# Physics-Guided Deep Learning Enabled Surrogate Modeling for Pneumatic Soft Robots

Sameh I. Beaber, *Graduate Student Member, IEEE*, Zhen Liu, *Member, IEEE*, and Ye Sun, *Member, IEEE*

***Abstract*—Soft robots, formulated by soft and compliant materials, have grown significantly in recent years toward safe and adaptable operations and interactions with dynamic environments. Modeling the complex, nonlinear behaviors and controlling the deformable structures of soft robots present challenges. This study aims to establish a physics-guided deep learning (PGDL) computational framework that integrates physical models into deep learning framework as surrogate models for soft robots. Once trained, these models can replace computationally expensive numerical simulations to shorten the computation time and enable real-time control. This PGDL framework is among the first to integrate first principle physics of soft robots into deep learning toward highly accurate yet computationally affordable models for soft robot modeling and control. The proposed framework has been implemented and validated using three different pneumatic soft fingers with different behaviors and geometries, along with two training and testing approaches, to demonstrate its effectiveness and generalizability. The results showed that the mean square error (MSE) of predicted deformed curvature and the maximum and minimum deformation at various loading conditions were as low as $10^{-4}$ $mm^2$. The proposed PGDL framework is constructed from first principle physics and intrinsically can be applicable to various conditions by carefully considering the governing equations, auxiliary equations, and the corresponding boundary and initial conditions.**

***Index Terms*—Soft robot applications, soft actuators, modeling and control, Physics-Informed Neural Networks (PINNs).**

## I. Introduction

SOFT robots are constructed from soft and compliant materials and structures. This unique nature endows soft robots with competitive advantages such as intrinsic adaptability to diverse environments, enhanced safety and dexterity during services, and flexibility for various applications. Soft robots exhibit a theoretically infinite number of degrees of freedom (DOFs) and thus can generate highly nonlinear behaviors, which leads to challenges in accurately modeling and controlling their deformation, compliance, and behaviors [1]. As a result, soft robotics theoretical models are normally much more complicated than those for traditional rigid robots. To tackle this major problem, remarkable studies have been performed, ranging from physics/geometry-based models and their approximation and discretization [2], numerical methods such as finite element methods (FEMs) [3], sensing enabled morphological and reservoir computing [4], to pure data-driven methods using machine learning techniques [5].

Consequently, several theoretical models have been developed considering the dynamics principles such as continuum mechanics models, geometrical models, discrete material models, and surrogate models [1]. Continuum mechanics models consider soft bodies as a set of continuum particles; thus, the main idea is to equivalently simplify the representative numbers of DOFs for control inputs for driving the motion [6]. Geometrical models are usually constructed from the soft robots' geometrical shapes after applying specific loads [7], hence can describe the deformed body as a space curve defined by mathematical functions or divide the soft body into a set of circular arc shapes. Discrete element models define the soft continuum body as discrete material elements such as springs, dampers, and masses [8]. Physics-based models can provide accurate descriptions of soft robots' nonlinear behaviors; however, they may struggle to integrate seamlessly with real-time sensing and control. Approximation and discretization can reduce the DOFs of such models but may need assumption and simplification and thus may sacrifice accuracy.

Machine learning/deep learning (ML/DL) techniques can enrich soft robotics with models that are able to deal with highly nonlinear problems while maintaining the necessary accuracy [9]. Such data-driven models are suitable for real-time control; however, they rely on extensive and high-quality training data, which can be costly to collect. In addition, applying ML to soft robot models while considering the underlying physics is still one of the major challenges [10]. This is because extracting all the features from only the given data without considering the underlying physical principles may yield unrealistic predictions, especially for those out of the training range.

Physics-guided deep learning (PGDL) offers a compelling approach to overcome the limitations of using either ML or physical models in complex engineering problems. Unlike traditional ML models, PGDL embeds governing physical laws directly into the learning process, ensuring that the predictions align with fundamental physics. Generally speaking, PGDL

Received 24 June 2024; accepted 10 October 2024. Date of publication 1 November 2024; date of current version 13 November 2024. This article was recommended for publication by Associate Editor D. Shin and Editor C. Laschi upon evaluation of the reviewers' comments. This work was supported in part by National Science Foundation under Grant 2135620, and in part by Egyptian government. *(Corresponding author: Ye Sun.)*

Sameh I. Beaber is with the Department of Mechanical and Aerospace Engineering, University of Virginia, Charlottesville, VA 22903 USA (e-mail: azq6wr@virginia.edu).

Zhen Liu is with the Department of Civil and Environmental Engineering, University of Virginia, Charlottesville, VA 22903 USA (e-mail: rtv7bq@virginia.edu).

Ye Sun is with the Department of Mechanical and Aerospace Engineering, University of Virginia, Charlottesville, VA 22903 USA, and also with the Department of Electrical and Computer Engineering, University of Virginia, Charlottesville, VA 22903 USA (e-mail: dzv7sg@virginia.edu).

Digital Object Identifier 10.1109/LRA.2024.3490258

can address four main problems, solving differential equations, learning dynamics residuals, discovering governing equations, and forecasting dynamic systems [11]. The main idea is to solve partial differential equations (PDEs) that describe the physics of the system using data from analytical and/or numerical solutions and thus have the uniqueness to integrate the strengths of ML with the robustness and interpretability of physics-based models. By utilizing numerical results obtained from physical models such as FEM for learning, this approach substantially lowers the computational effort required for network training, as it is consistent with the underlying physical laws that govern the dataset. This method enhances the prediction accuracy and reduces the reliance on large datasets by embedding physical constraints directly into the learning process. However, coupling physics into deep learning is not yet well studied for soft robot modeling and control.

This study aims to develop a new PGDL-based computational framework that integrates first principle physics and deep learning as surrogate models for soft robotics. We leveraged physics-informed neural network (PINN) [12] that can solve PDEs using spatial coordinates to model intricate soft robotics and established the physical model using the Navier-Cauchy equation to construct the framework. We constructed three soft finger models with different geometries and behaviors as three testbeds. These three chosen soft fingers have been experimentally validated in previously published work. The results demonstrated high accuracy and efficiency in all three cases, which showed that our model can be generalized over a wide range of soft pneumatic systems. In addition, two training and testing approaches were employed using 5% of the total data for training and both covered and uncovered scenarios for testing. Using both approaches, our proposed PGDL model accurately predicted the base curvature and the deformation, demonstrating its robustness and generalization capabilities across different loading conditions. This study is among the first to integrate first principle physics as the governing equation into deep learning for soft robot modeling.

The remainder of this letter is organized as follows: Section II presents the proposed computational framework. Section III covers the FEA implementation for data generation, and Section IV provides the experimental setup and the results for two different training models followed by discussion and conclusion in Section V.

## II. PGDL for Soft Robotics Framework

The proposed PGDL aims to fundamentally employ PDE solving in the learning process to capture soft robots' complex behaviors. Once trained, these surrogate models can replace computationally expensive numerical simulations, significantly reducing computation time and enabling real-time control. This section presents the proposed framework in detail.

### A. Related Work

Traditional applications of PGDL use PINNs to solve PDEs that describe physical models in terms of spatial and temporal variables as long as these physical models can be intrinsically represented by the corresponding governing equations, auxiliary equations, boundary conditions, and initial conditions. Modified PINNs have been attempted in modeling soft robotics, and some researchers have adapted the PINN framework to address the unique challenges of soft robotics. Notably, Sun et al. [13] proposed the Physics-Informed Recurrent Neural Networks (PIRNN) that used a first-order linear system with a rate-independent hysteresis as the physical model and integrated it with a Recurrent Neural Network (RNN) and used the explicit Euler method to predict the output of this physical model. Then this prediction was used as additional input and output to guide the RNN learning. Liu et al. [14] employed Lagrangian and Hamiltonian formulations for modeling soft robot dynamics into NN, which allows for non-collocated control using the generalized coordinates, momenta, and the control input as the input layer.

To generalize the applications of coupling physics and ML/DL for soft robotics, a clear description of the underlying physics with the temporal and spatial coordinates that align the control (or prediction) points with the physical model's behavior is still desired to be well integrated with the learning process, which is not yet well studied.

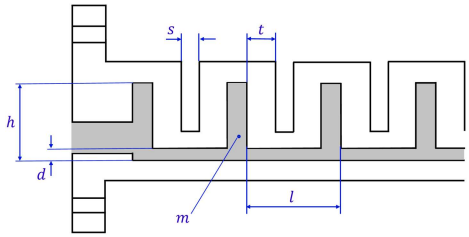



Fig. 1. Pneumatic soft finger design with PneuNets.

### B. Pneumatic Soft Finger

Pneumatic soft fingers constructed by pneumatic networks (PneuNets) are flexible and compliant actuators made from elastic or hyperelastic materials with integrated inflatable chambers and channels in certain sequences. These soft fingers are often designed and controlled to produce desired motions that can mimic human fingers for grasping and manipulation when well pressurized. Other designs such as bending joints and bellow structures have also been attempted.

This study used pneumatic soft fingers as testbeds to implement and validate the proposed PGDL framework. Three soft finger models with different geometries, materials, and behaviors were constructed as three testbeds, which were tested and validated by other studies [15], [16], [17]. FEA simulations were also performed in these three studies. The first is a silicon-based PneuNets made by [15]; the second is a 3D-printed soft finger with local bending joints [16]; and the third is a bellow-type soft actuator [17]. Fig. 1 shows the detailed design of the first case with the structure of its PneuNets as an example, and the dimensions of this design are detailed in Table I.

TABLE I
PARAMETRIC GEOMETRY OF THE PNEUNET

| PneuNets height | Channel height | Gap | Wall thickness | PneuNets length | PneuNets Number |
|---|---|---|---|---|---|
| $h$ | $d$ | $s$ | $t$ | $l$ | $m$ |
| 14 mm | 2 mm | 1 mm | 3 mm | 10.5 mm | 10 |

### C. Physical Model

Soft robots' behaviors can be complex due to their highly flexible and deformable nature. The physics underlying such mechanical behaviors of these soft structures made from elastic or hyperelastic materials can be described using the Navier-Cauchy equation [18]. For fully compliant soft robots such as PneuNets, considering an arbitrary volume, the differential form of the equilibrium equation can be given by:

$$\boldsymbol{\nabla} \cdot \boldsymbol{\sigma} + \mathbf{F} = 0 \tag{1}$$

where $\boldsymbol{\nabla}\cdot$ is the divergence operator; $\boldsymbol{\sigma}$ is the Cauchy stress tensor; $\mathbf{F}$ is the body force per unit volume caused by gravity.

The Navier-Cauchy equation in equilibrium can illustrate that the internal forces within the body and the external forces must balance out to maintain the state of equilibrium. The divergence of the stress tensor, $\boldsymbol{\nabla} \cdot \boldsymbol{\sigma}$, represents the internal forces within a material that result from stress, including the nine components of the normal and shear stresses; the stress tensor, $\boldsymbol{\sigma}$, describes how internal forces are distributed within the body, and the body force vector, $\mathbf{F}$, represented the external forces.

The Navier-Cauchy equation can be expressed in terms of the three-dimensional components, the indices $i$, $j$ take the values 1, 2, 3 for three-dimensional $x$, $y$, $z$, using Einstein summation convention as:

$$\sum_{j=1}^{3} \frac{\partial \sigma_{ij}}{\partial x_j} + F_i = \sigma_{ij,j} + F_i = 0, \quad i, j = 1, 2, 3 \tag{2}$$

where $\sigma_{ij,j}$ and $F_i$ represent the index notation of the stress tensor and the body force density, respectively; and $x_j$ represents the three spatial coordinates.

When relating stress to displacement as a dependent variable in the defined domain, Navier-Cauchy equation formulation is achieved using the stress-strain-displacement relationship. The constitutive equation that relates the stress with the strain (Hooke's law) in index notation is:

$$\sigma_{ij} = \lambda \varepsilon_{kk} \delta_{ij} + 2\mu \varepsilon_{ij} \tag{3}$$

where $\lambda$ and $\mu$ are Lame's constants; $\varepsilon_{ij}$ is the strain tensor in index notation; and $\delta_{ij}$ is the Kronecker delta:

$$\delta_{ij} = \begin{cases} 0, & \text{if } i \neq j \\ 1, & \text{i=j.} \end{cases}$$

Then, the strain-displacement relation can be given as:

$$\varepsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) = \frac{1}{2}(u_{i,j} + u_{j,i}) \tag{4}$$

where $u$ is the displacement components in the three spatial coordinates and also $i$, $j$ take the values 1, 2, 3.

Since both the stress-strain and strain-displacement relations are obtained, the stress can be represented in terms of displacement:

$$\begin{aligned} \sigma_{ij} &= \lambda \delta_{ij} \varepsilon_{k,k} + \mu \left(u_{i,j} + u_{j,i}\right) \\ &= \lambda \delta_{ij} u_{k,k} + \mu \left(u_{i,j} + u_{j,i}\right) \end{aligned} \tag{5}$$

Then, the obtained stress can be written as:

$$\begin{aligned} \sigma_{ij,j} &= \lambda u_{k,ki} + \mu \left(u_{i,jj} + u_{j,ij}\right) \\ &= \lambda u_{j,ji} + \mu \left(u_{i,jj} + u_{j,ij}\right) \\ &= (\lambda + \mu) u_{j,ji} + \mu u_{i,jj} \end{aligned} \tag{6}$$

The general formulation for describing deformations of elastic bodies in differential form can be given by substituting (6) into (2). The Navier-Cauchy equation is then obtained as a function of the displacement vector $\mathbf{u}$ and the body force vector $\mathbf{F}$ as:

$$(\lambda + \mu)\nabla(\nabla \cdot \mathbf{u}) + \mu \nabla^2 \mathbf{u} + \mathbf{F} = 0 \tag{7}$$

where Lame's constants, $\lambda$ and $\mu$, are $\lambda = 10.78$ MPa and $\mu = 0.22$ MPa. These values were obtained using the Yeoh model, where $\mu = 2C_{10}$ with $C_{10} = 0.11$ MPa, and $\lambda = \frac{2\mu\nu}{1-2\nu}$, with an approximation value for Poisson's ratio $\nu = 0.49$ [19].

In this form, the Navier-Cauchy equation contains two main terms. The first term, $(\lambda + \mu)\nabla(\nabla \cdot \mathbf{u})$, represents the volumetric response of the material, modeling the changes in bulk deformation and understanding how the body expands or contracts under an applied force. The second term, $\mu\nabla^2\mathbf{u}$, captures the internal shear deformations that explain how the body bends, twists, or experiences other complex deformations, which can be used along with the first term to describe the soft robots' elastic properties and deformation behaviors. By solving this equation within a physics-guided neural network, the model ensures adherence to physical laws, leading to more accurate and reliable predictions of the robot's behavior under various loading conditions.

In the proposed soft finger manipulations as described in Section II-B, the deformations can be considered in two dimensions (2D), $x$-$z$ plane. Then the governing equation can be given explicitly by:

$$(\lambda + \mu)\frac{\partial}{\partial x}\left(u_x + u_z\right) + \mu\left(\frac{\partial^2 u_x}{\partial x^2} + \frac{\partial^2 u_x}{\partial z^2}\right) + f_x = 0 \tag{8a}$$

$$(\lambda + \mu)\frac{\partial}{\partial z}\left(u_x + u_z\right) + \mu\left(\frac{\partial^2 u_z}{\partial x^2} + \frac{\partial^2 u_z}{\partial z^2}\right) + f_z = 0 \tag{8b}$$

where $u_x$ and $u_z$ are the deformations in $x$ and $z$ directions, respectively; and $f_x$ and $f_z$ are the body forces per unit volume caused by gravity in $x$ and $z$ directions, respectively.

This physical model starts from the first principle and provides a general physical model for fully soft and compliant robotics systems. This study focuses on studying the deformation in equilibrium aiming to provide a foundational understanding of how these structures balance internal stresses and external forces. This will offer critical insights into their stability and performance for future more complex transient behaviors.

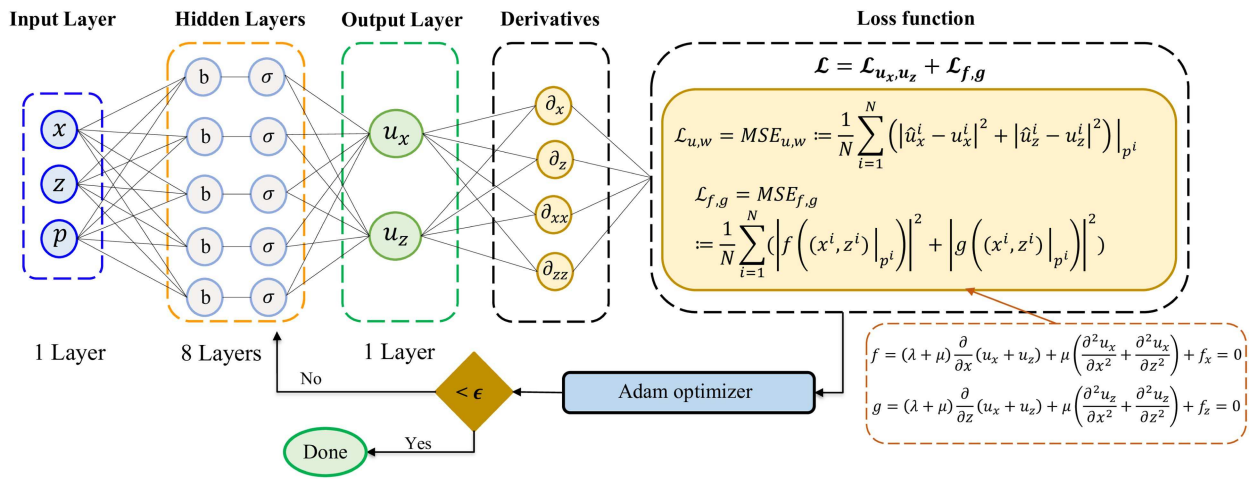



Fig. 2. The proposed PGDL framework.

### D. PGDL Framework

Coupling the physical model described in Section II-C into the deep learning process can be illustrated in our proposed PGDL framework as in Fig. 2. Through this framework, surrogate models are built via deep learning that incorporates and integrates the above physics by providing two main parts, the ML part through feedforward neural network (FNN) which is designed for static and kinematic modeling, and the physics-guided part through the physical model described using the Navier-Cauchy equation as the loss function, which enforces and constrains the output deformation from the FNN to satisfy and learn from PDEs in (8) that contain the underlying physics.

Regarding the ML part, an FNN was defined by fully connected layers including the input layer, hidden layers, and output layer to provide a complex multi-dimensional group of nested functions, which perform a linear transformation followed by a nonlinear activation function to capture the non-linearity relationship from the training data:

$$h = R_M\left(W_M, \ldots R_2\left(W_2, R_1\left(W_1 A + b_1\right)\right) \ldots + b_M\right) \quad (9)$$

where $h$ is the NN output layer, which contains the deformations $u_x$ in $x$ direction and $u_z$ in $z$ direction; $A$ is the NN input layer, which includes the pressure $p$ and the soft finger geometry $x$ and $z$; $R_i$ are the activation functions, where $i = 1, 2, \ldots, M$; and $M$ is the number of transformational layers where weights and activation functions are applied including hidden and output layers; and $W_i$ and $b_i$ are the edges' weights and biases, respectively.

The physics-guided part includes two residual functions to be minimized as $f(x, z)$ and $g(x, z)$:

$$f := (\lambda + \mu)\frac{\partial}{\partial x}\left(u_x + u_z\right) + \mu\left(\frac{\partial^2 u_x}{\partial x^2} + \frac{\partial^2 u_x}{\partial z^2}\right) + f_x \quad (10a)$$

$$g := (\lambda + \mu)\frac{\partial}{\partial z}\left(u_x + u_z\right) + \mu\left(\frac{\partial^2 u_z}{\partial x^2} + \frac{\partial^2 u_z}{\partial z^2}\right) + f_z \quad (10b)$$

Then, it proceeds by approximating $u_x$ and $u_z$ utilizing a single NN with two outputs. Incorporating this prior assumption along with (10), it leads to a physics-guided deep learning model $[f(x,z)|_p,\ g(x,z)|_p]$, where $f$ and $g$ are functions of the geometry, conditioned on the specific pressure $p$ that caused the deformation of the selected geometry node $x$ and $z$. The goal of the training process is to train the NN parameters $(u_x, u_z)|_p$ and $(f(x,z)|_p, g(x,z)|_p)$, for each corresponding pressure $p$ by minimizing the summation of the MSE of the total loss function $\mathcal{L}$:

$$\mathcal{L} = \mathcal{L}_{u_x,u_z} + \mathcal{L}_{f,g} \quad (11)$$

$$\mathcal{L}_{u_x,u_z} = MSE_{u_x,u_z} := \frac{1}{N}\sum_{i=1}^{N}\left(\left|\hat{u}_x^i - u_x^i\right|^2 + \left|\hat{u}_z^i - u_z^i\right|^2\right)|_{p^i} \quad (12)$$

$$\mathcal{L}_{f,g} = MSE_{f,g} := \frac{1}{N}\sum_{i=1}^{N}\left(\left|f(x^i,z^i)|_{p^i}\right|^2 + \left|g(x^i,z^i)|_{p^i}\right|^2\right) \quad (13)$$

where $\mathcal{L}_{u_x,u_z}$ and $\mathcal{L}_{f,g}$ are the loss terms for the deformations and the residual functions; $\{p^i, x^i, z^i, u_x^i, u_z^i\}_{i=1}^N$ represent the overall training (actual) data as vectors with a size of $N$=20,000, including the pressure $p$ that caused the deformation of the selected soft finger geometry ($x$ and $z$); exact deformation in $x$ direction ($u_x^i$) and $z$ direction ($u_z^i$); and $\hat{u}_x^i$ and $\hat{u}_z^i$ are the predicted deformations in the $x$ and $z$ direction, respectively.

### E. PGDL Framework Setup

The leading architecture of the PGDL framework was built using an FNN. In this static equilibrium problem, the PGDL consists of one input layer, eight hidden layers, and one output layer. The input layer contains the geometry of the soft finger in the $x$ and $z$ planes ($x$ and $z$) and the applied pressure ($p$). The hidden layers were tested with different sizes and ended by using eight layers with 100 neurons. The output layer is

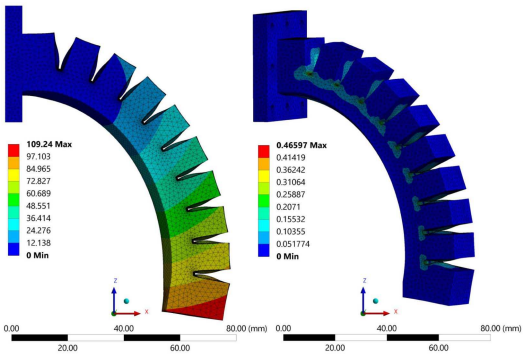



Fig. 3. FEA simulation results after applying air pressure and gravitational force on the soft finger. Left: Total deformation at 45 kPa pressure (with the unit of mm). Right: Equivalent (von-Mises) stress (with the unit of MPa).

designed to predict the deformation in the $x$ direction ($u_x$) and the deformation in the $z$ direction ($u_z$). Rectified Linear Unit (ReLU) was used as an activation function due to its ability to handle the non-linearity behavior in the data, thus effectively capturing the nonlinear behavior of the soft finger with the Navier-Cauchy equation.

The overall amount of data obtained for the finger face from FEM after applying the pressure in 100 sub-steps was 443,900. A total of 20,000 data points, corresponding to a mere 5%, were randomly selected to cover the overall data, forming the size of the inputs $x$, $z$, applied pressure $p$, and the corresponding output $\hat{u}_x$ and $\hat{u}_z$. The training started by flattening all the input and output data into a vector of size (20,000, 1), then normalized in a range of [0, 1] to ensure stable training. After prediction, the network's outputs ($\hat{u}_x$ and $\hat{u}_z$) are reshaped back to their original ranges before computing the derivatives required for the derivatives layer and the physics-based loss functions. This step is essential for accurately applying the neural network's predictions to the PDEs that govern the system and the optimization functions.

Adam optimizer was used to train the network and update the weights with a learning rate of 0.001, which was selected through preliminary experiments to balance convergence speed and stability. A batch size of 512 was implemented throughout the training process, and the training was conducted over 100,000 epochs, with the loss functions guiding the optimization process.

## III. Finite Element Analysis

The FEA of the three pneumatic soft fingers described in Section II-B was achieved by numerical simulation using ANSYS, in which the fingers are considered as a uniform continuum body. The silicon-based PneuNets model [15] was chosen to explain the FEA and the initial position was defined perpendicular to the gravity as a horizontal fixation by setting the boundary condition using a fixed support to the left side of the actuator, as shown in Fig. 3. An air pressure was then applied normal to all the internal walls. Contact among surfaces can be expected due to the material's nonlinear deformation and hyperelastic characteristics, so self-contact interactions between the outer walls were defined as frictionless surface-to-surface contact. The material was selected as a second-order hyperelastic Yeoh model with coefficients $C_1 = 0.11$ MPa and $C_2 = 0.02$ MPa with a mesh size of 3 mm, to describe the hyperelastic materials.

The simulation results of the curvature shape and the total deformation under the actuated pressure are shown in Fig. 3. Using FEA method, we created a high-resolution dataset for the training process. By applying air pressure and gravitational force at 100 sub-steps, we obtained deformation data under various loading conditions, resulting in 100 snapshots for the deformation in $x$ and $z$ coordinates. The amount of the face nodes after meshing was 4,439 nodes as $(x, z)$ points for 100 pressure snaps (steps) formulated the 443,900 overall training data (4,439×100) for all the inputs and outputs.

## IV. Experiments and Results

### A. Training and Testing Approches

Using the generated dataset from FEA, two different training and testing approaches were considered to check the efficiency of the PGDL framework. The first approach was comprehensive as the training data set was created by randomly selecting 20,000 points, a mere 5% of the total available data, to describe the entire domain of the exact solution domain after applying different pressures for each of the three soft pneumatic actuators (0-50 kPa for the first, 0-45 kPa for the second, and 0-120 kPa for the third) and testing with the remaining data. The second approach, which was more challenging, the first actuator was chosen as a case study to be able to further explore the performance of our framework by training the model at 0-44 kPa, using the same amount of data of 5%, and tested up to 50 kPa to be able to judge the ability of the framework to predict uncovered loading conditions and situations. The uncovered loading conditions were chosen to be 45 kPa, 47 kPa, and 50 kPa. Testing for both approaches was achieved by reshaping the overall data to its original 100 pressure snaps as (4,439, 100) and testing the prediction at different snapshots.

### B. PGDL Framework Testing

A detailed comparison between the FEA results and the PGDL framework is presented under different loading conditions for the three different actuators, seen and uncovered, to highlight the ability of this computational framework to address the limitations of the FEA. The results show that the PDGL framework can provide an accurate prediction and can also work in real-time control applications.

*1) First Approach:* The main idea of the first approach was to train the model till the maximum applied pressure using a small amount of the available data leveraging the power of the physical model in the learning process. Fig. 4 shows the deformation results in the $x$ direction with different loading conditions for the three different soft fingers the silicon-based PneuNets, the finger with local bending joint, and bellows-type actuator, respectively. The FEA results of the three conditions are presented in Fig. 4(a)–(c), and the PGDL predictions are shown in Fig. 4(d)–(f). The maximum and minimum deformation

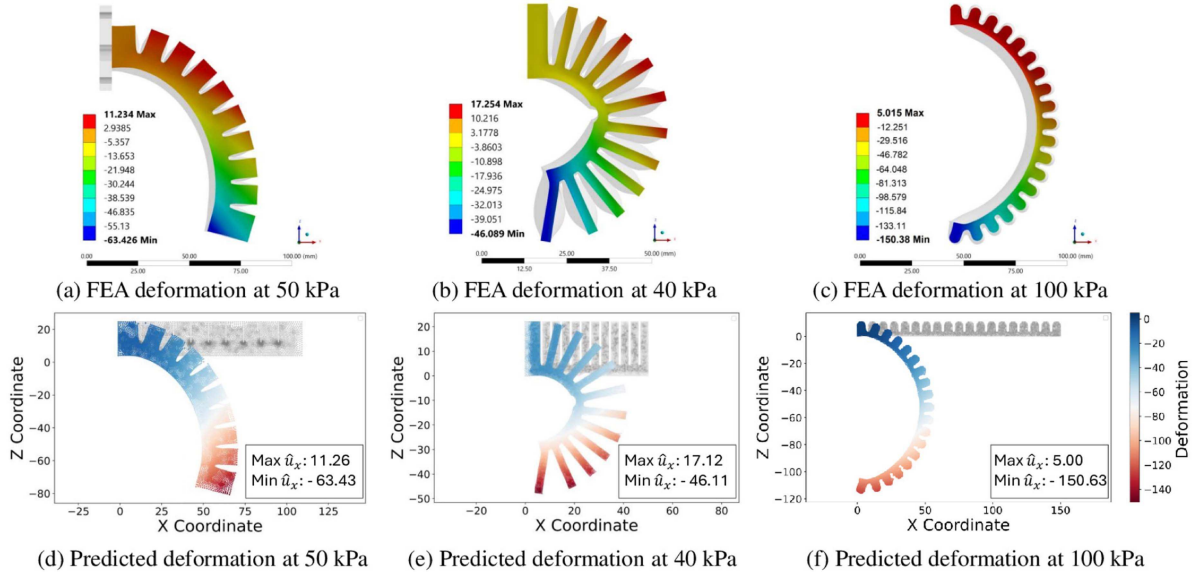



Fig. 4. Comparison of the predicted deformations from the PGDL framework and those of the FEA results for the three soft robot testbeds in $x$ direction. (a) and (d) are results for the first case; (b) and (e) are for the second case; (c) and (f) are for the third case (unit: mm).

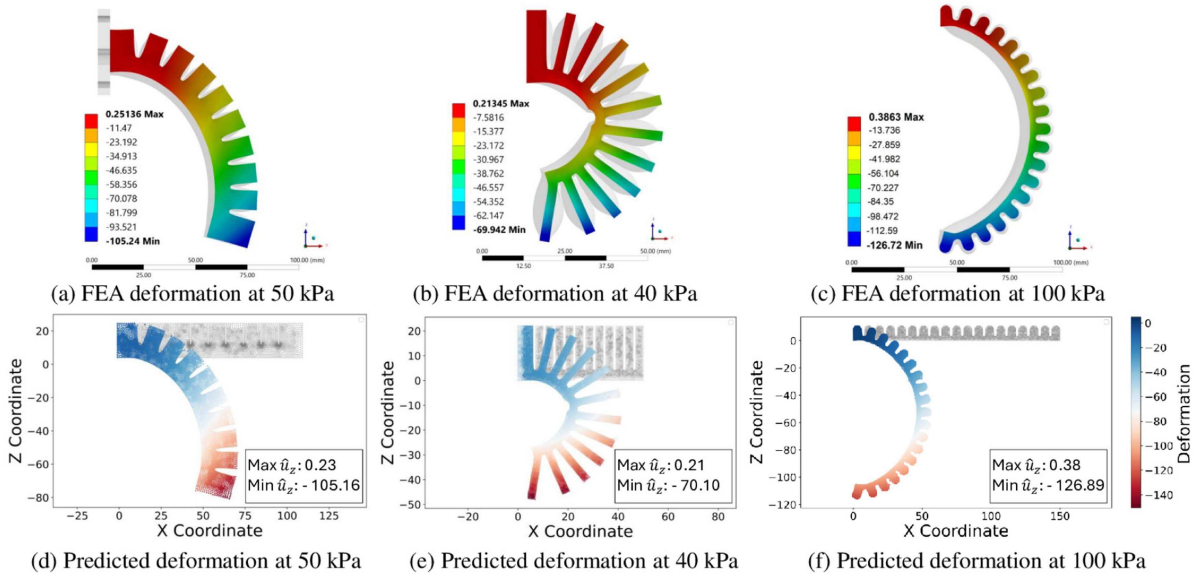



Fig. 5. Comparison of the predicted deformations from the PGDL framework and those of the FEA results for the three soft robot testbeds in $z$ direction. (a) and (d) are results for the first case; (b) and (e) are for the second case; (c) and (f) are for the third case (unit: mm).

values of FEA results and PGDL predictions were compared. Take Fig. 4(a) and (d) of the silicon-based PneuNets case for example, the maximum and minimum deformations in FEA are 11.234 mm and −63.426 mm, respectively; whereas those for PGDL prediction are 11.26 mm and −63.43 mm. In the bellows-type actuator case, as shown in Fig. 4(c) and (f), the maximum and minimum deformations in FEA are 5.015 mm and −150.38 mm, respectively; whereas those for PGDL prediction are 5.00 mm and −150.63 mm. The MSE for all three representative conditions were between $10^{-3}$ and $10^{-4}$ mm$^2$, showing the prediction accuracy.

Likewise, Fig. 5 illustrates the deformation results in $z$ direction for the same three soft fingers. Similar to the $x$ direction results, the PGDL predictions are compared with the FEA results, and the maximum and minimum deformations are calculated and compared. As shown in Fig. 5(a) and (d) that are for the silicon-based PneuNets case, the maximum and minimum deformations in FEA are 0.25 mm and −105.24 mm, respectively whereas those for PGDL prediction are 0.23 mm and −105.16 mm. In Fig. 5(b) and (e) of the finger with local bending joint case, the maximum and minimum deformations in FEA are 0.21 mm and −69.94 mm, respectively; whereas

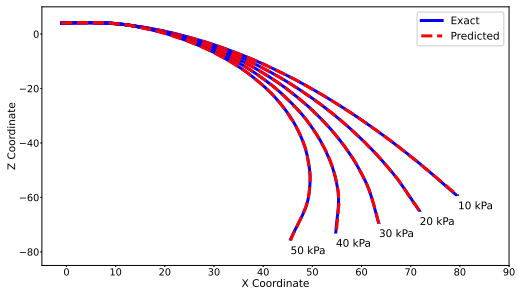
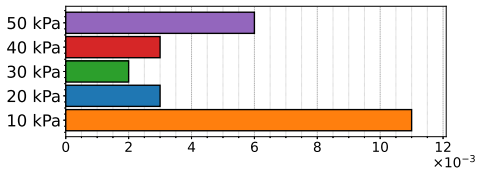
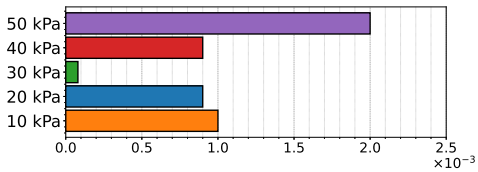



Fig. 6. Base curvatures of the soft robot at different loading conditions.




(a) MSE for $x$ directional deformation.




(b) MSE for $z$ directional deformation.

Fig. 7. MSE between the exact and actual deformation under different loading conditions (MSE unit: mm$^2$).

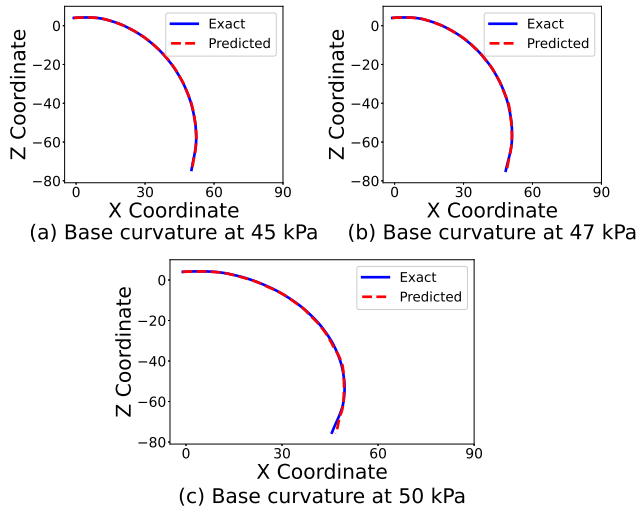



Fig. 8. Base curvatures for the uncovered loading conditions.

TABLE II
MSE OF THE UNCOVERED CONDITIONS (UNIT: MM$^2$)

| | 45 kPa | 47 kPa | 50 kPa |
|---|---|---|---|
| MSE for $x$ direction | 0.004 | 0.008 | 0.017 |
| MSE for $z$ direction | 0.002 | 0.004 | 0.007 |

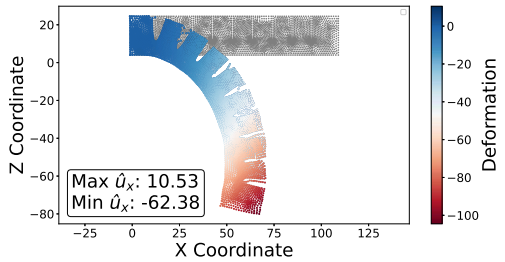
those for PGDL prediction are 0.21 mm and −70.10 mm. The MSE for the entire body is still within $10^{-3}$ and $10^{-4}$ mm$^2$ . In the experiments, we selected three soft actuators with different geometries and behaviors aiming to estimate the model's capability of predicting the deformation in different conditions without losing its accuracy.

In addition to the maximum and minimum deformation, we also used the curvature of the silicon-based PneuNets baseline to compare the results from FEA and from the PGDL prediction. The comparison under different applied air pressure conditions is shown in Fig. 6 with the air pressure of 10-50 kPa. The solid blue line is the curvature of the finger baseline in FEA whereas the red dash line is the PGDL-predicted finger baseline. The results show that the predicted baseline from the PGDL aligns with the FEA data. As depicted in Fig. 6, the actual and predicted curves overlap precisely for all tested pressures, demonstrating the model's exceptional predictive accuracy.

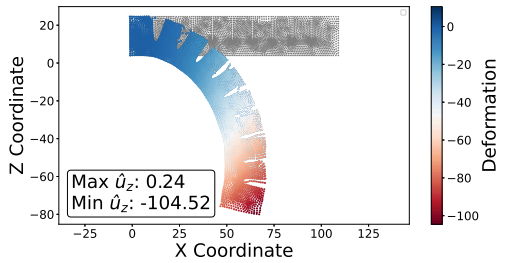
To quantitatively estimate the prediction accuracy, the MSE between the exact deformation from FEA and the predicted output from our framework was calculated. Fig. 7 lists the MSE for five conditions of the applied pressures of 10 kPa, 20 kPa, 30 kPa, 40 kPa, and 50 kPa, corresponding to the given cases in Fig. 6. The training was performed with less than 5% well distributed points over the whole body, and the MSE varied between $10^{-3}$ and $10^{-4}$ mm$^2$ for $x$ and $z$ directions. The beginning and ending cases (i.e., 10 kPa and 50 kPa) had relatively higher MSE but still in the range of $10^{-3}$ mm$^2$. The cases in between had lower MSE. The results from the first approach show that using the Navier-Cauchy equation as the physical model to capture the deformation of the soft robotic system and coupling it into the learning process enhanced the model's performance to a large extent for all the loading conditions within the training data range.

*2) Second Approach:* The second approach was performed using the silicon-based PneuNets soft actuator to perform training with a limited range of loading conditions and predict uncovered air pressure conditions to be able to further explore the full capability, robustness and generalizability of our framework. Specifically, the training was covered up to 44 kPa whereas the PGDL prediction was achieved up to 50 kPa. By training the model with a subset of the data and testing it on higher, uncovered pressures of 45 kPa, 47 kPa, and 50 kPa, we hoped to test the boundaries of the model's predictive capabilities. The results showed that this model was able to predict these uncovered conditions from limited training with low MSE. The experiment and results are detailed below.

The training for the second approach was performed up to 44 kPa and the performance at the three uncovered loading conditions (45 kPa, 47 kPa, and 50 kPa) was tested. The prediction performance was evaluated in terms of the base curvature as shown in Fig. 8. Despite not encountering these specific pressures during training, the model could still achieve high accuracy with the low MSE in the order of $10^{-3}$ mm$^2$. Table II lists the MSE values between the exact and predicted deformations for these three uncovered conditions. With such low MSE, the base curvatures for these three pressure conditions




(a) Predicted deformation in $x$ direction.




(b) Predicted deformation in $z$ direction.

Fig. 9. Predicted deformation in both directions at 50 kPa (unit: mm).

were nearly identical to the actual data. Fig. 9 presents the predicted deformation results and the maximum and minimum deformation values at 50 kPa in both $x$ and $z$ directions. The $x$ direction prediction had the MSE of 0.017 mm$^2$ whereas the $z$ direction prediction had the MSE of 0.007 mm$^2$. Compared with the FEA results and PGDL predictions from the first approach (Figs. 4(a) and (d) and 5(a) and (d)), the second approach can achieve comparable predicting performance using the proposed PGDL, which underscores the model's capability of predicting previously unencountered loading scenarios. The consistently low MSE values across these conditions demonstrate the model's robustness and capacity to maintain accuracy even when extrapolating beyond the training data.

## V. Discussion and Conclusion

This study proposed a physics-guided deep learning framework for soft robot modeling that integrates first principle physics into neural network, which offers a new approach for modeling soft robots' complex, nonlinear behaviors. This study demonstrated the effectiveness of the PGDL framework in accurately predicting the deformation and the base curvature of a pneumatic soft finger across various loading conditions. An FEA model was created to generate a high-resolution dataset by capturing the deformations under different loading scenarios. Two training and testing approaches were utilized the MSE of both approaches were between $10^{-3}$ and $10^{-4}$ mm$^2$. This study is among the first to couple first principle physics into deep learning for soft robot modeling, also there is still future work to further improve the PGDL framework.

This study starts with the formulation of equilibrium for predictions and therefore can be used for the prediction of equilibrium conditions at this stage. As a starting step, it provides insights into how soft structures balance internal stresses and external forces and is fundamental to the next-step complex transient behavior modeling using PGDL. This method intrinsically considers physics and has the capability of considering more complex situations by coupling the corresponding governing equations, auxiliary equations, initial, and boundary conditions into deep learning. Therefore, the framework's structure needs to be adaptive to consider these equations and still provide high accuracy. For future work, we will improve this proposed PGDL framework by using the transient model and including the temporal coordinates to capture the behavior over the time domain. This proposed PGDL has a high potential to be applicable and extended to different types of soft robots for soft robot modeling.

## References

[1] C. Armanini, F. Boyer, A. T. Mathew, C. Duriez, and F. Renda, "Soft robots modeling: A structured overview," *IEEE Trans. Robot.*, vol. 39, no. 3, pp. 1728–1748, Jun. 2023.
[2] F. Renda, F. Boyer, J. Dias, and L. Seneviratne, "Discrete cosserat approach for multisection soft manipulator dynamics," *IEEE Trans. Robot.*, vol. 34, no. 6, pp. 1518–1533, Dec. 2018.
[3] S. Grazioso, G. Di Gironimo, and B. Siciliano, "A geometrically exact model for soft continuum robots: The finite element deformation space formulation," *Soft Robot.*, vol. 6, no. 6, pp. 790–811, 2019.
[4] K. Nakajima, H. Hauser, T. Li, and R. Pfeifer, "Exploiting the dynamics of soft materials for machine learning," *Soft Robot.*, vol. 5, no. 3, pp. 339–347, 2018.
[5] A. Melingui, R. Merzouki, J. B. Mbede, C. Escande, and N. Benoudjit, "Neural networks based approach for inverse kinematic modeling of a compact bionic handling assistant trunk," in *Proc. 2014 IEEE 23rd Int. Symp. Ind. Electron.*, 2014, pp. 1239–1244.
[6] D. Petković, N. D. Pavlović, S. Shamshirband, and N. Badrul Anuar, "Development of a new type of passively adaptive compliant gripper," *Ind. Robot, Int. J.*, vol. 40, no. 6, pp. 610–623, 2013.
[7] D. C. Rucker, B. A. Jones, and R. J. Webster III, "A geometrically exact model for externally loaded concentric-tube continuum robots," *IEEE Trans. Robot.*, vol. 26, no. 5, pp. 769–780, Oct. 2010.
[8] M. Sfakiotakis, A. Kazakidi, A. Chatzidaki, T. Evdaimon, and D. P. Tsakiris, "Multi-arm robotic swimming with octopus-inspired compliant web," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2014, pp. 302–308.
[9] M. Giorelli, F. Renda, M. Calisti, A. Arienti, G. Ferri, and C. Laschi, "Neural network and jacobian method for solving the inverse statics of a cable-driven soft arm with nonconstant curvature," *IEEE Trans. Robot.*, vol. 31, no. 4, pp. 823–834, Aug. 2015.
[10] N. Sünderhauf et al., "The limits and potentials of deep learning for robotics," *Int. J. Robot. Res.*, vol. 37, pp. 405–420, 2018.
[11] R. Wang and R. Yu, "Physics-guided deep learning for dynamical systems: A survey," 2021, *arXiv:2107.01272*.
[12] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *J. Comput. Phys.*, vol. 378, pp. 686–707, 2019.
[13] W. Sun, N. Akashi, Y. Kuniyoshi, and K. Nakajima, "Physics-informed recurrent neural networks for soft pneumatic actuators," *IEEE Robot. Automat. Lett.*, vol. 7, no. 3, pp. 6862–6869, Jul. 2022.
[14] J. Liu, P. Borja, and C. Della Santina, "Physics-informed neural networks to model and control robots: A theoretical and experimental investigation," *Adv. Intell. Syst.*, vol. 6, no. 5, 2023, Art. no. 2300385.
[15] M. Manns, J. Morales, and P. Frohn, "Additive manufacturing of silicon based pneunets as soft robotic actuators," *Procedia CIRP*, vol. 72, pp. 328–333, 2018.
[16] M. Chahine, R. Bedran, G. Zeghondy, and C. Tawk, "A 3D printed soft gripper featuring pneumatic fingers with local bending joints," in *Proc. IEEE Int. Conf. Adv. Intell. Mechatron.*, 2024, pp. 1633–1638.
[17] H. K. Yap, H. Y. Ng, and C.-H. Yeow, "High-force soft printable pneumatics for soft robotic applications," *Soft Robot.*, vol. 3, no. 3, pp. 144–158, 2016.
[18] Z. Liu, "Mechano: Stress and strain," in *Multiphysics in Porous Materials*. Berlin, Germany: Springer, 1st ed., 2018, pp. 139–153.
[19] M. Anderson, P. Mott, and C. Roland, "The compression of bonded rubber disks," *Rubber Chem. Technol.*, vol. 77, no. 2, pp. 293–302, 2004.