numer Methods Eng*. 2022;123(12):2738-2759.
46. Chen P, Guillemot J. Polyconvex neural networks for hyperelastic constitutive models: a rectification approach. *Mech Res Commun*. 2022;125:103993.
47. Cai C, Vlassis N, Magee L, et al. Equivariant geometric learning for digital rock physics: estimating formation factor and effective permeability tensors from morse graph. *Int J Multiscale Comput Eng*. 2023;21(5):1-24.
48. Schröder J, Neff P. Invariant formulation of hyperelastic transverse isotropy based on polyconvex free energy functions. *Int J Solids Struct*. 2003;40(2):401-445.
49. Schröder J, Neff P, Balzani D. A variational approach for materially stable anisotropic hyperelasticity. *Int J Solids Struct*. 2005;42(15):4352-4371.
50. Schröder J, Neff P. *Poly-, Quasi-and Rank-One Convexity in Applied Mechanics*. Vol 516. Springer Science & Business Media; 2010.
51. Ball JM. Does rank-one convexity imply quasiconvexity? *Metastability and Incompletely Posed Problems*. Springer; 1987:17-32.
52. Shirani M, Steigmann DJ. Convexity and quasiconvexity in a cosserat model for fiber-reinforced elastic solids. *J Elast*. 2022;154:555-567.
53. Dayhoff JE, DeLeo JM. Artificial neural networks: opening the black box. *Cancer*. 2001;91(S8):1615-1635.
54. Seong Joon O, Schiele B, Fritz M. Towards reverse-engineering black-box neural networks. In: Samek W, Montavon G, Vedaldi A, Hansen L, Müller KR, eds. *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Springer; 2019:121-144.
55. Versino D, Tonda A, Bronkhorst CA. Data driven modeling of plastic deformation. *Comput Methods Appl Mech Eng*. 2017;318:981-1004.
56. Bomarito GF, Townsend TS, Stewart KM, Esham KV, Emery JM, Hochhalter JD. Development of interpretable, data-driven plasticity models with symbolic regression. *Comput Struct*. 2021;252:106557.
57. Park H, Cho M. Multiscale constitutive model using data-driven yield function. *Compos Part B Eng*. 2021;216:108831.
58. Brunton SL, Proctor JL, Kutz JN. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proc Natl Acad Sci USA*. 2016;113(15):3932-3937.
59. Flaschel M, Kumar S, De Lorenzis L. Unsupervised discovery of interpretable hyperelastic constitutive laws. *Comput Methods Appl Mech Eng*. 2021;381:113852.
60. Wang Z, Estrada JB, Arruda EM, Garikipati K. Inference of deformation mechanisms and constitutive response of soft material surrogates of biological tissue by full-field characterization and data-driven variational system identification. *J Mech Phys Solids*. 2021;153:104474.
61. Linka K, Kuhl E. A new family of constitutive artificial neural networks towards automated model discovery. *Comput Methods Appl Mech Eng*. 2023;403:115731.
62. Wang M, Chen C, Liu W. Establish algebraic data-driven constitutive models for elastic solids with a tensorial sparse symbolic regression method and a hybrid feature selection technique. *J Mech Phys Solids*. 2022;159:104742.
63. Nguyen LTK, Keip M-A. A data-driven approach to nonlinear elasticity. *Comput Struct*. 2018;194:97-115.
64. Platzer A, Leygue A, Stainier L, Ortiz M. Finite element solver for data-driven finite strain elasticity. *Comput Methods Appl Mech Eng*. 2021;379:113756.
65. Crespo J, Latorre M, Montáns FJ. WYPiWYG hyperelasticity for isotropic, compressible materials. *Comput Mech*. 2017;59:73-92.
66. Amores VJ, Benítez JM, Montáns FJ. Average-chain behavior of isotropic incompressible polymers obtained from macroscopic experimental data. A simple structure-based WYPiWYG model in Julia language. *Adv Eng Softw*. 2019;130:41-57.

67. Moreno S, Amores VJ, Benítez JM, Montáns FJ. Reverse-engineering and modeling the 3D passive and active responses of skeletal muscle using a data-driven, non-parametric, spline-based procedure. *J Mech Behav Biomed Mater*. 2020;110:103877.
68. Akbari R, Morovati V, Dargazany R. Reverse physically motivated frameworks for investigation of strain energy function in rubber-like elasticity. *Int J Mech Sci*. 2022;221:107110.
69. Ibanez R, Abisset-Chavanne E, Aguado JV, Gonzalez D, Cueto E, Chinesta F. A manifold learning approach to data-driven computational elasticity and inelasticity. *Arch Comput Methods Eng*. 2018;25:47-57.
70. He X, He Q, Chen J-S. Deep autoencoders for physics-constrained data-driven nonlinear materials modeling. *Comput Methods Appl Mech Eng*. 2021;385:114034.
71. González D, García-González A, Chinesta F, Cueto E. A data-driven learning method for constitutive modeling: application to vascular hyperelastic soft tissues. *Materials*. 2020;13(10):2319.
72. Bahmani B, Sun WC. Manifold embedding data-driven mechanics. *J Mech Phys Solids*. 2022;166:104927.
73. Bahmani B, Sun WC. Distance-preserving manifold denoising for data-driven mechanics. *Comput Methods Appl Mech Eng*. 2023;405:115857.
74. Truesdell C, Noll W, Truesdell C, Noll W. *The Non-linear Field Theories of Mechanics*. Springer; 2004.
75. Truesdell CA. *A First Course in Rational Continuum Mechanics VI*. Academic Press; 1992.
76. Gurtin ME. *An Introduction to Continuum Mechanics*. Academic Press; 1982.
77. Holzapfel GA. Nonlinear solid mechanics: a continuum approach for engineering science. *Meccanica*. 2002;37:489-490.
78. Hill R. On uniqueness and stability in the theory of finite elastic strain. *J Mech Phys Solids*. 1957;5(4):229-241.
79. Marsden JE, Hughes TJR. *Mathematical Foundations of Elasticity*. Courier Corporation; 1994.
80. Ball JM. Convexity conditions and existence theorems in nonlinear elasticity. *Arch Ration Mech Anal*. 1976;63:337-403.
81. Ball JM. Strict convexity, strong ellipticity, and regularity in the calculus of variations. *Mathematical Proceedings of the Cambridge Philosophical Society*. Vol 87. Cambridge University Press; 1980:501-513.
82. Kuhl E, Askes H, Steinmann P. An illustration of the equivalence of the loss of ellipticity conditions in spatial and material settings of hyperelasticity. *Eur J Mech A/Solids*. 2006;25(2):199-214.
83. Charles B, Jr M. Quasi-convexity and the lower semicontinuity of multiple integrals. *Pac J Math*. 1952;2:25-53.
84. Hoger A. On the determination of residual stress in an elastic body. *J Elast*. 1986;16(3):303-324.
85. Amos B, Xu L, Kolter JZ. Input convex neural networks. *International Conference on Machine Learning*. PMLR; 2017:146-155.
86. Kalina KA, Linden L, Brummund J, Kästner M. FE ANN: an efficient data-driven multiscale approach based on physics-constrained neural networks and automated data mining. *Comput Mech*. 2023;71(5):827-851.
87. Kingma DP, Ba J. Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.
88. Koza JR. Genetic programming as a means for programming computers by natural selection. *Stat Comput*. 1994;4:87-112.
89. Schmidt M, Lipson H. Distilling free-form natural laws from experimental data. *Science*. 2009;324(5923):81-85.
90. Wang Y, Wagner N, Rondinelli JM. Symbolic regression in materials science. *MRS Commun*. 2019;9(3):793-805.
91. Petersen BK, Landajuela M, Mundhenk TN, Santiago CP, Kim SK, Kim JT. Deep symbolic regression: recovering mathematical expressions from data via risk-seeking policy gradients. arXiv preprint arXiv:1912.04871, 2019.
92. Landajuela M, Petersen BK, Kim S, et al. Discovering symbolic policies with deep reinforcement learning. *International Conference on Machine Learning*. PMLR; 2021:5979-5989.
93. Cranmer M. Interpretable machine learning for science with PySR and SymbolicRegression.jl. arXiv preprint arXiv:2305.01582, 2023.
94. Kim S, Lu PY, Mukherjee S, et al. Integration of neural network-based symbolic regression in deep learning for scientific discovery. *IEEE Trans Neural Netw Learn Syst*. 2020;32(9):4166-4177.
95. Icke I, Bongard JC. Improving genetic programming based symbolic regression using deterministic machine learning. *2013 IEEE Congress on Evolutionary Computation*. IEEE; 2013:1763-1770.
96. Paszke A, Gross S, Massa F, et al. PyTorch: an imperative style, high-performance deep learning library. *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Curran Associates Inc.; 2019:8026-8037.
97. Treloar LRG. Stress-strain data for vulcanized rubber under various types of deformation. *Rubber Chem Technol*. 1944;17(4):813-825.
98. Khajehsaeid H, Arghavani J, Naghdabadi R. A hyperelastic constitutive model for rubber-like materials. *Eur J Mech A/Solids*. 2013;38:144-151.
99. Cho H, Lee J, Moon J, et al. Large strain micromechanics of thermoplastic elastomers with random microstructures. arXiv preprint arXiv:2308.14607, 2023.
100. Kalina KA, Linden L, Brummund J, Metsch P, Kästner M. Automated constitutive modeling of isotropic hyperelasticity based on artificial neural networks. *Comput Mech*. 2022;69:213-232.
101. Clevert D-A, Unterthiner T, Hochreiter S. Fast and accurate deep network learning by exponential linear units (ELUs). arXiv preprint arXiv:1511.07289, 2015.
102. Goodfellow I, Bengio Y, Courville A. *Deep Learning*. MIT Press; 2016.
103. Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: machine learning in python. *J Mach Learn Res*. 2011;12:2825-2830.
104. Weiss JA, Maker BN, Govindjee S. Finite element implementation of incompressible, transversely isotropic hyperelasticity. *Comput Methods Appl Mech Eng*. 1996;135(1-2):107-128.
105. Schröder J, Viebahn N, Wriggers P, Auricchio F, Steeger K. On the stability analysis of hyperelastic boundary value problems using three-and two-field mixed finite element formulations. *Comput Mech*. 2017;60:479-492.

106. Belytschko T, Liu WK, Moran B, Elkhodary K. *Nonlinear Finite Elements for Continua and Structures*. John Wiley & Sons; 2014.
107. Alnæs M, Blechta J, Hake J, et al. The FEniCS project version 1.5. *Arch Numer Softw*. 2015;3(100):9-23.

How to cite this article: Bahmani B, Sun WC. Physics-constrained symbolic model discovery for polyconvex incompressible hyperelastic materials. *Int J Numer Methods Eng*. 2024;125(15):e7473. doi: 10.1002/nme.7473

APPENDIX A. BASIC CONVEXITY RELATIONSHIPS

Lemma 5. A tensor-valued function $W(\mathbf{F})$ is convex with respect to $\mathbf{F}$ if its Hessian has the following property such that for any $\mathbf{F}$ and $\delta\mathbf{F}$,

$$\delta\mathbf{F} : \frac{\partial^2 W(\mathbf{F})}{\partial \mathbf{F} \partial \mathbf{F}} : \delta\mathbf{F} \geq 0. \quad (\text{A1})$$

Proof. See Lemma B.3 in Schröder and Neff.⁴⁸ ■

Lemma 6. The mapping $W(\mathbf{F}) : \mathbf{F} \rightarrow \text{tr}(\mathbf{F}^T \mathbf{F})$ is convex. As a result, the first invariant $I_1(\mathbf{C})$ is a convex function of $\mathbf{F}$.

Proof. Hessian of the function with respect to $\mathbf{F}$ is as follows:

$$\frac{\partial^2 W}{\partial F_{ij} \partial F_{kl}} = 2\delta_{ki}\delta_{lj}. \quad (\text{A2})$$

By setting $\mathbf{A} = \delta\mathbf{F}$, the convexity criterion becomes as follows,

$$\mathbf{A} : \frac{\partial^2 W(\mathbf{F})}{\partial \mathbf{F} \partial \mathbf{F}} : \mathbf{A} = 2A_{ij}\delta_{ki}\delta_{lj}L_{kl} = 2A_{iL}A_{iL} \geq 0. \quad (\text{A3})$$

Lemma 7. The second invariant $I_2(\mathbf{C})$ is a convex function with respect to $\text{adj}(\mathbf{F})$.

Proof. One can obtain the following relationships for the second invariant,

$$I_2 = \text{tr}(\text{adj}(\mathbf{F}^T \mathbf{F})), \quad (\text{A4})$$

$$= \text{tr}(\text{adj}(\mathbf{F})\text{adj}(\mathbf{F}^T)) \quad \text{from Lemma A.6.3 in Reference 48,} \quad (\text{A5})$$

$$= \text{tr}(\text{adj}(\mathbf{F})\text{adj}(\mathbf{F})^T) \quad \text{from Lemma A.6.4 in Reference 48,} \quad (\text{A6})$$

$$= \text{tr}(\text{adj}(\mathbf{F})^T \text{adj}(\mathbf{F})). \quad (\text{A7})$$

Utilizing the last equation, in conjunction with Lemma 6, we deduce that the second invariant is a convex function with respect to $\text{adj}(\mathbf{F})$. ■

Lemma 8. The third invariant $I_3(\mathbf{C})$ is a convex function with respect to $\det(\mathbf{F})$.

Proof. By utilizing basic properties of matrix operations, we can derive the following relationships:

$$I_3 = \det(\mathbf{C}) = \det(\mathbf{F}^T \mathbf{F}) = \det(\mathbf{F}^T) \det(\mathbf{F}) = \det(\mathbf{F})^2. \quad (\text{A8})$$

Therefore, the third invariant is a quadratic function of $\det(\mathbf{F})$, making it convex in relation to this determinant. ■

APPENDIX B. MODEL SETUP AND HYPERPARAMETERS

In the PNAM, for each shape function, both examples employ ICNN models using a single hidden layer consisting of 50 neurons. The activation functions for the first and second layers are softplus and softplus2, respectively. To ensure non-negativity of the weights, ReLU activation is utilized. The optimization process utilizes the ADAM optimizer with an initial learning rate of 0.001. Strain invariants are normalized using the MinMaxScaler method in both examples. Furthermore, in the first example, the stress data undergoes a pre-processing step and is scaled by a factor of 0.05.

B.1 Treloar's data

In the vanilla ANN model, one hidden layer with 50 neurons is utilized with an ELU activation layer¹⁰¹ (see the following remark). Its training parameters remain consistent with those of the PNAM.

In the SR configuration of $\psi_1(I_1)$, we confine the binary operators to addition and multiplication, while the unary operator is limited to the exponential function (exp). The time budget is capped at 6 minutes or a maximum of 100 iterations, whichever comes first. The expression trees' maximum depth and size are both set to 30. As a result of the observed simplicity in the learned ICNN function for $\psi_2(I_2)$, the maximum size and depth of the expression trees is reduced to 10, and the exponential function is excluded as a possible operator for applying SR to find $\psi_2(I_2)$.

Remark 6. The ELU activation function is defined as follows,

$$\text{ELU}(x) = xH(x) + (\exp(x) - 1)(1 - H(x)), \quad (\text{B1})$$

where $H(x)$ is the Heaviside function.

B.2 Composite data

In the SR configuration of $\psi_1(I_1)$, the binary operators are restricted to addition and multiplication, and the unary operators are limited to the exponential (exp) and logarithm (ln) functions. The time budget is set to 6 minutes or a maximum of 100 iterations, whichever is reached first. The expression trees are constrained to a maximum depth of 10 and a size of 50. Classical energy functionals rarely employ nested ln or exp operators; therefore, during the SR iterations, we exclude the possibility of such combinations. A similar SR configuration is employed for $\psi_2(I_2)$ with the following differences: the expression trees have a maximum size of 15 and a maximum depth of 5.

APPENDIX C. REMARK ON INPUT DATA NORMALIZATION

Input data normalization is a practical technique to enhance the training of machine learning models,¹⁰² particularly when the input or output has more than one feature. StandardScaler and MinMaxScaler are two common normalization methods. In StandardScaler, the input features are linearly mapped to have zero mean and unit standard deviation, that is, $(\mathbf{x} - \boldsymbol{\mu}_x)/\boldsymbol{\sigma}_x$ where $\boldsymbol{\mu}_x$ and $\boldsymbol{\sigma}_x$ are mean and standard deviation per feature, respectively. In MinMaxScaler, the input features are linearly mapped to have zero min and unit max, that is, $(\mathbf{x} - \mathbf{m}_x)/(\mathbf{M}_x - \mathbf{m}_x)$ where $\mathbf{m}_x$ and $\mathbf{M}_x$ are minimum and maximum values per feature, respectively.

Since our formulation requires the convexity of the built neural network, the following lemma is provided to ensure that common linear normalization does not affect our utilized formulation.

Lemma 9. *Affine transformation preserves convexity.*

Proof. Assuming the convexity of $f(\mathbf{x})$ in $\mathbf{x} \in \mathbb{R}^n$ and defining $g(\mathbf{x}) = f(\mathbf{A}\mathbf{x} + \mathbf{b})$ where $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{b} \in \mathbb{R}^n$ are constants, then for two arbitrary $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^n$

$$g(\gamma\mathbf{x}_1 + (1 - \gamma)\mathbf{x}_2) = f(\mathbf{A}(\gamma\mathbf{x}_1 + (1 - \gamma)\mathbf{x}_2) + \mathbf{b}), \quad (\text{C1})$$

$$= f(\gamma(\mathbf{A}\mathbf{x}_1 + \mathbf{b}) + (1 - \gamma)(\mathbf{A}\mathbf{x}_2 + \mathbf{b})), \quad (\text{C2})$$

$$\leq \gamma f(\mathbf{A}\mathbf{x}_1 + \mathbf{b}) + (1 - \gamma)f(\mathbf{A}\mathbf{x}_2 + \mathbf{b}) \quad \text{using convexity of } f, \quad (\text{C3})$$

$$= \gamma g(\mathbf{x}_1) + (1 - \gamma)g(\mathbf{x}_2). \quad (\text{C4})$$

■

The affine transformation lemma guarantees that the usual linear data normalization (standardization) in machine learning, such as `StandardScaler` and `MinMaxScaler`,¹⁰³ do not alter the convexity results.

APPENDIX D. FINITE ELEMENT DISCRETIZATION

For ease of implementation, we use penalty approach to weakly impose incompressibility. Although this method does not enforce incompressibility exactly, it leads to a simple system of nonlinear equations based solely on the displacement field. Other multi-field formulations^{104,105} are possible but they are computationally more demanding which is not considered in this work.

For numerical stability reasons, one may prefer to additively decompose the strain energy functional to volumetric ψ_{vol} and deviatoric ψ_{dev} parts, that is, $\psi(\mathbf{C}) = \psi_{\text{dev}}(\bar{\mathbf{C}}) + \psi_{\text{vol}}(J)$. In the nearly incompressible regime, the volumetric part acts as a penalty term to enforce $|J - 1| < \epsilon$. In this work, the penalty term is chosen as follows,

$$\psi_{\text{vol}} = \frac{1}{2} \kappa \ln^2(J), \quad (\text{D1})$$

where κ is the penalty parameter. Higher value of this parameter enforces incompressibility more strictly, however for numerical instability issues one may need to choose it reasonably high. The deviatoric part of the right Cauchy–Green is defined,

$$\bar{\mathbf{C}} = J^{-\frac{2}{3}} \mathbf{C}. \quad (\text{D2})$$

The unknown displacement field $\mathbf{U}(\mathbf{X})$ defined over the undeformed domain Ω_0 is the stationary point of the total energy functional when neglecting body forces and having non-zero traction boundary conditions,

$$\mathbf{U} = \arg \min_{\mathbf{U}} \Pi(\mathbf{U}(\mathbf{X})), \quad (\text{D3})$$

$$\Pi(\mathbf{U}) = \int_{\Omega_0} \psi(\mathbf{U}(\mathbf{X})) \, d\Omega. \quad (\text{D4})$$

To find the minima, we discretize the domain using hexahedral elements and use linear basis functions for the displacement field,

$$\Pi(\mathbf{U}) \approx \Pi(\bar{\mathbf{U}}) = \sum_{i=1}^{N_{\text{elem}}} \int_{\Omega_i^e} \psi(\mathbf{N}^T(\mathbf{X}) \bar{\mathbf{U}}) \, d\Omega, \quad (\text{D5})$$

where interpolation using finite element basis is utilized, that is, $\mathbf{U} = \mathbf{N}^T(\mathbf{X}) \bar{\mathbf{U}}$, and $\mathbf{N}$ is the linear basis function for the vector field in 3D. The element-level integration is performed using a second-order quadrature rule. To solve the nonlinear system of equations based on the nodal displacement $\bar{\mathbf{U}}$, we employ the Newton–Raphson method. For more detailed information, readers are referred to Belytschko et al.¹⁰⁶ Our implementation is carried out using FEniCS.¹⁰⁷