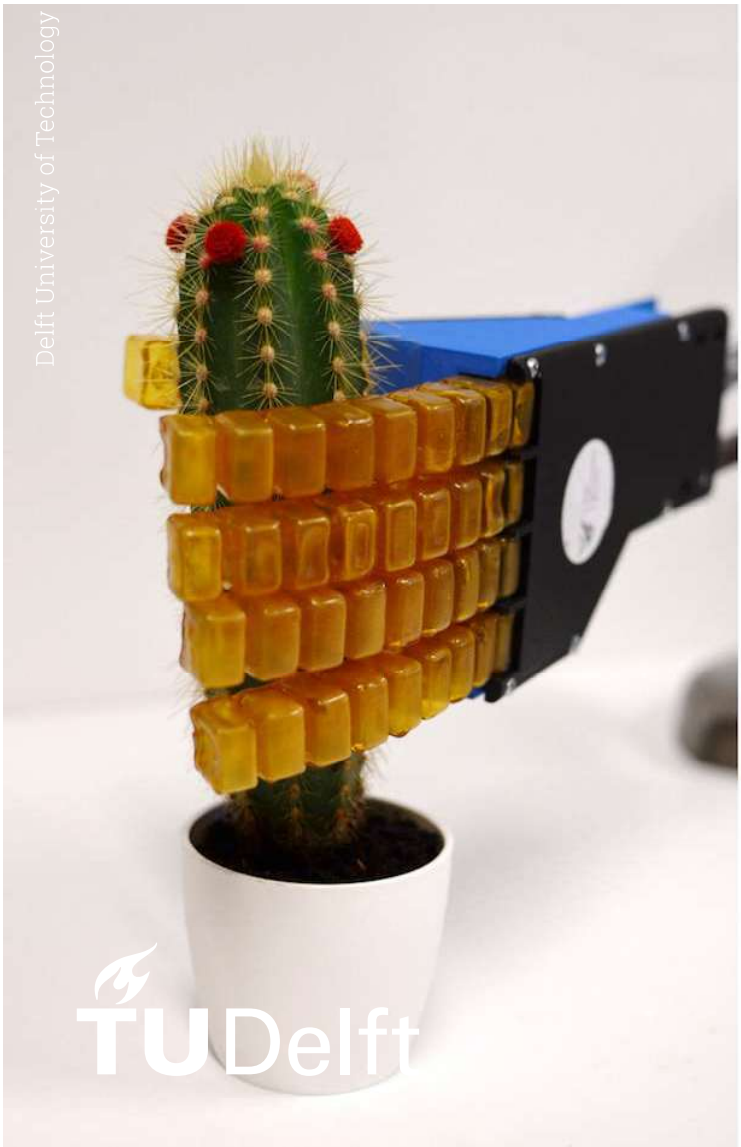
Transfer Learning for Self-Healing Soft Robots

Physics-Informed Neural Models with SVD-Based Task Subspace Adaptation

Mourelatos Eleftherios

Delft University of Technology





Transfer Learning for Self-Healing Soft Robots

Physics-Informed Neural Models with SVD-Based Task Subspace Adaptation

by

Mourelatos Eleftherios

Student Name	Student Number
Eleftherios Mourelatos	6298710

MSc Thesis Report

Supervisor: Prof. Cosimo Della Santina
Daily Supervisors: Jingyue Liu, Dr. Ebrahim ShahabiShalghouni
Faculty: Faculty of Mechanical Engineering (3mE)
Department: Department of Cognitive Robotics (CoR)

Cover : Obtained from SHERO PROJECT

TU Delft

SVD-Based Task Subspace Adaptation for Transfer Learning in Deep Lagrangian Neural Networks for Self-Healing Soft Robots

Eleftherios Mourelatos
Cognitive Robotics
Delft University of Technology
Delft, The Netherlands

***Abstract*—Soft robots enable safe and adaptable interaction with their environment, but their compliance also makes them susceptible to mechanical damage that alters system dynamics and degrades controller performance. While self-healing polymers can restore structural integrity, the repaired robot often exhibits mechanical properties that differ from its pre-damage state, resulting in a mismatch between the learned model and the underlying dynamics. This paper presents a singular value decomposition (SVD) based transfer learning framework for Deep Lagrangian Neural Networks (DeLaN) that enables efficient post-healing model adaptation. The proposed approach identifies network parameters that exhibit the greatest variation across different robot states and constructs a low-dimensional task subspace defined by the principal directions of this variation. Adaptation to the healed robot is then performed by retraining only within this subspace, significantly reducing the optimization search space and improving data efficiency. This results in lower prediction errors than competing methods when only limited training data are available and also enables continual online learning, as accurate model updates can be achieved from small batches of newly collected data. The effectiveness of the method is demonstrated through both simulation studies and experiments on a self-healing soft robotic system.**

***Index Terms*—self-healing soft robots, dynamics modeling, transfer learning, control**

I. Introduction

Traditional rigid robots lack compliance, limiting adaptability and increasing injury risk during human–robot interaction [1]. This limitation has motivated the development of soft robots inspired by biological systems, which enable safe operation in unstructured environments across applications such as agriculture [2], medicine [3], and search-and-rescue [4]. However, the compliant materials that enable these advantages are also susceptible to mechanical damage such as punctures, fatigue, and overloading [5].

Self-healing polymers have been introduced to mitigate this issue, providing tunable mechanical properties and the ability to recover from damage under suitable conditions [6], [7]. However, the healing process is inherently non-deterministic, depending on factors such as time, material alignment, and environmental conditions, which leads to variability in post-healing robot behavior [8]. More broadly, fabrication variability further limits controller consistency across nominally identical systems [9]. Prior work has demonstrated control and sensing strategies for self-healing soft robots [10], [11], yet efficient post-healing model adaptation remains largely unexplored.

Controller adaptation, following changes in the mechanical properties of soft robots, can be naturally formulated as a transfer learning task [12]. Once the self-healing process is complete, post-healing data can be collected prior to redeployment. To achieve a fully autonomous recovery pipeline, the adaptation method must incorporate newly acquired data while preserving the knowledge encoded in the original model, enabling the retrained network to accurately represent the healed system and recover the robot's original behavior as closely as possible despite changes in dynamics caused by damage and healing.

In this work, we propose an SVD-based transfer

learning framework for DeLaN [13] to enable efficient post-healing model adaptation in self-healing soft robots. The method is validated through simulation studies and physical experiments, demonstrating that targeted subspace retraining can :

- Accurately learn the robot dynamics while requiring the optimization of significantly fewer parameters than existing methods
- Recover accurate post-healing dynamics models using only limited post-healing training data
- Efficiently adapt to continual online learning from small data batches while maintaining accurate reference trajectory tracking with minimal feedback contribution

II. Related Work

A. Learning-Based Dynamics Modeling for Soft Robots

The highly nonlinear and compliant nature of soft robotic systems has motivated the widespread use of neural network-based methods for both modeling and control [14]. Purely data-driven control approaches have been proposed for soft robotic platforms, including deep learning and reinforcement learning techniques that directly map observations to control actions [15], [16]. Beyond direct control, data-driven dynamics models have been learned from state and actuator measurements and subsequently integrated with control frameworks such as sliding mode control and model predictive control (MPC) [17]–[19]. More recently, physics-informed approaches embed governing equations or structural constraints directly into the learning process [20]. Among these, DeLaN parameterize the equations of motion through the Euler–Lagrange formalism. The network decomposes system energy into kinetic and potential components while enforcing structural constraints such as a positive-definite inertia matrix and passive dissipation [21]. These structural guarantees make DeLaN particularly well-suited to soft robots, where physically consistent predictions are critical for stable control under the large deformations characteristic of compliant systems.

B. Transfer Learning in Robotics

However, such controllers are fundamentally constrained by the distribution of their training data, i.e. a model trained on one robot will accumulate substantial errors when deployed on a physically distinct unit, even one fabricated from the same mold and under the same conditions [9]. To address changes in robot dynamics caused by fabrication variability, material degradation, damage, or environmental effects, transfer learning offers a principled approach to re-adaptation. In robotics, transfer learning has been applied to both control and dynamics modeling [22].

Online adaptation methods leverage a pretrained model while identifying robot-specific parameters during deployment, enabling interchangeability across units [9]. Similarly, adaptive control techniques have been combined with neural networks, where most network parameters are kept fixed and only a subset is updated online to preserve stability while adapting to changes in system dynamics [23]. Parameter transfer has also been investigated for system identification tasks, where pretrained models are fine-tuned using limited experimental data to estimate robot-specific dynamic parameters [24].

Some studies have specifically investigated transfer learning for learned dynamics models. In [25], the authors showed that naive fine-tuning of neural network dynamics models under sparse-data conditions can increase the maximal Lyapunov exponent of the learned transition function (a measure of trajectory sensitivity to initial conditions), causing trajectories to diverge chaotically over time. In addition, a recurring issue across these works is that data-driven solutions may not be particularly adaptive by default, changes in condition, task, or morphology can require retraining from scratch rather than building upon pre-learned information [12].

As mentioned before, for dynamical systems where physical consistency and interpretability are essential, physics-informed neural networks provide a natural modeling framework by embedding governing equations into the network architecture, however, this structured formulation complicates the application of transfer learning. Transfer learning approaches can be broadly categorized into instance-based, feature-based, relation-based, and parameter-based methods [26]. The first three categories operate at the level of data distributions, learned representations, or structural correspondences between domains, and typically rely on flexible representa-

tions or transferable data that preserve consistency across domains; in physics-informed settings, even small inconsistencies can propagate through numerical integration and lead to significant trajectory errors. These characteristics, flexible representations and transferable data, are not directly accessible in physics-informed architectures such as deep lagrangian networks, where the network topology is defined by the Euler–Lagrange equations and the physical meaning of each parameter is fixed by construction [27]. Parameter-based transfer, which directly reuses or adapts the learned parameters of a source model, is therefore the most natural fit for such architectures, as it operates within the same structured parameter space while preserving the physical constraints encoded in the network.

These findings highlight that the challenge is not simply *which* parameters to transfer, but *how* to adapt them without destabilizing the learned dynamics under scarce data. A key insight from recent work in neural network adaptation is that meaningful parameter changes during fine-tuning tend to lie in a low-dimensional subspace of the full parameter space [28], that is, despite the high dimensionality of the model, the directions relevant to a new task can be captured by a small number of basis vectors. This suggests that rather than updating all parameters freely, adaptation can be constrained to this intrinsic subspace, improving data efficiency and reducing the risk of instability.

For physics-informed dynamics models such as DeLaN, this principle is particularly appealing. If task-relevant directions of parameter variation can be identified from a set of source robots, for example through SVD, adaptation to a new robot can be reformulated as optimization within a low-dimensional subspace while preserving the physical structure encoded in the model. This naturally motivates the use of low-rank adaptation methods for robotics applications.

C. Low-Rank and Subspace Methods for Neural Network Adaptation

Low-rank adaptation methods have recently gained attention as an efficient mechanism for adapting pretrained neural networks to shifted data distributions while avoiding full retraining. In robotics, particularly in soft robotic systems, such distribution shifts commonly arise due to wear, damage, environmental changes, or healing-induced changes in material properties, all of which can significantly alter the effective dynamics of the system after deployment.

In [29] they proposed a framework for fast adaptation of neural robot dynamics models that combines offline training with low-rank second-order parameter updates online, enabling rapid adaptation to novel operational conditions without full retraining. Similarly, low-rank adapters based on low-rank adaptation (LoRA) [28] have been adopted for continual adaptation of pretrained robotic policies, where parameter-efficient updates are preferred to avoid catastrophic forgetting of previously learned behaviours [30]. These works demonstrate that the low-rank adaptation principle transfers meaningfully to robotics settings, where data efficiency and stability are critical constraints.

In the context of physics-informed networks, SVD-based transfer has also been explored. In [31] they proposed SVD-PINNs, a transfer learning method for PINNs in which the singular vectors of the weight matrices are frozen after source training and only the singular values are optimized for a new target PDE. This approach explicitly exploits the spectral structure of the network weights, treating the singular vectors as encoding the structural knowledge of the source task and the singular values as the adaptation mechanism. The method was shown to reduce training cost significantly when transferring across a class of PDEs sharing the same differential operator.

Closely related to our approach is BOLT [32], which constructs an orthogonal spectral basis from the dominant singular directions of multiple source task fine-tuning updates and adapts to a new task by optimizing only a small set of coefficients within this basis. This multi-source subspace construction, identifying shared directions of variation across tasks and constraining adaptation to lie within them, is conceptually aligned with our use of SVD across multiple source robots to identify a task subspace for efficient transfer in DeLaN.

III. Problem Formulation

The dynamics of a soft robotic system can be described by [33]:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + D(q)\dot{q} = A(q)u \quad (1)$$

where $q \in \mathbb{R}^n$ denotes the generalized coordinates. The terms $M(q)\ddot{q}+C(q,\dot{q})\dot{q}+G(q)$ represent the *multibody dynamics*, where $M(q) \in \mathbb{R}^{n\times n}$ is the positive-definite inertia matrix, $C(q,\dot{q}) \in \mathbb{R}^{n\times n}$ captures Coriolis and centrifugal effects, and $G(q) \in \mathbb{R}^n$ represents the generalized conservative forces, including gravitational and elastic restoring forces arising from the compliant structure of the soft robot. The term $D(q)\dot{q}$ model the *dissipative forces*, where $D(q) \in \mathbb{R}^{n\times n}$ denotes the damping matrix accounting for dissipative effects. Finally, $A(q) \in \mathbb{R}^{n\times m}$ is the actuation matrix, which maps the control input $u \in \mathbb{R}^m$ to generalized forces and captures the underactuated nature of the system. The network is trained similarly as shown in [21].

For a static configuration, the system is at rest, where $\dot{q} = 0$ and $\ddot{q} = 0$. Under this condition, the inertial, Coriolis, and damping terms vanish, and the dynamics reduce to

$$G(q) = A(q)u \quad (2)$$

The static equilibrium condition therefore requires that the actuator-generated generalised forces $A(q)u$ balance the combined gravitational and elastic effects.

Training reduces to learning the inverse dynamics, where the predicted input is given by $\hat{u} = A^+(q)G(q)$, where A^+ is the Moore–Penrose pseudoinverse of the actuation matrix and the loss is defined as

$$\mathcal{L} = \|u - \hat{u}\|^2 \quad (3)$$

The potential compensation input for a reference position q_{ref} is then given by

$$u_{ff} = A^+(q_{ref})\, G(q_{ref}) \quad (4)$$

Let θ_0 denote the parameters of the DeLaN model trained on the nominal pre-damage robot. The adaptation problem is therefore stated as follows: given the pre-damage model θ_0 and a small dataset $\mathcal{D}_h = \{(q_i, \dot{q}_i, \tau_i)\}_{i=1}^N$ collected from the healed robot prior to redeployment, find adapted parameters θ^* that minimize the prediction error on the healed system while preserving the structure and behavior encoded in θ_0.

IV. Proposed Algorithm

A. Task Subspace Construction

Directly fine-tuning θ_0 on $\mathcal{D}_h$ may suffer from the well-known challenges of fine-tuning, including overfitting under limited data and catastrophic forgetting [35]. To address this, we exploit the observation that healing-induced parameter variation is not arbitrary but is physically constrained by the material and geometric properties of the soft robot. Therefore, we make the following assumption based on reduced order modeling theory [34] :

Assumption*: We assume that the robot exhibits similar behavior before and after healing, with any differences arising only from scaling or rotation of the behavior. Under this assumption, the corresponding set of solutions is expected to lie within a common subspace of the weight space, that is, the parameter vectors of different robot instances are related by low-rank linear transformations.*

To characterize this variation, model parameters are collected from N robot instances, including robots of the same class subject to different damage and healing states, as well as nominally identical robots that differ due to fabrication variability. Let $\Delta\theta_i = \theta_i - \theta_0$ denote the parameter deviation of the i-th instance from the nominal model. Stacking these deviations into a matrix $\Delta\Theta \in \mathbb{R}^{d\times N}$, where d is the total number of network parameters, the principal directions of variation are extracted via SVD:

$$\Delta\Theta = U\Sigma V^\top \quad (5)$$

The leading k left singular vectors $U_k \in \mathbb{R}^{d\times k}$ span the task subspace, the low-dimensional manifold capturing the dominant directions along which parameters vary within this class of soft robotic system. The rank k is selected based on the cumulative explained variance of the singular values, retaining enough principal directions to capture most of the observed variation while keeping the subspace dimension small to avoid overfitting and unnecessary deviation from the source model.

B. Subspace Adaptation

Assumption*: Since the robots belong to the same class and are expected to exhibit similar qualitative*

behavior, we assume that inter-robot variation is well-captured by the task subspace.

Given the task subspace U_k, adaptation to the healed robot is performed by restricting the parameter update to this subspace:

$$\theta = \theta_0 + U_k \alpha \tag{6}$$

where $\alpha \in \mathbb{R}^k$ is a low-dimensional vector of trainable coefficients optimized on $\mathcal{D}_h$. This reduces the effective dimensionality of the adaptation problem from d to $k \ll d$, improving data efficiency and ensuring that updates remain consistent with the directions of variation observed across this class of system. The coefficients α are optimized by minimizing the prediction error on the healed robot dataset:

$$\alpha^* = \arg\min_{\alpha} \sum_{i=1}^{N} \mathcal{L}\left(f(\mathbf{x}_i, \tau_i;\, \theta_0 + U_k\alpha),\, \mathbf{x}_i^{\text{next}}\right) + \lambda \|\alpha\|_2^2 \tag{7}$$

where $\mathbf{x}_i = (q_i, \dot{q}_i)$. where $f(\cdot\,;\theta)$ denotes the DeLaN forward pass, $\mathcal{L}$ is the prediction loss, and $\lambda\|\alpha\|_2^2$ is a regularization term that penalizes large deviations in the subspace coefficients. Since θ_0 is held fixed and only α is optimized, the pre-damage model is preserved by construction, and the adapted model is guaranteed to lie within the learned task subspace. To allow for additional adaptation, the bias parameters of the network are also considered freely trainable. The algorithm is summarized in Algorithm 1.

C. Online Adaptation Strategy

Rather than requiring a fixed dataset to be collected in full before adaptation begins, the proposed framework is also implemented in an online fashion, where the coefficients α are updated incrementally as post-healing data become available. This is practically motivated by the soft robotics setting, where collecting large amounts of data prior to redeployment is costly and time-consuming; online adaptation allows the model to begin improving from the first observations and converge progressively as more data is gathered.

Observations $(q_i, \dot{q}_i, \tau_i)$ collected from the healed robot are accumulated into a buffer $\mathcal{B}$. Once $\mathcal{B}$ reaches a minimum size $N_{\min}$, the coefficients α are updated for E epochs, each consisting of a full pass over the buffer $\mathcal{B}$, with gradients computed on the entire buffer at each epoch while θ_0 and U_k remain fixed throughout:

$$\alpha \leftarrow \alpha - \eta\, \nabla_\alpha \Big[\mathcal{L}\big(f(q_i, \dot{q}_i, \tau_i;\, \theta_0 + U_k\alpha)\big) + \lambda\|\alpha\|_2^2\Big], \quad (q_i, \dot{q}_i, \tau_i) \sim \mathcal{B} \tag{8}$$

After these E updates, the buffer is cleared and the process repeats as new data continues to arrive online.

Algorithm 1 Subspace Adaptation for Healed Soft Robot

Require: Pre-damage nominal model θ_0, healing dataset $\mathcal{D}_h = \{(q_i, \dot{q}_i, \tau_i, q_i^{\text{next}}, \dot{q}_i^{\text{next}})\}_{i=1}^{N_h}$, robot instance parameters $\{\theta_i\}_{i=1}^{N}$, subspace rank k, regularization weight λ, integration timestep Δt, learning rate η

Ensure: Adapted parameters $\theta^* = \theta_0 + U_k\alpha^*$

Phase 1: Task Subspace Construction

for $i = 1$ **to** N **do**
  Compute parameter deviation: $\Delta\theta_i \leftarrow \theta_i - \theta_0$
end for
Stack deviations: $\Delta\Theta \leftarrow [\Delta\theta_1 \mid \Delta\theta_2 \mid \cdots \mid \Delta\theta_N] \in \mathbb{R}^{d\times N}$
Compute SVD: $\Delta\Theta = U\Sigma V^\top$
Select rank k based on cumulative explained variance:

$$k = \min\left\{ r : \frac{\sum_{j=1}^{r}\sigma_j^2}{\sum_{j=1}^{d}\sigma_j^2} \geq \rho \right\}$$

Extract task subspace: $U_k \leftarrow U_{:,\,1:k} \in \mathbb{R}^{d\times k}$

Phase 2: Subspace Adaptation

Initialise $\alpha \leftarrow \mathbf{0}_k$, freeze all parameters θ_0, unfreeze the bias parameters of all network layers
Define regularized objective from Eq. 7:
repeat
  for each mini-batch $(q_i, \dot{q}_i, \tau_i, q_i^{\text{next}}, \dot{q}_i^{\text{next}}) \in \mathcal{D}_h$ **do**
    Compute adapted parameters: $\theta \leftarrow \theta_0 + U_k\alpha$
    Predict acceleration via DeLaN: $\ddot{\hat{q}}_i \leftarrow f(q_i, \dot{q}_i, \tau_i;\, \theta)$
    Integrate dynamics using RK4: $\hat{\mathbf{x}}_i^{\text{next}} \leftarrow \text{RK4}(f, \mathbf{x}_i, \tau_i, \Delta t)$
    Compute regularized loss: $\ell \leftarrow \mathcal{L}\big((\hat{q}_i^{\text{next}}, \hat{\dot{q}}_i^{\text{next}}),\, (q_i^{\text{next}}, \dot{q}_i^{\text{next}})\big) + \lambda\|\alpha\|_2^2$
    Update α (and biases b of each neuron) via gradient descent: $\alpha \leftarrow \alpha - \eta\,\nabla_\alpha \ell$
  end for
until convergence
return $\theta^* \leftarrow \theta_0 + U_k\alpha^*$

Because adaptation is confined to the low-dimensional subspace, the method requires a substantially smaller buffer to produce a stable, meaningful update than full fine-tuning of θ_0 does. This is what makes the buffered online scheme practical: full fine-tuning needs a much larger accumulation of data before updates are reliable, whereas subspace adaptation converges from a small number of buffered observations, making it naturally suited to incremental, online deployment.

V. Simulation Results

A. Simulation Setup

The simulations were conducted using the SOFA environment [36]. Since self-healing introduces stochastic behavior at the molecular level of the polymer matrix [37], accurately simulating the healing process in a virtual environment is challenging. Nevertheless, the proposed method is designed to adapt to variations in material properties, which can also arise when external stimuli are used to trigger the self-healing mechanism of the robotic finger [38]. The different material properties and therefore the different robots the proposed method was used on are shown on Table I along with a short description.

TABLE I: Linear elastic material properties of robot configurations

Robot	E (MPa)	ν	Description
1	0.699	0.440	Baseline
2	0.610	0.430	Softer
3	0.660	0.445	Moderate
4	0.760	0.455	Stiffer
5	0.850	0.465	Very stiff

To evaluate the performance of the proposed algorithm, simulation experiments were conducted using the soft finger model presented in [11]. A pseudo-random perturbation pressure signal was applied as the control input to sufficiently excite the system dynamics and enable to ensure sufficient excitation for accurate system identification. [39]. An example of the excitation signal is shown in Figure 2. The system was sampled at a period of 0.005 s.

The robot state was defined using the endpoint position relative to the base frame together with the endpoint bending angle. More specifically, the generalized coordinates were chosen as

$$q = \begin{bmatrix} x & y & z & \phi \end{bmatrix}$$

where x, y, and z denote the Cartesian coordinates of the endpoint with respect to the base frame, and ϕ denotes the endpoint bending angle relative to the base. The corresponding generalized velocities were defined as

$$\dot{q} = \begin{bmatrix} \dot{x} & \dot{y} & \dot{z} & \dot{\phi} \end{bmatrix}$$

Consequently, the full system state at a timestep t is given by:

$$x_s(t) = \begin{bmatrix} q(t) & \dot{q}(t) \end{bmatrix} \in \mathbb{R}^8$$

The DeLaN architecture used in all experiments consists of three main networks. The Lagrangian network uses two sub-networks each with three hidden layers with 64 neurons per layer and a softplus activation function for the inertia and the gravitational and elastic restoring forces. The dissipative network uses three hidden layers with 5 neurons per layer and a tanh activation function, while the input transformation network uses two hidden layers with 32 neurons per layer and a softplus activation function. The model has four configuration degrees of freedom, corresponding to the end-effector position and bending angle, and one actuator input corresponding to the pressure signal.

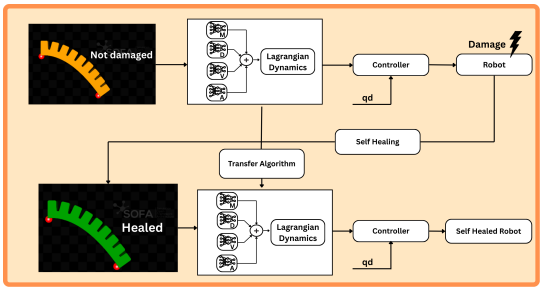


Fig. 1: Framework followed throughout the simulations and the experiments. Damage/Self Healing in the simulation corresponds to altered material properties.

B. Simulation Results

After training on the datasets collected from the different robot configurations, We evaluated adaptation performance under a leave-one-out protocol:

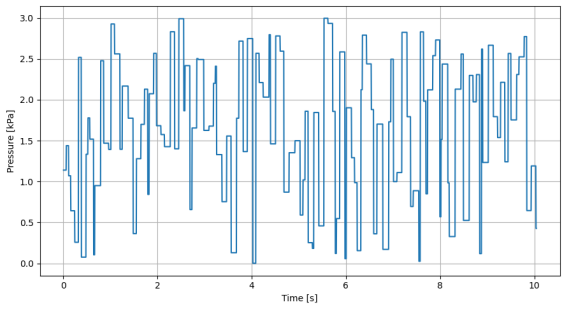


Fig. 2: Random perturbation signal to actuate the soft finger and allow us to capture the dynamics.

Robots 1–4 were used as source instances to construct the task subspace, and Robot 5 was held out entirely as the adaptation target. Two approaches were then compared on this held-out robot: (1) full fine-tuning of the network initialized with the weights obtained from the first robot, and (2) the proposed subspace adaptation method. The SVD analysis of the task subspace is shown in Figure 3. The first two singular components capture approximately 95% of the variance, indicating that the task-specific parameter changes lie in a low-dimensional subspace.

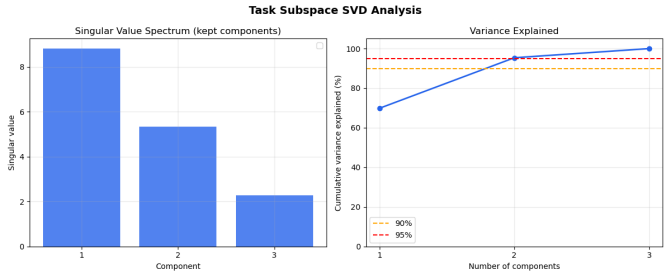


Fig. 3: SVD analysis of the task subspace. Left: singular value spectrum of the parameter deviation matrix $\Delta\Theta$. Right: cumulative explained variance as a function of retained components.

For the fifth robot, full fine-tuning was performed for 6000 epochs using the AdamW optimizer with a learning rate of 2.7×10^{-4}, weight decay of 3.2×10^{-5}, and a minibatch size of 128. For a fair comparison, the subspace method was trained under identical conditions, using the same optimizer, learning rate, weight decay. The regularization weight was set to $\lambda = 0.1$, as smaller values led to unstable training while larger values over-constrained the adaptation, preventing meaningful parameter updates.

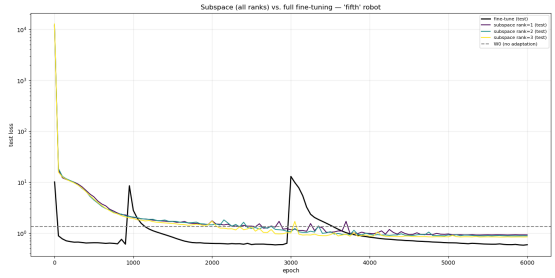


Fig. 4: Validation loss over training epochs for the proposed subspace method and full fine-tuning, evaluated on the held-out fifth robot for different adaptation ranks.

Full fine-tuning updates all $10,940$ network parameters, whereas the proposed method optimizes only $k = 2$ subspace coefficients plus 290 bias parameters, for a total of 292 trainable parameters, a reduction of approximately 97%.

Since full fine-tuning updates all network parameters, it is more prone to overfitting the training data. Therefore, the comparison is performed on the held-out validation set, which corresponds to 20% of the fifth robot dataset. The results are presented in Figure 4. As shown in the figure, the fully fine-tuned model exhibits training instability, with the validation loss showing pronounced spikes throughout optimization. Although the model ultimately converges, these oscillations indicate sensitivity to the unconstrained high-dimensional parameter updates and suggest that the optimization landscape is poorly conditioned under limited data. In contrast, the proposed method achieves smooth convergence with lower final validation error, as the low-dimensional subspace regularizes the optimization implicitly.

VI. Experimental Validation

A. Experimental Setup

The experimental setup used in this work is shown in Figure 5. The self-healing robot, presented in [40], is a silicone pneumatic soft finger. Actuation is achieved by supplying pressurized air through the internal chambers of the finger, causing

it to bend. Figure 6 illustrates the finger in both its actuated and resting states.

Air pressure was controlled using the VTEM proportional terminal [41], which regulated the air pressure supplied to the robot. To measure the bending angle of the finger, an AprilTag was attached to the fingertip and tracked using an Intel RealSense depth camera. When the endpoint AprilTag was not detected, the previous detection was used. The same camera was also used to ensure a constant distance between the robot base and the sensing system throughout all experiments. Additionally, a marker was placed at the base of the finger to ensure proper alignment with the camera. Although a second AprilTag could also be used for this purpose, a static marker was sufficient. The experimental data were acquired using a camera operating at approximately 30 FPS. Because of the limited sampling rate of only the endpoint position, the system was modeled using the static configuration described in Eq. 2.

Damage to the robot was introduced using a scalpel, with an example shown in Figure 7. The same figure also illustrates the healing process after 24 hours at room temperature. The robot state used in the experiments was:

$$q = \begin{bmatrix} x & y & \phi \end{bmatrix},$$

where x and y denote the robot's planar position and ϕ its bending angle. While the simulations additionally included the height coordinate z, the experimental setup relies solely on an overhead camera, which does not provide height measurements, the state is therefore restricted to the planar components.

B. Post-Healing Variation

To evaluate the effect of the damage–healing cycle on robot behavior, a small incision was intentionally inflicted on the robot. Following the completion of the healing process, the robot was re-evaluated under the same experimental conditions. Figure 8 compares the robot behavior before damage and after healing. The observed differences in behavior indicate that the healing process modifies the system dynamics, despite restoring the structural integrity of the robot. It should be noted that less pressure was applied to the robot after self-healing because, although the damage site had closed, the healed surface remained mechanically weaker and susceptible to reopening. All datasets from the experiments are publicly available here.

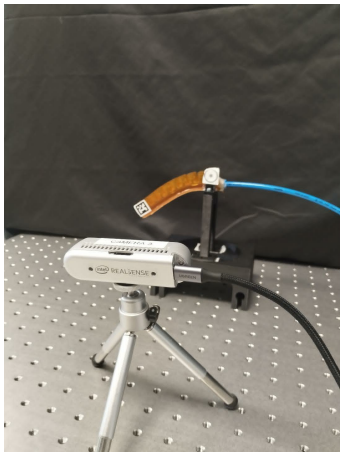


Fig. 5: Experimental setup used throughout the experiments in this study.



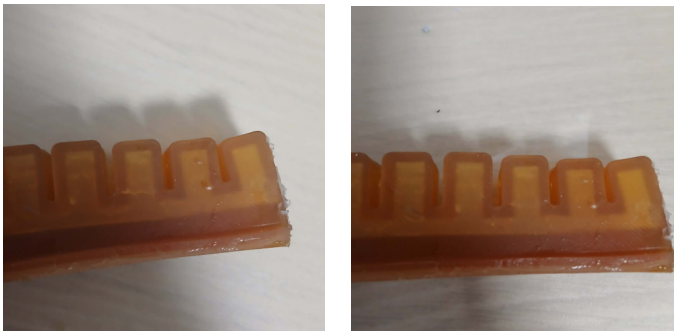
(a) Completely Resting state (b) Partially actuated state (c) Fully actuated state

Fig. 6: States of the soft robotic finger.

C. Offline Training

Given that our network now uses 2 our model eventually becomes an inverse dynamics task.

For data collection a trapezoidal input was applied to the robot. An example of this input is shown in Figure 9. The dataset also included an upward ramp and a downward ramp input; however, we observed that the trapezoidal input alone was sufficient to capture both the loading and unloading behavior of the finger, especially in the static configuration. To collect the training data, two robots were used, with each healing state also treated as a separate robot. Different physical orientations of the robot were also treated as distinct instances in the source set. This choice is motivated by the static formulation of Eq. 2, in which gravity and elastic restoring forces are the dominant terms: rotating the robot changes the direction of the gravitational



(a) Damaged (b) Healed

Fig. 7: A vertical cut was introduced along the third chamber of the robot, parallel to its backbone. The post-healing state shows successful closure of the damaged surface.

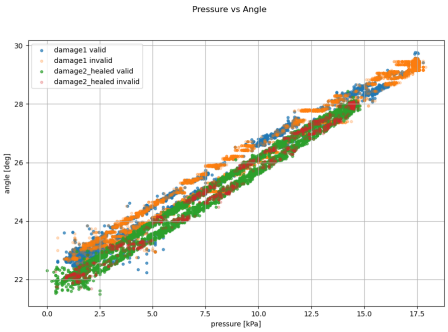


Fig. 8: Input pressure (kPa) versus bending angle (deg) for the undamaged robot and the same robot after damage and healing. Valid points correspond to successful AprilTag detections; invalid points use the last detected position. All the states can be considered static cases.

load relative to the finger's structure, producing a distinct mapping from pressure to bending angle. Each orientation therefore corresponds to a different effective dynamics, and including multiple orientations enriches the diversity of the source set used to construct the task subspace.

Similar to the simulation, in Figure 10, we analyze the variance across different robots. We observe that most of the variance among the networks is concentrated in the first six ranks, which explain 95% of the total variance. Based on this analysis, we construct the task subspace for the first phase, as shown in Algorithm 1, using $k = 6$. We then evaluate the proposed method on data collected from the

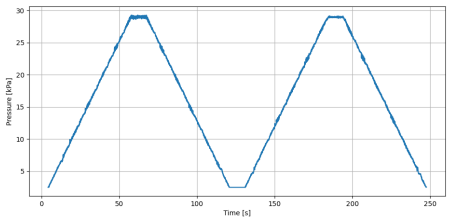


Fig. 9: Input pressure (kPa) was supplied to the robot to actuate it and enable data collection during the trial. The slight distortion observed is due to the VTEM's internal controller attempting to generate the target pressure. The dataset used for analysis was derived from the measured output pressure rather than the desired pressure.

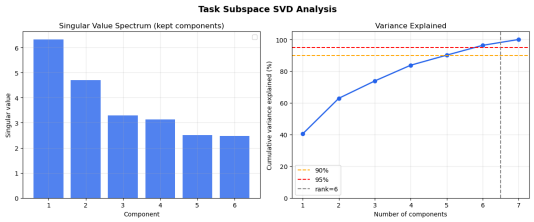


Fig. 10: SVD analysis of the task subspace across experimental robot instances. Left: singular value spectrum. Right: cumulative explained variance as a function of retained components.

robot and compare it against training from scratch, full fine-tuning, and LoRA. These networks were all formed using the static DeLaN formulation shown in 2, with the potential energy network consisting of 3 hidden layers of width 64 and the input transform network consisting of 2 hidden layers of width 32. The results are presented in Figure 11. As expected, all methods fit the data collected from the robot well in the offline task. The LoRA rank was set to *one* even so, LoRA trains 782 parameters compared to 266 for the proposed method.

To demonstrate the advantages of the proposed method, we train the networks on small fragments of the dataset instead of using randomly sampled data. This setup enables evaluation of how well each method generalizes to the overall task, since each model is trained on only a limited portion of the dataset and subsequently evaluated on the remaining unseen data. Specifically, we train using only the middle portion of each robot dataset, correspond-

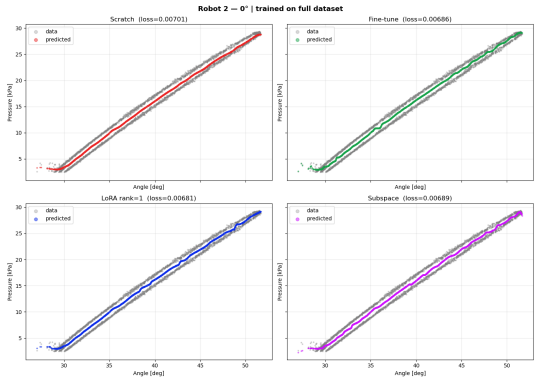


Fig. 11: Offline training of all four methods : training from scratch, full fine tuning with weights initialized from the undamaged state of the first robot, the proposed method with weights initialized with the same method and finally LoRA with rank = 1 with weights again initialized from the same.

ing to a small fraction of the available samples. An illustration of the selected data region, along with the performance of all methods, is shown in Figures 12 and 13. Unlike previous experiments, this evaluation applies the proposed method to all robots rather than only the second robot, as this task can be performed offline. Training follows a leave-one-robot-out setup, where one robot is excluded during training and later used as the target task for evaluation. The proposed subspace method achieves the lowest mean RMSE across all target tasks with the smallest variance. LoRA performs reliably but is marginally outperformed by full fine-tuning on average. In extreme cases, such as R1D3, full fine-tuning exhibits substantially larger errors, exceeding 7 kPa, highlighting its sensitivity to limited data. Training from scratch consistently yields the largest errors, as expected in the absence of prior knowledge.

D. Online Experiment

For the online training experiments, the proposed method was compared against three alternative approaches: full fine-tuning, LoRA with rank $r = 1$, and training from scratch. Each network was initialized according to its respective setup and updated according to the buffered online scheme described in Section IV-C, identically across all four methods.

The collected data consisted of the pressure applied to the robot and the corresponding bending

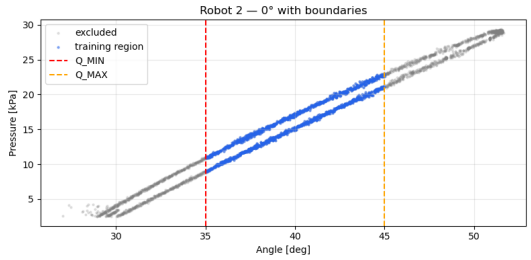
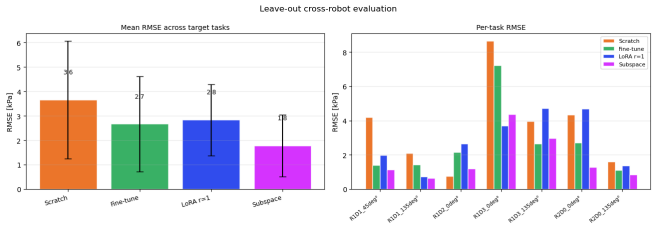
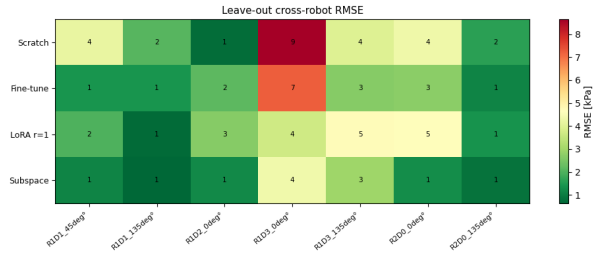


Fig. 12: Example illustrating the fraction of the dataset used for training. The blue points indicate the selected training samples within the angular bounds $Q_{\min}$ and $Q_{\max}$, while the gray points represent excluded samples outside this region.



(a) Mean RMSE versus target-data fraction.



(b) Relative improvement of Subspace over LoRA.

Fig. 13: Target-data efficiency evaluation under the middle-window sampling strategy.

angle at that instant. However, this setup made the use of purely feedforward control challenging, as no exploration of the state space was performed during data collection. As a result, we extended the control law applied to the system.

The control objective was to follow a predefined trajectory. Since the task was static, no reference velocity was provided. Additionally, no damping term was included in the controller, as the low update rate of the camera could result in large, unstable control steps when estimating velocity. Regarding the feedback gains, only the bending angle error term was assigned a nonzero value, since

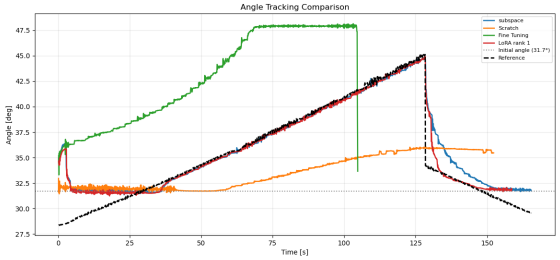


Fig. 14: Online tracking results for all four methods. Only LoRA ($r = 1$) and the proposed subspace method successfully track the reference trajectory. Full fine-tuning was terminated early to prevent damage to the robot from sustained high pressure. Training from scratch fails to adapt within the available data. The initial bending angle of the robot is $31.7°$.

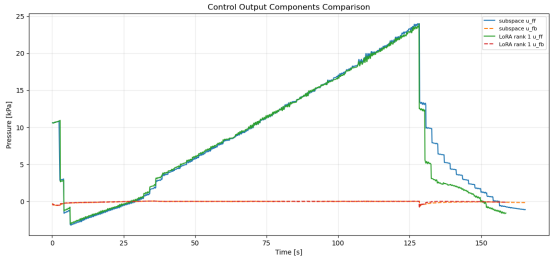


Fig. 15: Results for LoRA and our proposed subspace method, illustrating the contributions of the feedforward and feedback control terms.

the endpoint coordinates in the x and y directions were not meaningful control objectives for this task.

$$u = u_{ff} + K(q_{ref} - q) \tag{9}$$

The feedback gain was set to $K = 0.4$ throughout the experiment, with the feedforward term u_{ff} calculated from 4, computed from the continually updated network, applied jointly with the feedback term $u_{fb} = K(q - q_{ref})$ from the start. This applied identically to every method under comparison.

Figure 14 shows that only LoRA and the proposed subspace method successfully track the reference trajectory. For these two methods, the contributions of the feedforward and feedback control terms are also presented in Figure 15. In both cases, the feedforward term accounts for the majority of the control effort, with the feedback term contributing only marginally to correct residual tracking errors and improve state coverage. A subtle difference between the two methods is that LoRA tends to approach the reference more directly, whereas the subspace method converges more gradually. Finally, since the finger cannot unbend below its natural resting configuration, the reference trajectory could not be followed beyond $31.7°$. The reference intentionally extends below the robot's feasible workspace to illustrate the physical limitation imposed by the natural resting configuration.

From the above results, it is evident that only the subspace methods are capable of learning from a small dataset, as they retain prior knowledge from previously trained robots. This prior knowledge enables efficient adaptation and makes online learning feasible. In addition, both of the methods were able to adapt quickly to new information and try to keep the feedback term's contribution very low since the major contribution was due the feedforward terms.

VII. Conclusion and Discussion

This work presented an SVD-based task subspace adaptation framework for Deep Lagrangian Neural Networks, enabling efficient post-healing model recovery in self-healing soft robots. By identifying a low-dimensional subspace of parameter variation across multiple robot instances, adaptation to a healed robot can be reformulated as optimization over a small set of subspace coefficients α rather than the full parameter vector θ.

Across both simulation and physical experiments, the proposed method consistently outperformed full fine-tuning under limited post-healing data, while matching or exceeding the performance of LoRA across both experimental settings. In the offline experiment, where LoRA rank was set to 1 to give a fair comparison in terms of trainable parameters, the proposed method trained 266 parameters versus 782 for LoRA. In the online setting, only the subspace-constrained methods, ours and LoRA, were able to track the reference trajectory, with the adapted model accounting for the majority of the control effort and the feedback term contributing only marginally. By contrast, training from scratch could not recover an accurate model from the small online data batches, whereas full fine-tuning overfit the limited adaptation data and modified a large

number of parameters unnecessarily, resulting in the loss of previously learned dynamics. These results demonstrate that restricting adaptation to a low-dimensional subspace of meaningful parameter variations substantially improves data efficiency while preserving the physical consistency of DeLaN models.

The main limitations of this study are the relatively small number of source robot instances used to construct the task subspace, the restriction to the static configuration due to the sampling rate of the vision-based tracking system, and the exclusion of the embedded flex sensor due to unreliable post-healing readings. Future work will focus on expanding the diversity of source robots used for subspace construction, extending the framework to the full dynamic regime, and evaluating its generalization to other soft robot morphologies and actuation schemes. Additionally, the embedded flex resistive sensor, whose readings were excluded here due to unreliable post-healing behavior [42], presents a further adaptation target: future work will investigate whether the proposed subspace framework can also be applied to recalibrate sensor models following a damage–healing cycle.

ACKNOWLEDGMENT

The author would like to thank Prof. Cosimo Della Santina, Dr. Jingyue Liu and Dr. Ebrahim Shahabi Shalghouni for their supervision, guidance, and valuable feedback throughout this work.

REFERENCES

[1] S. Kim, C. Laschi, and B. Trimmer, "Soft robotics: a bioinspired evolution in robotics," Trends in Biotechnology, vol. 31, no. 5, pp. 287–294, May 2013, doi: https://doi.org/10.1016/j.tibtech.2013.03.002.

[2] F. Visentin, F. Castellini, and Riccardo Muradore, "A soft, sensorized gripper for delicate harvesting of small fruits," Computers and Electronics in Agriculture, vol. 213, pp. 108202–108202, Oct. 2023, doi: https://doi.org/10.1016/j.compag.2023.108202.

[3] J. Rogatinsky et al., "A multifunctional soft robot for cardiac interventions," Science Advances, vol. 9, no. 43, Oct. 2023, doi: https://doi.org/10.1126/sciadv.adi5559.

[4] E. W. Hawkes, L. H. Blumenschein, J. D. Greer, and A. M. Okamura, "A soft robot that navigates its environment through growth," Science Robotics, vol. 2, no. 8, p. eaan3028, Jul. 2017, doi: https://doi.org/10.1126/scirobotics.aan3028.

[5] S. Terryn et al., "A review on self-healing polymers for soft robotics," Materials Today, vol. 47, pp. 187–205, Jul. 2021, doi: https://doi.org/10.1016/j.mattod.2021.01.009.

[6] S. Terryn, E. Roels, J. Brancart, G. Van Assche, and B. Vanderborght, "Self-Healing and High Interfacial Strength in Multi-Material Soft Pneumatic Robots via Reversible Diels–Alder Bonds," Actuators, vol. 9, no. 2, p. 34, Apr. 2020, doi: https://doi.org/10.3390/act9020034.

[7] S. White, B. Blaiszik, S. Kramer, S. Olugebefola, J. Moore, and N. Sottos, "Self-healing Polymers and Composites," American Scientist, vol. 99, no. 5, p. 392, 2011, doi: https://doi.org/10.1511/2011.92.392.

[8] Seyedreza Kashef Tabrizian, S. Terryn, and Bram Vanderborght, "Toward Autonomous Self-Healing in Soft Robotics: A Review and Perspective for Future Research," Advanced Intelligent Systems, Feb. 2025, doi: https://doi.org/10.1002/aisy.202400790.

[9] Z. Chen, X. Ren, M. Bernabei, V. Mainardi, G. Ciuti, and C. Stefanini, "A Hybrid Adaptive Controller for Soft Robot Interchangeability," IEEE Robotics and Automation Letters, vol. 9, no. 1, pp. 875–882, Jan. 2024, doi: https://doi.org/10.1109/lra.2023.3337705.

[10] S. Terryn, J. Brancart, D. Lefeber, G. Van Assche, and B. Vanderborght, "Self-healing soft pneumatic robots," Science Robotics, vol. 2, no. 9, Aug. 2017, doi: https://doi.org/10.1126/scirobotics.aan4268.

[11] P. Ferrentino, Seyedreza Kashef Tabrizian, Joost Brancart, G. V. Assche, Bram Vanderborght, and S. Terryn, "FEA-Based Inverse Kinematic Control: Hyperelastic Material Characterization of Self-Healing Soft Robots," IEEE Robotics & Automation Magazine, vol. 29, no. 3, pp. 78–88, Dec. 2021, doi: https://doi.org/10.1109/mra.2021.3132803.

[12] Z. Chen, D. Wu, Q. Guan, D. Hardman, F. Renda, J. Hughes, T. G. Thuruthel, C. D. Santina, B. Mazzolai, H. Zhao, and C. Stefanini, "A Survey on Soft Robot Adaptability: Implementations, Applications, and Prospects [Survey]," *IEEE Robotics & Automation Magazine*, vol. 33, no. 2, pp. 183–195, Jun. 2026. doi: 10.1109/MRA.2025.3584346.

[13] M. Lutter, C. Ritter, and J. Peters, "Deep Lagrangian Networks: Using Physics as Model Prior for Deep Learning," *arXiv preprint arXiv:1907.04490*, 2019. Available: https://arxiv.org/abs/1907.04490

[14] M. T. Gillespie, C. M. Best, E. C. Townsend, D. Wingate and M. D. Killpack, "Learning nonlinear dynamic models of soft robots for model predictive control with neural networks," 2018 IEEE International Conference on Soft Robotics (RoboSoft), Livorno, Italy, 2018, pp. 39-45, doi: 10.1109/ROBOSOFT.2018.8404894. keywords: Neural networks;Soft robotics;Bladder;Valves;Atmospheric modeling;Task analysis,

[15] D. Wu et al., "Deep-Learning-Based Compliant Motion Control of a Pneumatically-Driven Robotic Catheter," in IEEE Robotics and Automation Letters, vol. 7, no. 4, pp. 8853-8860, Oct. 2022, doi: 10.1109/LRA.2022.3186497. keywords: Catheters;Robots;Robot sensing systems;Motion control;Force;Hysteresis;Robot kinematics;Robotic catheter;compliant motion control;hysteresis;LSTM;pneumatic artificial muscle,

[16] S. Satheeshbabu, N. K. Uppalapati, G. Chowdhary and G. Krishnan, "Open Loop Position Control of Soft Continuum Arm Using Deep Reinforcement Learning," 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 2019, pp. 5133-5139, doi: 10.1109/ICRA.2019.8793653. keywords: Manipulators;Load modeling;Mathematical model;Numerical models;Strain,

[17] Dimitrios Papageorgiou, Gurún óra Sigurardóttir, Egidio Falotico, Silvia Tolu, "Sliding-mode control of a soft

robot based on data-driven sparse identification", Control Engineering Practice, Volume 144, 2024, 105836, ISSN 0967-0661, https://doi.org/10.1016/j.conengprac.2023.105836. (https://www.sciencedirect.com/science/article/pii/S0967066123004057)

[18] Gillespie, Morgan & Best, Charles & Townsend, Eric & Wingate, David & Killpack, Marc. (2018). Learning nonlinear dynamic models of soft robots for model predictive control with neural networks. 39-45. 10.1109/ROBOSOFT.2018.8404894.

[19] Johnson CC, Quackenbush T, Sorensen T, Wingate D and Killpack MD (2021) Using First Principles for Deep Learning and Model-Based Control of Soft Robots. Front. Robot. AI 8:654398. doi: 10.3389/frobt.2021.654398

[20] Habich, Tim-Lukas & Mohammad, Aran & Ehlers, Simon & Bensch, Martin & Seel, Thomas & Schappler, Moritz. (2025). Generalizable and Fast Surrogates: Model Predictive Control of Articulated Soft Robots Using Physics-Informed Neural Networks. IEEE Transactions on Robotics. PP. 1-18. 10.1109/TRO.2025.3631818.

[21] Liu, Jingyue & Borja, Pablo & Della Santina, Cosimo. (2024). Physics-Informed Neural Networks to Model and Control Robots: A Theoretical and Experimental Investigation. Advanced Intelligent Systems. 6. 10.1002/aisy.202300385.

[22] Jaquier, Noémie & Welle, Michael & Gams, Andrej & Yao, Kunpeng & Fichera, Bernardo & Billard, Aude & Ude, Aleš & Asfour, Tamim & Kragic, Danica. (2024). Transfer learning in robotics: An upcoming breakthrough? A review of promises and challenges. The International Journal of Robotics Research. 44. 10.1177/02783649241273565.

[23] Elkins, Jacob & Fahimi, Farbod. (2024). Control-Theoretic Techniques for Online Adaptation of Deep Neural Networks in Dynamical Systems.

[24] Seyed Hosseini, Seyed Hamed & Hajzargarbashi, Seyedhossein & Côté, Gabriel & Liu, Zhaoheng. (2025). Transfer Learning Approach for Estimating Modal Parameters of Robot Manipulators Using Minimal Experimental Data. Vibration. 8. 65. 10.3390/vibration8040065.

[25] L. Schramm, A. Sintov and A. Boularias, "Learning to Transfer Dynamic Models of Underactuated Soft Robotic Hands," 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 2020, pp. 4579-4585, doi: 10.1109/ICRA40945.2020.9197300. keywords: Adaptation models;Robots;Data models;Neural networks;Analytical models;Friction;Predictive models

[26] F. Zhuang et al., "A Comprehensive Survey on Transfer Learning," in Proceedings of the IEEE, vol. 109, no. 1, pp. 43-76, Jan. 2021, doi: 10.1109/JPROC.2020.3004555. keywords: Transfer learning;Semisupervised learning;Data models;Covariance matrices;Machine learning;Adaptation models;Domain adaptation;interpretation;machine learning;transfer learning,

[27] Lutter, Michael & Ritter, Christian & Peters, Jan. (2019). Deep Lagrangian Networks: Using Physics as Model Prior for Deep Learning. 10.48550/arXiv.1907.04490.

[28] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-Rank Adaptation of Large Language Models," arXiv preprint arXiv:2106.09685, 2021.

[29] Altawaitan, Abdullah & Atanasov, Nikolay. (2026). Adapting Neural Robot Dynamics on the Fly for Predictive Control. 10.48550/arXiv.2604.04039.

[30] Zhu, Ruiqi & Sun, Endong & Huang, Guanhe & Celiktutan, Oya. (2025). Efficient Continual Adaptation of Pretrained Robotic Policy with Online Meta-Learned Adapters. 10.48550/arXiv.2503.18684.

[31] Y. Gao, K. C. Cheung and M. K. Ng, "SVD-PINNs: Transfer Learning of Physics-Informed Neural Networks via Singular Value Decomposition," 2022 IEEE Symposium Series on Computational Intelligence (SSCI), Singapore, Singapore, 2022, pp. 1443-1450, doi: 10.1109/SSCI51031.2022.10022281. keywords: Deep learning;Costs;Partial differential equations;Transfer learning;Neural networks;Explosives;Task analysis;Physics Informed Neural Networks;Transfer Learning;Singular Value Decomposition,

[32] Park, Junghwan & Cho, Woojin & Heo, Junhyuk & Kwon, Darongsae & Lee, Kookjin. (2025). Basis-Oriented Low-rank Transfer for Few-Shot and Test-Time Adaptation. 10.48550/arXiv.2512.02441.

[33] C. Della Santina, C. Duriez and D. Rus, "Model-Based Control of Soft Robots: A Survey of the State of the Art and Open Challenges," in IEEE Control Systems, vol. 43, no. 3, pp. 30-65, June 2023, doi: 10.1109/MCS.2023.3253419. keywords: Surveys;Musculoskeletal system;Animals;Terminology;Robot control;Soft robotics;Biomechanics;Biological systems,

[34] P. Kaushik, S. Chaudhari, A. Vaidya, R. Chellappa, and A. Yuille, "The Universal Weight Subspace Hypothesis," *arXiv preprint arXiv:2512.05117*, 2025. Available: https://arxiv.org/abs/2512.05117

[35] M. Mosbach, M. Andriushchenko, and D. Klakow, "On the Stability of Fine-tuning BERT: Misconceptions, Explanations, and Strong Baselines," *arXiv preprint arXiv:2006.04884*, 2021. Available: https://arxiv.org/abs/2006.04884

[36] F. Faure et al., "SOFA: A Multi-Model Framework for Interactive Physical Simulation," Studies in Mechanobiology, Tissue Engineering and Biomaterials, pp. 283–321, 2012, doi: https://doi.org/10.1007/8415_2012_125.

[37] Lau, Diana & Michael, Philipp & Binder, Wolfgang. (2013). Principles of Self-Healing Polymers. 10.1002/9783527670185.ch1.

[38] Li, Bingrui & Cao, Pengfei & Saito, Tomonori & Sokolov, Alexei. (2022). Intrinsically Self-Healing Polymers: From Mechanistic Insight to Current Challenges. Chemical Reviews. 123. 10.1021/acs.chemrev.2c00575.

[39] R. Mehra, "Optimal input signals for parameter estimation in dynamic systems–Survey and new results," in IEEE Transactions on Automatic Control, vol. 19, no. 6, pp. 753-768, December 1974, doi: 10.1109/TAC.1974.1100701. keywords: Parameter estimation;White noise;Design for experiments;Time domain analysis;Biology computing;Control system synthesis;Signal synthesis;Systems engineering and theory;Signal processing;Data mining,

[40] S. Terryn, J. Brancart, D. Lefeber, G. Van Assche, and B. Vanderborght, "Self-healing soft pneumatic robots," *Science Robotics*, vol. 2, no. 9, p. eaan4268, 2017. DOI: 10.1126/scirobotics.aan4268. Available: https://www.science.org/doi/abs/10.1126/scirobotics.aan4268

[41] Festo SE & Co. KG, "VTEM Proportional Terminal," [Online]. Available: https://www.festo.com/nl/nl/a/8047502/. [Accessed: 21-Jun-2026].

[42] Ellen Roels, Seppe Terryn, Joost Brancart, Fatemeh Sahraeeazartamar, Frank Clemens, Guy Van Assche, Bram Vanderborght, Self-healing sensorized soft robots, Materials Today Electronics, Volume 1, 2022, 100003, ISSN 2772-9494, https://doi.org/10.1016/j.mtelec.2022.100003.