# KineSoft: Learning Proprioceptive Manipulation Policies with Soft Robot Hands

Author Names Omitted for Anonymous Review. Paper-ID 515.

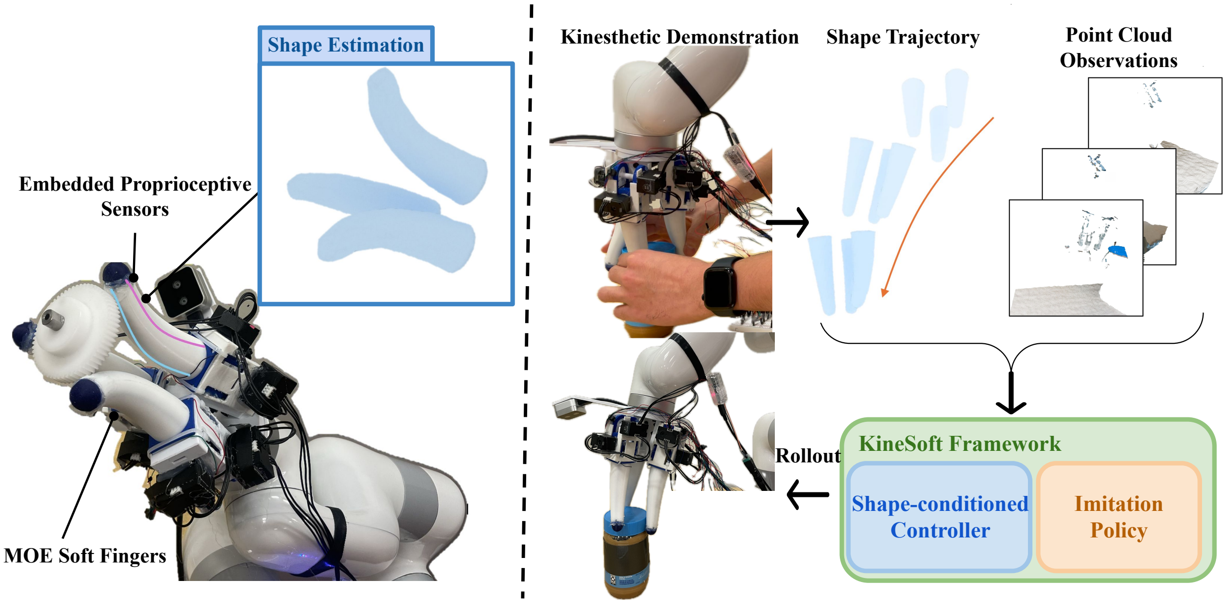

Fig. 1: **KineSoft** is a framework for learning from kinesthetic demonstration, enabling free-shaped soft end-effectors to perform dexterous in-hand manipulation. Three key components are: 1) proprioceptive model for high-fidelity shape estimation, 2) diffusion-based imitation learning for predicting the changes in shape and end-effector poses, and 3) shape-conditioned controller that allows the soft hand to track given shape trajectories.

***Abstract*—Underactuated soft robot hands offer inherent safety and adaptability advantages over rigid systems, but developing dexterous manipulation skills remains challenging. While imitation learning shows promise for complex manipulation tasks, traditional approaches struggle with soft systems due to demonstration collection challenges and ineffective state representations. We present KineSoft, a framework enabling direct kinesthetic teaching of soft robotic hands by leveraging their natural compliance as a skill teaching advantage rather than only as a control challenge. KineSoft makes two key contributions: (1) an internal strain sensing array providing occlusion-free proprioceptive shape estimation, and (2) a shape-based imitation learning framework that uses proprioceptive feedback with a low-level shape-conditioned controller to ground diffusion-based policies. This enables human demonstrators to physically guide the robot while the system learns to associate proprioceptive patterns with successful manipulation strategies. We validate KineSoft through physical experiments, demonstrating superior shape estimation accuracy compared to baseline methods, precise shape-trajectory tracking, and higher task success rates compared to baseline imitation learning approaches. KineSoft's results demonstrate that embracing the inherent properties of soft robots leads to intuitive and robust dexterous manipulation capabilities. Videos and code will be available upon final decision.**

## I. Introduction

Underactuated soft robotic hands offer significant advantages over rigid counterparts, including inherent safety through material compliance [49] and robust adaptability to uncertain object geometries [6, 14, 43]. These properties make them particularly well-suited for applications requiring close human-robot interaction, such as assistive robotics, and collaborative manufacturing [49, 11]. However, imparting dexterous in-hand manipulation skills to soft hands remains challenging. Traditional methods for soft robot manipulation often rely on hand-crafted primitives [6, 43, 1] that necessitate expert operators and limit system adaptability.

Recent advances in imitation learning, particularly frameworks like diffusion policy, have shown promise in teaching complex manipulation skills [8, 22, 15]. These approaches have been successfully applied to various tasks, from long-horizon mobile manipulation with rigid grippers [12] to deformable object manipulation with simple end-effectors [47]. Unlike reinforcement learning methods that depend on carefully crafted reward functions and simulation environments [26, 36], imitation learning requires only demonstration trajectories of successful task execution. However, collecting such demonstrations for soft hands presents challenges: traditional demonstration collection methods generally do not apply to underactuated soft systems with virtually infinite degrees of freedom. Conventional teleoperation interfaces [27], designed for rigid anthropomorphic hands, fail to capture the unique capabilities and constraints of underactuated soft end-effectors, which often lack an intuitive mapping to rigid human hand joints. Additionally, standard robot state representations for rigid robots in imitation learning frameworks [50], such as rigid transformation poses, struggle to provide meaningful state information when applied to continuously deforming structures. Despite recent advancements in expressive state representation learning for soft robots [37, 46, 48], these have not yet been applied to skill learning frameworks for in-hand manipulation. These limitations have restricted the application of imitation learning to soft robotic manipulation.

We present KineSoft, a hierarchical framework that enables direct kinesthetic teaching of soft robotic hands. Our key insight is that soft robots' natural compliance is an advantage for teaching rather than just a control challenge. This compliance allows human demonstrators to physically guide the soft robot fingers through desired movements, enabling intuitive demonstration collection that naturally accounts for the system's mechanical properties without fighting against kinematic constraints. As shown in Figure 1, KineSoft is composed of three key components. First, the proprioceptive system achieves state-of-the-art shape estimation using internal strain sensing arrays and a model trained on large simulated data of the robot's high-dimensional configurations. These sensors provide rich proprioceptive feedback while preserving the hand's natural compliance, allowing KineSoft to capture detailed information about the hand's deformation state during manipulation tasks in real time. Next, we train an imitation policy on these shape trajectories and use it to generate deformation trajectories during rollout. Finally, KineSoft's low-level shape-conditioned controller then tracks these desired shapes. Experiments demonstrate that KineSoft achieves accurate shape state estimation, precise trajectory tracking through the shape-conditioned controller, and superior performance in learned manipulation tasks through these shape-based representations.

In summary, this paper provides the following contributions:

- KineSoft, a framework for learning from kinesthetic demonstrations for soft robot hands that enables dexterous in-hand manipulation,
- A state-of-the-art proprioceptive shape estimation approach using strain sensing integrated with soft robot hands that enables precise tracking of finger deformations during contact-rich tasks,
- Shape-conditioned controller for tracking the generated deformation trajectories and performing dexterous manipulation tasks, and
- Simulated dataset and trained model for state estimation and control, which we demonstrate can be readily deployed to open-source soft robot hands, such as the MOE platform [49].

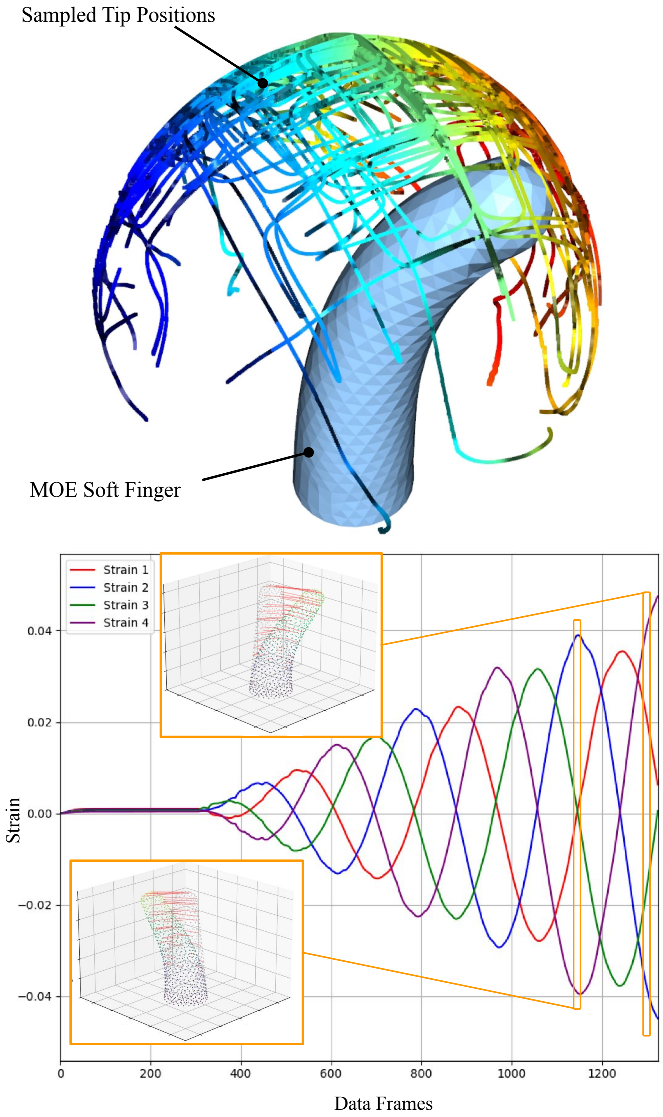

Fig. 2: **Simulated Workspace of the Soft Finger.** Top: The soft finger's workspace is visualized through sampled tip positions that form a semi-hemispherical shape, demonstrating its wide range of motion. Different colors represent different finger configurations. Bottom: example simulated sensor strain and corresponding mesh vertex displacements.

## II. Related Work

### A. Learning for In-hand Dexterity

Recent advances in reinforcement learning have driven significant progress in rigid robot in-hand dexterity [36, 26, 43, 2]. However, these approaches face challenges in real-world deployment due to difficulty in creating resettable training environments. To mitigate these issues, many methods transfer policies trained in simulation to the real world, which has shown success with rigid robots with fine tuning in the real world [26, 36] but remains less applicable for soft robots due to the complexities involved in modeling deformable materials, forward kinematics, and contact dynamics [10]. Despite recent interest in leveraging reinforcement learning for soft robot arm control and trajectory tracking [34, 29, 5], difficulty in modeling simultaneous contact and soft robot body deformation dynamics have hindered their application

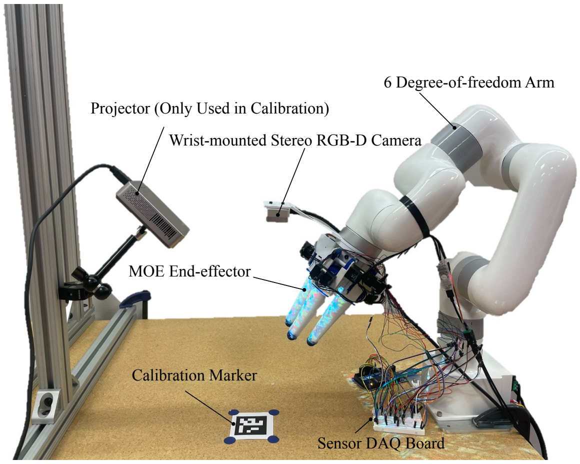

Fig. 3: **Robot setup.** A three-fingered MOE hand is mounted on a 6 degree-of-freedom robot arm. A RGB-D camera is mounted on the wrist of the arm to provide object-interaction observations. During domain alignment and shape-tracking evaluation phases, we project random color patterns on the MOE fingers to add visual features for the RGB-D camera.

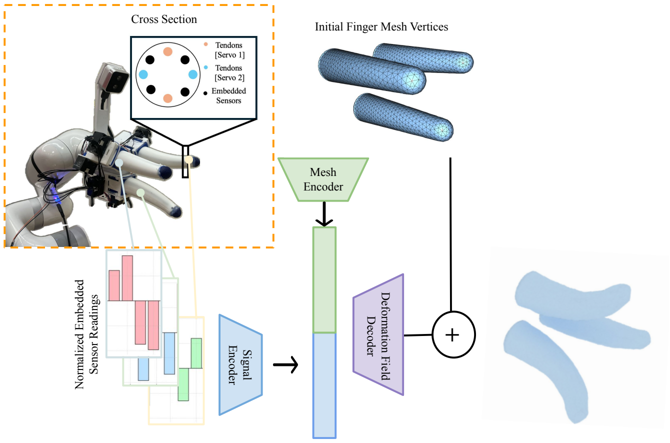

Fig. 4: **Proprioception Network Architecture.** Sensors are embedded compactly between the tendons. KineSoft's proprioception learns a mapping from the observed embedded sensors' signals to the deformation and displacement of the mesh vertices conditioned on the initial points of the mesh.

to soft robot dexterous manipulation.

Imitation learning has emerged as a promising alternative for reducing the reliance on explicit physics simulation environments, enabling robots to acquire manipulation skills efficiently with real-world data [16, 8, 50]. In-hand manipulation skills through imitation learning are typically achievable with anthropomorphic hands that provide an intuitive mapping between human hand and robot poses [3, 4, 39]. However, despite inherent benefits in safety and dexterity through compliance, which soft robot hands naturally provide, soft robots were subject to some unique challenges—particularly due to the absence of reliable proprioceptive feedback [40] and thus the lack of practical frameworks for collecting demonstrations for soft robots. To the best of our knowledge, *KineSoft* is the first framework that effectively leverages passive compliance of the soft robots to collect demonstrations and enables soft robots to acquire dexterous in-hand manipulation skills.

### B. Soft Robot Dexterity

Soft robot hands excel in grasping and manipulation tasks due to their material compliance, which allows passive adaptation to diverse object geometries [28, 53]. This compliance facilitates robust grasping and safe interactions with humans and delicate objects by distributing contact forces across large areas [1, 18, 49]. Recent advancements in soft robotics have aimed to enhance dexterity through innovative actuator designs, material improvements, and bio-inspired morphologies [25, 11]. Despite these strides, controlling soft hands remains a significant challenge due to their complex dynamics and the high-dimensional nature of their deformation spaces [44]. Consequently, learning-based approaches for soft robotic dexterity have primarily focused on grasping [13], while the development of dexterous in-hand manipulation skills has been hindered by the lack of intuitive demonstration methods and reliable proprioceptive feedback [40]. The unique advantages of soft robot hands in robust in-hand manipulation stem from their lack of rigid skeletal structures [23]. However, this absence also introduces challenges, as soft robot kinematics differ significantly from human hand motions. The *KineSoft* framework bridges these gaps by integrating novel and accurate shape estimation methods with learned imitation policies, enabling efficient skill acquisition for dexterous manipulation with soft robots.

### C. Soft Robot State Representation

Proprioceptive shape sensing is critical for enabling dexterous control in soft robots, particularly for accurate shape tracking and feedback-driven control [52]. Existing works often utilize low degree-of-freedom shape representations such as constant curvature models [32, 45] or bending angles [35], which fail to capture the full richness of soft robot deformation states. Toward capturing these complex deformation behaviors of soft robot manipulators, recent approaches based on mechanics models have employed Cosserat rod models and high-dimensional Frenet–Serret frames, associated with the continuum cross sections [20]. However, these approaches are computationally costly to update and to preserve hard constraints [20, 19]. Recent learning-based models have introduced more expressive representations of soft robot states using point clouds [37, 46] and meshes [48, 33]. However, these learned representations have not been connected to policy learning for dexterous manipulation tasks. Addressing this gap, *KineSoft* proposes a novel framework that leverages proprioceptive sensing and learned shape representations, based on vertex displacement fields over meshes, to facilitate dexterous in-hand manipulation skill learning in soft robots.

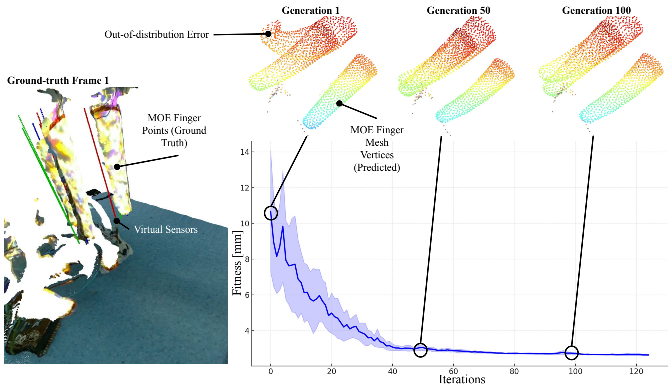

Fig. 5: **Domain alignment results.** We iteratively update the coefficients to align the real-world sensor signal distribution to the simulated sensor signals. After 200 iterations, we converged to a solution.

## III. Problem Statement

We consider the problem of dexterous in-hand manipulation with a soft robotic hand. Let $\mathbf{M}$ represent the true underlying continuous deformation state space of the hand. The hand is actuated through a set of control inputs $\mathbf{U}$, with the relationship between actuation and deformation being governed by the hand's material properties and mechanical design. The problem of learning dexterous manipulation skills consists of key challenges: i) we need an observable surrogate $\mathbf{S}$ of the true deformation state space, ii) we need a reliable mapping between the surrogate space $\mathbf{S}$ and the true deformation state space $\mathbf{M}$, and iii) we need a policy $\pi$, which maps the current state to appropriate control actions: $\pi : \mathbf{M} \times \mathcal{O} \mapsto \mathbf{U}$, where $\mathcal{O}$ represents additional task-relevant observations. This presents unique challenges for soft robots compared to rigid systems: the continuous deformation space $\mathbf{M}$ is theoretically infinite-dimensional as a continuum, demonstrations must account for the robot's inherent compliance, and the mapping between actuation and deformation is highly nonlinear or difficult to simulate in its entirety [20]. The objective is to develop a framework that can effectively learn and execute manipulation skills while embracing these fundamental characteristics of soft robots.

## IV. Method

### A. MOE Soft Robot End-effector

In this work, we leverage the MOE soft robot platform, introduced by Yoo et al. [49], which comprises modular finger units that each operate independently. Each finger is actuated by two servo motors, which apply tension to four tendons, as shown in Figure 4. This modular design provides flexibility, allowing the fingers to be rearranged into various configurations to suit specific task requirements. In this study, we examine a three-finger variant of MOE, inspired by research on object controllability using three-fingered rigid end-effectors [21].

Building on the original MOE finger design, we present an enhancement by embedding low-cost conductive elastic rubber directly into the silicone elastomer body of each finger. These sensors measure deformation by varying their electrical resistance as they stretch, providing real-time proprioceptive feedback. Each finger incorporates four of these sensors, compactly positioned between the tendons. The sensors are connected to a data acquisition (DAQ) circuit and board capable of recording resistance readings at approximately 400 Hz. By seamlessly integrating the sensors into the elastomer during fabrication, we developed a fully deformable and sensorized finger body. The combined state spaces from all twelve (12) strain sensors provides our estimate $\mathbf{S}$ of the soft robot hand's true deformation state space $\mathbf{M}$ (Section III).

In contrast to prior works [33, 24], which relied on custom sensor fabrication or specialized materials, our proposed modifications to the MOE design leverage off-the-shelf conductive rubber, eliminating the need for complex manufacturing processes, enabling straightforward integration into robotic systems, and enhancing accessibility and scalability across a wide range of applications. Furthermore, we present a shape estimation model trained on large simulated soft robot dataset, which can be deployed on sensorized MOE hands through autonomous domain alignment as discussed in later sections.

### B. Shape Estimation Model

The objective of the shape estimation model is to learn a mapping between sensor readings from embedded strain sensors $\mathbf{S}$ and the vertex displacements of the MOE fingers. Each finger contains four strain sensors arranged between the tendons, giving us $n = 12$ resistance measurements total for the three-fingered hand, denoted as $\mathbf{R} \in \mathbb{R}^n$.

The model learns a function $f$ that maps from sensor resistance readings and initial vertex positions to displacement vectors. Let $\mathbf{V}_{j,0} \in \mathbb{R}^{N \times 3}$ denote the initial vertex positions of finger $j$, where $N$ is the number of vertices in the mesh. The mapping can be expressed as:

$$f : (\mathbf{R}, \{\mathbf{V}_{j,0}\}_{j=1}^{3}) \mapsto \{\Delta\mathbf{V}_j\}_{j=1}^{3}$$

We implement this mapping using a FoldingNet-inspired architecture. The network first encodes each finger's four strain measurements into a latent representation $\mathbf{z}_j \in \mathbb{R}^{128}$ through an encoding function:

$$\mathbf{z}_j = h_{\text{enc}}(\mathbf{R}_j)$$

The decoder then processes each vertex independently by learning a mapping from the concatenated initial vertex position and latent displacement:

$$\Delta\mathbf{v}_{j,i} = h_{\text{dec}}(\mathbf{v}_{j,0}^{i}, \mathbf{z}_j),$$

where $\mathbf{v}_{j,0}^{i} \in \mathbb{R}^3$ is the initial position of vertex $i$ in finger $j$, and $[;]$ denotes concatenation. This vertex-wise processing allows the network to learn local deformation patterns while maintaining spatial relationships defined by the initial mesh topology. The deformed vertex positions are then obtained by applying the predicted displacements:

$$\mathbf{v}_{j,t}^{i} = \mathbf{v}_{j,0}^{i} + \Delta\mathbf{v}_{j,i}$$

The FoldingNet decoder architecture enables learning vertex-level deformations while leveraging the spatial structure of the initial mesh configuration. By predicting displacements rather than absolute positions, the network learns naturally centered and scaled deformations, leading to more stable training and better generalization. To train the model, we generate a large dataset of deformed meshes using SOFA (Simulation Open Framework Architecture) [41]. Our simulation setup uses a tetrahedral finite element mesh of the MOE finger with a Neo-Hookean hyperelastic material model parameterized by elastic material properties that are randomized at runtime.

We model the tendon actuation with massless, inextensible cables running through a series of fixed points within the finger body. Each tendon path is discretized into segments defined by 3D attachment points embedded in the tetrahedral mesh. The cable constraint applies forces to these points to maintain constant length while allowing sliding, effectively simulating the mechanical behavior of Bowden cable transmission. The soft body scene is solved with an implicit Euler time integration scheme and uses a conjugate gradient solver for the system matrices. We generate training data by randomly sampling tendon actuation commands within the feasible range and recording the resulting deformed vertex positions and embedded sensor strains. To simulate rich deformation behaviors including contact-like effects, we apply random external forces to the finger surface. These forces are randomly applied over time with sufficiently large radii to ensure smooth deformations that mimic natural contact interactions, without requiring explicit and difficult-to-model contacts in the scene.

The model is trained using mean squared error (MSE) loss on vertex displacements:

$$\mathcal{L} = \frac{1}{3N} \sum_{j=1}^{3} \sum_{i=1}^{N} \|\Delta \mathbf{v}_{j,i} - \Delta \mathbf{v}^*_{j,i}\|^2,$$

where $\Delta \mathbf{v}^*_{j,i}$ represents the ground truth displacement for vertex $i$ of mesh finger $j$. This choice of loss function provides strong supervision by enforcing explicit vertex-wise correspondence between predicted and ground truth meshes. Because we leverage simulated data to train the model, we can exploit the vertex-level correspondences in the meshes unlike prior works that had to rely on chamfer distance loss over real-world partial observations [38], MSE loss ensures that each vertex learns to track its specific local deformation patterns, enabling precise reconstruction of the full finger shape.

### *C. Sim-to-Real Domain Alignment*

We assume the embedded sensors are perfectly incompressible and isotropic, a common assumption in soft body mechanics for highly elastic rubber, particularly when infused with particle fillers [31]. These fillers, like those used in the off-the-shelf conductive rubbers embedded in MOE, enable the sensors to exhibit changes in resistivity when stretched. The sensors have a cylindrical shape, so we model the relationship between the cross-sectional area and the strain in the axial direction for sensor $i$ at time $t \geq 0$ with the incompressibility assumption as:

$$L_{i,0} A_{i,0} = L_{i,t} A_{i,t}, \tag{1}$$

where $L_{i,0}$ and $A_{i,0}$ are the initial length and cross-sectional area; $L_{i,t}$ and $A_{i,t}$ are the corresponding values at time $t$.

For conductive materials, resistance generally has a linear relationship with strain. The observed resistance for the sensor indexed at $i$ is given by:

$$R_{i,t} = \rho_i \frac{L_{i,t}}{A_{i,t}}, \tag{2}$$

where $\rho_i$ is the conductivity factor, assumed to be constant for sensor $i$ across time. Relating Equation 1 and Equation 2, we derive:

$$\sqrt{\frac{R_{i,t}}{R_{i,0}}} - 1 = \frac{L_{i,t} - L_{i,0}}{L_{i,0}}. \tag{3}$$

This relationship is independent of the material conductivity $\rho_i$, enabling a direct mapping from observed resistance to strain. However, in real-world applications, fabrication imperfections, such as connecting wires to the DAQ boards, can introduce errors into the initial length of the embedded sensors. These imperfections result in a deviation between the real sensor lengths ($L^R_{i,0}$, $L^R_{i,t}$) and simulated sensor lengths ($L^S_{i,0}$, $L^S_{i,t}$):

$$L^R_{i,0} = L^S_{i,0} + \epsilon_i, \quad L^R_{i,t} = L^S_{i,t} + \epsilon_i,$$

where $\epsilon_i$ is a constant error specific to each sensor $i$. This error propagates to the strain relationship as:

$$\frac{L^R_{i,t} - L^R_{i,0}}{L^R_{i,0}} = \frac{1}{1 + \frac{\epsilon_i}{L^S_{i,0}}} \cdot \frac{L^S_{i,t} - L^S_{i,0}}{L^S_{i,0}} \tag{4}$$

The constant factor $\frac{1}{1+\frac{\epsilon_i}{L^S_{i,0}}}$ can be denoted as $\kappa_i \in \kappa$, representing the constant correction factor for sensor $i$. Substituting this into Equation 3, we obtain:

$$\sqrt{\frac{R_{i,t}}{R_{i,0}}} - 1 = \kappa_i \frac{L^S_{i,t} - L^S_{i,0}}{L^S_{i,0}}, \tag{5}$$

where the observed resistances $R_{i,t}, R_{i,0}$ are measured with the DAQ setup. For the $n$ embedded sensors, aligning the simulated and observed distributions involves optimizing the constant correction parameters $\kappa_0, \kappa_1, \ldots, \kappa_{n-1}$. The objective is defined as:

$$\underset{\kappa_0,\ldots,\kappa_{n-1}}{\arg\min} \sum_{i=0}^{n-1} \sum_{t=0}^{T-1} \left( \sqrt{\frac{R_{i,t}}{R_{i,0}}} - \kappa_i \frac{L^S_{i,t} - L^S_{i,0}}{L^S_{i,0}} - 1 \right)^2, \tag{6}$$

where $T$ is the number of time steps for which resistance data is observed.

Finally, because the corresponding simulated lengths for an observed resistance are not available (i.e., there is no direct sim-to-real correspondence between the observed resistance values and the simulated lengths), we instead optimize the correction factors $\kappa_0, \kappa_1, \ldots, \kappa_{n-1}$ by minimizing the Chamfer distance between the observed points on the fingers and the model's predicted surface points. This optimization assumes that there is a unique mapping between the internal strain of the sensors and the deformation of the soft structure, which is inferred through the observed resistance values.

The model defines a mapping from the sensor resistance observations to the surface mesh of the fingers based on the predicted displacements of the vertices. However, in the real-world deployment of the model as shown in the Figure 3, we do not have access to labeled pairs of real-world resistance values $\mathbf{R}$ and $\mathbf{L}^{\mathbf{R}}$:

$$\mathcal{L}_{\text{UCD}} = \sum_{j=0}^{m-1} \sum_{\mathbf{p}_{\text{obs}} \in \mathcal{P}_{\text{obs}}} \min_{\mathbf{v} \in \mathbf{V}_{j,t}} \|\mathbf{p}_{\text{obs}} - \mathbf{v}\|^2, \qquad (7)$$

where $\mathcal{P}_{\text{obs}}$ is the set of observed points on the surface of the fingers from the RGB-D camera feed and $\mathbf{V}_{j,t}$ are vertices of the predicted surface mesh of finger $j$ defined by $\mathbf{M}_{j,t} = f(\mathbf{R}, \mathbf{V}_{j,0})$. We compute $\mathcal{L}_{\text{UCD}}(\kappa)$ where the correction factors $\kappa_i$ are adjusted such that the predicted surface points align with the observed deformation of the fingers.

The function $f$ is differentiable and thus gradient descent-based optimization is possible here for locally optimal $\kappa$. However, because of the noisy loss landscape and relatively cheap inference cost, we achieved better performance by optimizing $\kappa$ with a sample-based evolutionary strategy as shown in Algorithm 1. By optimizing for the alignment factors $\kappa$, the model effectively aligns the observed resistance-to-strain relationship with the real-world deformation states, ensuring robust sim-to-real transfer for the shape estimation model.

**Algorithm 1** Domain Alignment Optimization

```
Input: Shape predictor $f_\theta$, transformations $\{T\}$, observations
  $\mathcal{P}_{\text{obs}}$, initial resistances $R_0$
  Initialize $\theta = \{\kappa \in \mathbb{R}^{24}, \phi \in \mathbb{R}^3\} \leftarrow \mathbf{0}$
  Initialize CMA-ES optimizer $\mathcal{O}(\theta)$
  while not converged do
      Sample candidates $\{\theta\} \sim \mathcal{O}$
      for each $\theta$ do
          $\Delta R \leftarrow \sqrt{R/R_0} - 1$   ▷ Strain model for resistances
          Apply correction: $S \leftarrow \Delta R \cdot \kappa_{\mathbb{I}[\Delta R<0]}$
          $\mathbf{V} \leftarrow f_\theta(S)$   ▷ Predict vertices
          $\mathcal{P}_{\text{pred}} \leftarrow \bigcup \mathbf{V} \cdot T \cdot [R_y(\phi)|\mathbf{0}]$   ▷ Combine
          $\ell \leftarrow \mathcal{L}_{\text{UCD}}(\mathcal{P}_{\text{obs}}, \mathcal{P}_{\text{pred}})$   ▷ Compute error
      end for
      Update $\mathcal{O}$ with candidates and losses
  end while
  return $\theta^*$ with minimum loss
```

### D. Shape-conditioned Controller

The shape-conditioned controller leverages real-time proprioceptive mesh state estimation of the MOE fingers to track desired shape trajectories. For each finger $j$, the controller compares the current estimated vertex positions $\mathbf{V}_t$ with desired target positions $\mathbf{V}_t^D$ generated from the policy trajectory. The shape residual is calculated over the corresponding vertices of the meshes:

$$\mathbf{e}_{j,t} = \mathbf{V}_{t,j}^D - \mathbf{V}_{t,j}$$

Each finger is actuated by a pair of antagonistic tendons controlled by two servos. The actuation directions for each servo pair are represented by unit vectors $\mathbf{d}_{2j}, \mathbf{d}_{2j+1} \in \mathbb{R}^2$ that capture the primary deformation modes. The servo adjustments $\delta u_{j,t}$ for each finger are computed by projecting the shape error onto these actuation directions:

$$\delta u_{j,t} = k_p \sum_n \mathbf{e}_{j,t}^n \cdot [\mathbf{d}_{2j}, \mathbf{d}_{2j+1}]^T$$

where $k_p$ is a scalar gain and $\mathbf{e}_{j,t}^n$ is the error for vertex $n$. In deployment, the actions are clipped to prevent overloading the actuators. This controller runs at 100Hz with the shape estimation at each step, enabling responsive shape trajectory tracking. By projecting shape errors onto fitted actuation directions, the controller effectively translates desired deformations into appropriate servo commands despite the complex relationship between tendon actuation and finger deformation.

### E. Imitation Policy

The shape estimation model from Section IV-B provides proprioceptive feedback through predicted vertex positions $\mathbf{V}_t$. The wrist-mounted RGB-D camera captures point cloud observations of the manipulation workspace. Using these complementary sources of state information, we train a diffusion policy to learn manipulation skills via imitation learning.

The policy predicts actions that combine shape deformations and end-effector pose changes:

$$a_t = \{\Delta\mathbf{V}_t, \Delta\mathbf{p}_t\}$$

The state representation combines two different point-based encodings:

$$s_t = \{h_{\text{shape}}(\mathbf{V}_t), h_{\text{pc}}(\mathcal{P}_t), \mathbf{p}_t\},$$

where $h_{\text{shape}}$ processes the predicted vertex positions using MLPs that leverage vertex correspondence, $h_{\text{pc}}$ is a DP3 point cloud encoder that processes subsampled RGB-D observations $\mathcal{P}_t$, and $\mathbf{p}_t$ is the current end-effector pose.

The diffusion model iteratively denoises a Gaussian distribution into meaningful actions through a reverse process:

$$a_{t-1} = \mu_\theta(a_t, s_t, t) + \sigma_t \mathbf{z},$$

where $\mu_\theta$ is a learned denoising model and $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$.

The trained policy works with the shape-conditioned controller from Section IV-D, which maps the policy's deformation predictions to actuator commands. This hierarchical approach enables robust manipulation by combining learned high-level behaviors with precise low-level tracking.

## V. Experiments

### A. Baselines

**Shape-estimation Baselines.** For shape estimation, we compare with analytical and learning-based baselines:

- ***Constant Curvature Model*** [9, 45]. Constant curvature model is a common representation for the continuum deformation behavior of soft robot that parametrizes the

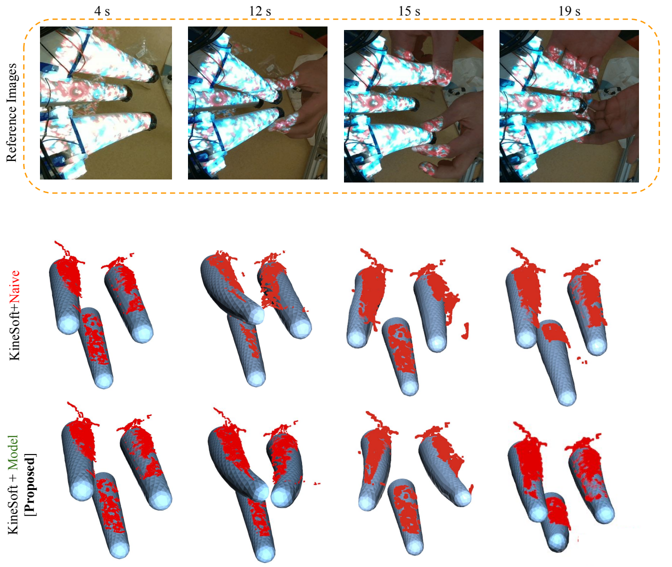

Fig. 6: **Shape Estimation Fidelity Results.** We compare the reconstructed shapes against baselines from literature and ablate with our model-based approach from Section IV-B.

shape with a single curvature curve [51]. Typically, the independent parameters of the state of the robot are defined by $r_{curve}$ and $\theta_{curve}$. Assuming a constant length, $L_{curve}$ of the robot, we get the constraint:

$$L_{curve} = r_{curve}\theta_{curve}.$$

In typical applications, additional term $\phi_{curve}$ is introduced to represent the plane of bending [17]. We implemented this simplified representation for soft robot shape using the proposed strain model as outlined in Section IV-C and fitting $r_{curve}$ and $\theta_{curve}$ to the observed strains in each side of the curve. We transformed the cross-section boundary to the curve during the evaluation and measured the chamfer distance to the reference.

- ***DeepSoRo*** [38]. DeepSoRo architecture deploys a FoldingNet [42] decoder conditioned on visual observations to predict the current shape of a deformable body. Crucially, it is trained with chamfer distance and originally trained on partial real-world shape observations, resulting in partial point cloud reconstruction outputs without frame-to-frame correspondences. Additionally, the model directly outputs the point cloud positions in contrast to KineSoft, which learns a deformation field and produces vertex displacement with frame-to-frame correspondences. We augment DeepSoRo for evaluation by training the model on KineSoft's simulated training data and using the proposed domain alignment process.

**Shape-tracking Baselines.** For shape tracking and task performance evaluation we provide the results against the following:

- ***Strain Policy.*** Strain policy, based on prior works that directly use sensor readings without intermediate representations for learning manipulation policies [30], uses raw sensor measurements instead of reconstructed shapes. For shape tracking evaluation, we modified the low-level controller from Section IV-D to track reference sensor readings directly through proportional tendon actuation. For task performance evaluation, we trained a diffusion policy using the same 50 demonstrations we use for KineSoft, but with raw strain signals and wrist-mounted camera observations as input states.

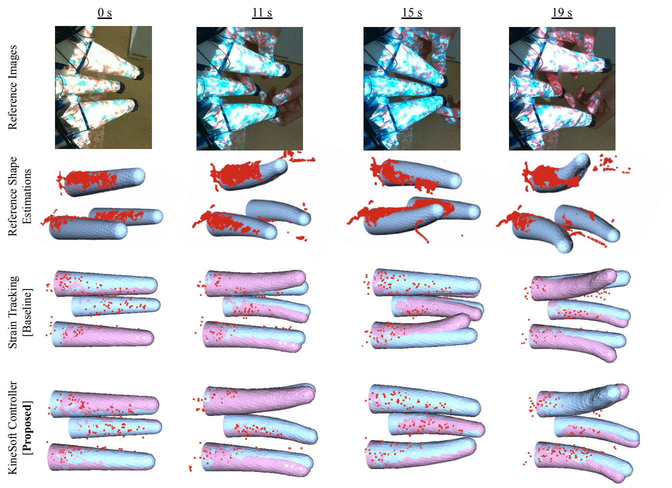
0 s
11 s
15 s
19 s
Reference Images
Reference Shape Estimations
Strain Tracking [Baseline]
KineSoft Controller [Proposed]

Fig. 7: **Trajectory tracking performance evaluation.** We provide each of the controllers with kinesthetically deformed shape trajectories.

TABLE I: Shape Tracking Error Comparison

| Method | Representation | Error (mm) |
|---|---|---|
| Strain-tracking [6, 30] | Strain | $6.20 \pm 2.39$ |
| **KineSoft** | Mesh | $\mathbf{3.29 \pm 0.91}$ |

### *B. Shape Estimation Fidelity*

We evaluated the proposed shape estimation model against two baseline methods from the literature: constant curvature model [9, 45], a common analytical approach for soft robot shape representation, and DeepSoRo [38], a learning-based point cloud reconstruction method. Additionally, we demonstrate that our soft body mechanics-based mapping between observed resistance values and simulated strains, as outlined in Section IV-B, substantially improves upon a naively calibrated linear mapping while using the same domain alignment approach proposed in Section IV-C.

We performed evaluations using observed point clouds generated by the physical setup shown in Figure 3, using the unidirectional Chamfer distance metric as outlined in Equation 7. The quantitative results are outlined in Table II and visualized in Figure 6. Our model-based approach achieved a shape estimation error of 1.92 mm, representing a 41.3%

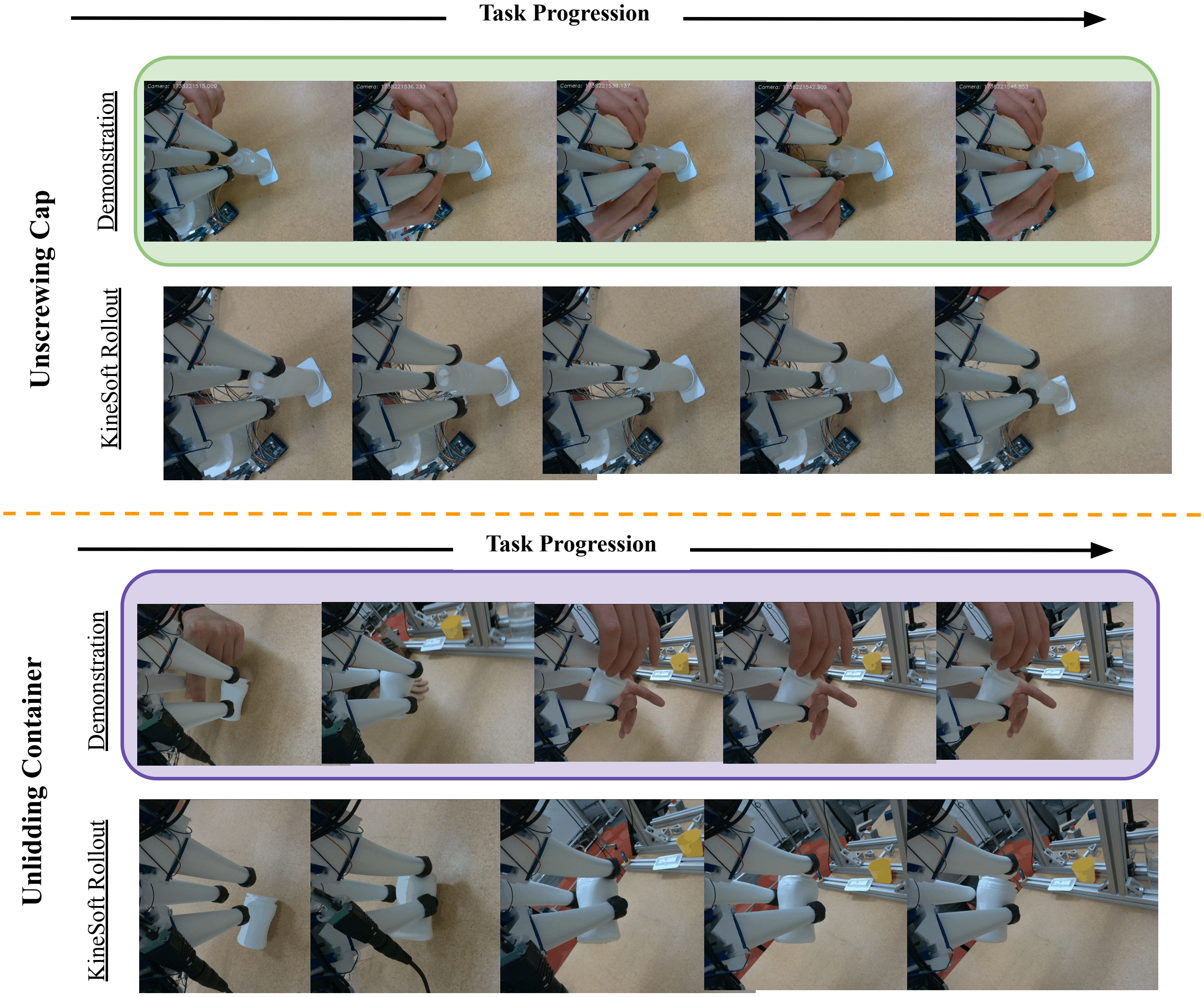

Fig. 8: **Tasks and KineSoft Rollout.**

improvement over the best baseline method (DeepSoRo) and a 60.9% improvement over the linear variant of our approach.

TABLE II: Shape Estimation Fidelity

| Method | Input Signals | Representation | Shape Error [mm] |
|---|---|---|---|
| PneuFlex Sensor | Strain | curvature | $3.70 \pm 1.36$ |
| DeepSoRo | Strain | point cloud | $3.27 \pm 1.05$ |
| **KineSoft** | Strain (naive) | mesh | $4.91 \pm 2.85$ |
| **KineSoft** | Strain (model) | mesh | $\mathbf{1.92 \pm 0.39}$ |

### C. *Shape-conditioned Controller Performance*

We evaluated our shape-conditioned controller against a strain-tracking baseline that directly uses sensor readings for control, as commonly implemented in prior soft robot manipulation works [6, 30]. For evaluation, we collected reference trajectories through kinesthetic teaching, where a demonstrator physically deformed the fingers into desired configurations. During execution, the controllers had to track these trajectories using tendon actuation.

This evaluation specifically highlights a fundamental challenge in soft robot imitation: the sensor signals generated during kinesthetic demonstration (when fingers are manually deformed) differ significantly from those produced during autonomous execution (when tendons are actuated). The strain-tracking baseline, which attempts to directly match these sensor readings, struggles with this demonstration-execution gap, achieving 6.20 mm tracking error. In contrast, our shape-conditioned controller bridges this gap by tracking the geometric shape itself rather than the underlying sensor signals, achieving 3.29 mm error, which is a 47% improvement.

Figure 5 visualizes this performance difference. While both controllers receive reference trajectories collected through kinesthetic teaching, the shape-conditioned controller successfully reproduces the demonstrated shapes using tendon actuation, even though the sensor signals during execution differ from those during the demonstration. The strain-tracking baseline exhibits larger errors because it attempts to match sensor readings that are different between demonstration and execution modes.

The improved shape tracking performance of KineSoft's shape-conditioned controller stems from its ability to abstract away the differences in how deformations are achieved (manual guidance vs. tendon actuation) by focusing on the resulting geometric configurations. This shape-based abstraction provides a consistent representation across demonstration and execution, enabling functionally successful reproduction of demonstrated skills.

### D. In-hand Manipulation Task Performance

**In-hand Manipulation Tasks.** We evaluated KineSoft on in-hand manipulation tasks requiring precise finger control and in-hand dexterity:

- ***Cap Unscrewing.*** MOE makes contact with a bottle cap that is screwed on. Then it must coordinate the three fingers to rotate the cap sufficiently to unscrew it then raise it up. The task is considered successful if the initially screwed on container is detached from the bottle.
- ***Container Unlidding.*** MOE initially grasps an empty container that has a closed hinged snap top lid. It must then lift up the container then move the fingers to the right positions and flip the lid open. To apply sufficient force on the lid, two of the MOE fingers must hold the container's body while one of the fingers pries open the lid. The task is considered successful if the container lid is open in MOE's grasp.

For each task, we collected 50 successful demonstrations through kinesthetic teaching, where an expert demonstrator physically guided the fingers through the required motions. This kinesthetic teaching approach highlights a key advantage of our framework: while humans can intuitively demonstrate complex manipulations by directly deforming the soft fingers, these demonstrated configurations must ultimately be achieved through tendon actuation during autonomous execution. This creates a gap, as the sensor signals during human demonstration (direct finger deformation) differ substantially from those during autonomous execution (tendon actuation). KineSoft bridges this gap by focusing on reproducing the demonstrated shapes rather than matching the sensor signals directly.

As shown in Table III, KineSoft achieved a 100% success rate (5/5 trials) on the unscrewing task compared to 0/5 for the strain-matching baseline. The baseline policy failed primarily because it attempted to reproduce sensor signals from demonstration that were either impossible to achieve through tendon actuation alone or not grounded to the functionally correct robot states to perform the task. In contrast, KineSoft's shape-based approach successfully translated the demonstrated trajectories into executable tendon commands that produced the desired finger configurations.

These results demonstrate that by focusing on tracking demonstrated shapes rather than raw sensor signals, KineSoft effectively bridges the gap between human demonstration and autonomous execution. The shape estimation model provides a consistent representation across both modes, while the shape-conditioned controller reliably reproduces the demonstrated behaviors despite the gap in how deformations are achieved.

TABLE III: Task Performance Across Different Policies

| Task | Policy | State Representation | Success Rate |
|---|---|---|---|
| Cap Unscrewing | Strain-Matching | Strain | 0/5 |
| | **KineSoft** | Mesh | **5/5** |
| Container Unlidding | Strain-Matching | Strain | 0/5 |
| | **KineSoft** | Mesh | **4/5** |

## VI. Limitations

While KineSoft demonstrates effective shape estimation and trajectory tracking for soft robotic manipulation, several limitations remain. First, the current system lacks explicit force feedback mechanisms, relying instead on kinematic trajectories and deformation states to encode manipulation skills. This limits precise force control during object interactions.

Additionally, although kinesthetic teaching provides an intuitive approach to demonstrating some behaviors, it requires the expert demonstrator to recognize the MOE finger's workspace limitations to avoid demonstrating shapes that the MOE cannot reach. In our future work, we aim to leverage KineSoft's real-time shape estimation to communicate robot affordances to users via a visual user interface during demonstrations.

Although the sensors used in this work are readily available and low-cost, they introduce fabrication complexity when integrating into the prior MOE design, which was relatively simple to fabricate. We also observed that the sensors may not respond dynamically in fast manipulation tasks due to commonly observed soft body phenomena such as creep and hysteresis, where the sensor signals require some time to reach equilibrium conditions when stretched quickly.

The framework also requires careful calibration of the strain sensors and domain alignment parameters to achieve reliable shape estimation. To this end, the proposed domain alignment method provides an algorithmic approach to self-calibrate and align the real-world sensor readings. However, this process still requires a controlled test environment, as shown in Figure 3, which may limit reproducibility and accessibility.

KineSoft's shape-conditioned low-level controller weighs the residuals from vertices of the reference meshes equally. Implicitly, the controller prioritizes the vertices closer to the tip higher because the displacement tends to be greater further away from the base. For many tasks, this is an appropriate weighing as the finger tips play a vital role in performing dexterous in-hand manipulation tasks [7]. However, different tasks may require different regions of the fingers to be prioritized based on contact probability, for instance.

## VII. Conclusion and Lessons

This paper presents KineSoft, a framework for learning dexterous manipulation skills with soft robot hands that embraces, rather than fights against, their inherent compliance. The key insight of our work is recognizing that while this compliance enables intuitive kinesthetic teaching, it creates a fundamental gap between demonstration and execution, where the deformations and sensor signals during human demonstration differ substantially from those during autonomous execution through tendon actuation.

KineSoft addresses this challenge through a hierarchical approach. A shape estimation model provides consistent geometric representations across demonstration and execution modes while a domain alignment method enables robust transfer of simulation-trained models to real hardware and a shape-conditioned controller reliably tracks the policy's generated deformation trajectories despite the different underlying actuation mechanisms. Then, a high-level imitation policy learns to generate target vertex deformations from demonstrations, capturing the intended manipulation strategy in geometric grounding. Our experimental results demonstrate that this shape-based hierarchical approach enables more effective skill transfer than methods that attempt to directly match sensor signals or joint configurations. This work suggests that successful imitation learning for dexterous soft robots requires careful consideration of how demonstration and execution modes differ, and appropriate intermediate representations to bridge this gap.

## References

[1] Sylvain Abondance, Clark B Teeple, and Robert J Wood. A dexterous soft robotic hand for delicate in-hand manipulation. *IEEE Robotics and Automation Letters*, 5(4): 5502–5509, 2020.

[2] OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39 (1):3–20, 2020.

[3] Sridhar Pandian Arunachalam, Irmak Güzey, Soumith Chintala, and Lerrel Pinto. Holo-dex: Teaching dexterity with immersive mixed reality. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5962–5969. IEEE, 2023.

[4] Sridhar Pandian Arunachalam, Sneha Silwal, Ben Evans, and Lerrel Pinto. Dexterous imitation made easy: A learning-based framework for efficient dexterous manipulation. In *2023 ieee international conference on robotics and automation (icra)*, pages 5954–5961. IEEE, 2023.

[5] Sarthak Bhagat, Hritwick Banerjee, Zion Tsz Ho Tse, and Hongliang Ren. Deep reinforcement learning for soft, flexible robots: Brief review with impending challenges. *Robotics*, 8(1):4, 2019.

[6] Aditya Bhatt, Adrian Sieler, Steffen Puhlmann, and Oliver Brock. Surprisingly robust in-hand manipulation: An empirical study. *arXiv preprint arXiv:2201.11503*, 2022.

[7] Ian M Bullock, Thomas Feix, and Aaron M Dollar. Analyzing human fingertip usage in dexterous precision manipulation: Implications for robotic finger design. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1622–1628. IEEE, 2014.

[8] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023.

[9] Cosimo Della Santina, Antonio Bicchi, and Daniela Rus. On an improved state parametrization for soft robots with piecewise constant curvature and its use in model based control. *IEEE Robotics and Automation Letters*, 5(2): 1001–1008, 2020.

[10] Cosimo Della Santina, Christian Duriez, and Daniela Rus. Model-based control of soft robots: A survey of the state of the art and open challenges. *IEEE Control Systems Magazine*, 43(3):30–65, 2023.

[11] Charlotte Firth, Kate Dunn, M Hank Haeusler, and Yi Sun. Anthropomorphic soft robotic end-effector for use with collaborative robots in the construction industry. *Automation in Construction*, 138:104218, 2022.

[12] Zipeng Fu, Tony Z Zhao, and Chelsea Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. *arXiv preprint arXiv:2401.02117*, 2024.

[13] Abhishek Gupta, Clemens Eppner, Sergey Levine, and Pieter Abbeel. Learning dexterous manipulation for a soft robotic hand from human demonstrations. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3786–3793. IEEE, 2016.

[14] Bianca S Homberg, Robert K Katzschmann, Mehmet R Dogar, and Daniela Rus. Robust proprioceptive grasping with a soft robot hand. *Autonomous robots*, 43:681–696, 2019.

[15] Yafei Hu, Quanting Xie, Vidhi Jain, Jonathan Francis, Jay Patrikar, Nikhil Keetha, Seungchan Kim, Yaqi Xie, Tianyi Zhang, Hao-Shu Fang, et al. Toward general-purpose robots via foundation models: A survey and meta-analysis. *arXiv preprint arXiv:2312.08782*, 2023.

[16] Edward Johns. Coarse-to-fine imitation learning: Robot manipulation from a single demonstration. In *2021 IEEE international conference on robotics and automation (ICRA)*, pages 4613–4619. IEEE, 2021.

[17] Robert K Katzschmann, Cosimo Della Santina, Yasunori Toshimitsu, Antonio Bicchi, and Daniela Rus. Dynamic motion control of multi-segment soft robots using piecewise constant curvature matched with an augmented rigid body model. In *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)*, pages 454–461. IEEE, 2019.

[18] Fukang Liu, Kavya Puthuveetil, Akhil Padmanabha, Karan Khokar, Zeynep Temel, and Zackory Erickson. Skingrip: An adaptive soft robotic manipulator with capacitive sensing for whole-limb bathing assistance. *arXiv preprint arXiv:2405.02772*, 2024.

[19] Yang Liu, Seong Hyo Ahn, Uksang Yoo, Alexander R Cohen, and Farshid Alambeigi. Toward analytical modeling and evaluation of curvature-dependent distributed

friction force in tendon-driven continuum manipulators. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8823–8828. IEEE, 2020.

[20] Yang Liu, Uksang Yoo, Seungbeom Ha, S Farokh Atashzar, and Farshid Alambeigi. Influence of antagonistic tensions on distributed friction forces of multisegment tendon-driven continuum manipulators with irregular geometry. *IEEE/ASME Transactions on Mechatronics*, 27 (5):2418–2428, 2021.

[21] Matthew T Mason and J Kenneth Salisbury Jr. Robot hands and the mechanics of manipulation. *The MIT Press, Cambridge, MA*, 1985.

[22] Marius Memmel, Jacob Berg, Bingqing Chen, Abhishek Gupta, and Jonathan Francis. Strap: Robot sub-trajectory retrieval for augmented policy learning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=4VHiptx7xe.

[23] Amir Pagoli, Frédéric Chapelle, Juan Antonio Corrales, Youcef Mezouar, and Yuri Lapusta. A soft robotic gripper with an active palm and reconfigurable fingers for fully dexterous in-hand manipulation. *IEEE Robotics and Automation Letters*, 6(4):7706–7713, 2021.

[24] Tessa J Pannen, Steffen Puhlmann, and Oliver Brock. A low-cost, easy-to-manufacture, flexible, multi-taxel tactile sensor and its application to in-hand object recognition. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 10939–10944. IEEE, 2022.

[25] Steffen Puhlmann, Jason Harris, and Oliver Brock. Rbo hand 3: A platform for soft dexterous manipulation. *IEEE Transactions on Robotics*, 38(6):3434–3449, 2022.

[26] Haozhi Qi, Brent Yi, Mike Lambeta, Yi Ma, Roberto Calandra, and Jitendra Malik. From simple to complex skills: The case of in-hand object reorientation. *arXiv preprint arXiv:2501.05439*, 2025.

[27] Yuzhe Qin, Wei Yang, Binghao Huang, Karl Van Wyk, Hao Su, Xiaolong Wang, Yu-Wei Chao, and Dieter Fox. Anyteleop: A general vision-based dexterous robot arm-hand teleoperation system. *arXiv preprint arXiv:2307.04577*, 2023.

[28] Daniela Rus and Michael T Tolley. Design, fabrication and control of soft robots. *Nature*, 521(7553):467–475, 2015.

[29] Pierre Schegg, Etienne Ménager, Elie Khairallah, Damien Marchal, Jérémie Dequidt, Philippe Preux, and Christian Duriez. Sofagym: An open platform for reinforcement learning based on soft robot simulations. *Soft Robotics*, 10(2):410–430, 2023.

[30] Adrian Sieler and Oliver Brock. Dexterous soft hands linearize feedback-control for in-hand manipulation. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8757–8764. IEEE, 2023.

[31] O Starkova and A Aniskevich. Poisson's ratio and the incompressibility relation for various strain measures with the example of a silica-filled sbr rubber in uniaxial tension tests. *Polymer Testing*, 29(3):310–318, 2010.

[32] Francesco Stella, Qinghua Guan, Cosimo Della Santina, and Josie Hughes. Piecewise affine curvature model: a reduced-order model for soft robot-environment interaction beyond pcc. In *2023 IEEE International Conference on Soft Robotics (RoboSoft)*, pages 1–7. IEEE, 2023.

[33] Javier Tapia, Espen Knoop, Mojmir Mutnỳ, Miguel A Otaduy, and Moritz Bächer. Makesense: Automated sensor design for proprioceptive soft robots. *Soft robotics*, 7(3):332–345, 2020.

[34] Thomas George Thuruthel, Egidio Falotico, Federico Renda, and Cecilia Laschi. Model-based reinforcement learning for closed-loop dynamic control of soft robotic manipulators. *IEEE Transactions on Robotics*, 35(1): 124–134, 2018.

[35] Vincent Wall, Gabriel Zöller, and Oliver Brock. A method for sensorizing soft actuators and its application to the rbo hand 2. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4965–4970. IEEE, 2017.

[36] Jun Wang, Ying Yuan, Haichuan Che, Haozhi Qi, Yi Ma, Jitendra Malik, and Xiaolong Wang. Lessons from learning to spin "pens". In *CoRL*, 2024.

[37] Liangliang Wang, James Lam, Xiaojiao Chen, Jing Li, Runzhi Zhang, Yinyin Su, and Zheng Wang. Soft robot proprioception using unified soft body encoding and recurrent neural network. *Soft Robotics*, 10(4):825–837, 2023.

[38] Ruoyu Wang, Shiheng Wang, Songyu Du, Erdong Xiao, Wenzhen Yuan, and Chen Feng. Real-time soft body 3d proprioception via deep vision-based sensing. *IEEE Robotics and Automation Letters*, 5(2):3382–3389, 2020.

[39] Dehao Wei and Huazhe Xu. A wearable robotic hand for hand-over-hand imitation learning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 18113–18119. IEEE, 2024.

[40] Abraham Itzhak Weinberg, Alon Shirizly, Osher Azulay, and Avishai Sintov. Survey of learning-based approaches for robotic in-hand manipulation. *Frontiers in Robotics and AI*, 11:1455431, 2024.

[41] JD Westwood et al. Sofa–an open source framework for medical simulation. *Medicine Meets Virtual Reality 15: In Vivo, in Vitro, in Silico: Designing the Next in Medicine*, 125:13, 2007.

[42] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 206–215, 2018.

[43] Yunchao Yao, Uksang Yoo, Jean Oh, Christopher G Atkeson, and Jeffrey Ichnowski. Soft robotic dynamic in-hand pen spinning. *arXiv preprint arXiv:2411.12734*, 2024.

[44] Oncay Yasa, Yasunori Toshimitsu, Mike Y Michelis, Lewis S Jones, Miriam Filippi, Thomas Buchner, and

Robert K Katzschmann. An overview of soft robotics. *Annual Review of Control, Robotics, and Autonomous Systems*, 6(1):1–29, 2023.

[45] Uksang Yoo, Yang Liu, Ashish D Deshpande, and Farshid Alamabeigi. Analytical design of a pneumatic elastomer robot with deterministically adjusted stiffness. *IEEE robotics and automation letters*, 6(4):7773–7780, 2021.

[46] Uksang Yoo, Hanwen Zhao, Alvaro Altamirano, Wenzhen Yuan, and Chen Feng. Toward zero-shot sim-to-real transfer learning for pneumatic soft robot 3d proprioceptive sensing. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 544–551. IEEE, 2023.

[47] Uksang Yoo, Adam Hung, Jonathan Francis, Jean Oh, and Jeffrey Ichnowski. Ropotter: Toward robotic pottery and deformable object manipulation with structural priors. In *2024 IEEE-RAS 23rd International Conference on Humanoid Robots (Humanoids)*, pages 843–850. IEEE, 2024.

[48] Uksang Yoo, Ziven Lopez, Jeffrey Ichnowski, and Jean Oh. Poe: Acoustic soft robotic proprioception for omnidirectional end-effectors. *arXiv preprint arXiv:2401.09382*, 2024.

[49] Uksang Yoo, Nathaniel Dennler, Eliot Xing, Maja Matarić, Stefanos Nikolaidis, Jeffrey Ichnowski, and Jean Oh. Soft and compliant contact-rich hair manipulation and care. *arXiv preprint arXiv:2501.02630*, 2025.

[50] Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy. *arXiv preprint arXiv:2403.03954*, 2024.

[51] Annan Zhang, Tsun-Hsuan Wang, Ryan L Truby, Lillian Chin, and Daniela Rus. Machine learning best practices for soft robot proprioception. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2564–2571. IEEE, 2023.

[52] Shuai Zhou, Yuanhang Li, Qianqian Wang, and Zhiyang Lyu. Integrated actuation and sensing: Toward intelligent soft robots. *Cyborg and Bionic Systems*, 5:0105, 2024.

[53] Xiaolong Zhou, Xiaoting Chen, and Tianmiao Yang. Soft robotic grippers. *Advanced Intelligent Systems*, 5(1): 2000198, 2023.

# Appendix A
# Sensor Signals

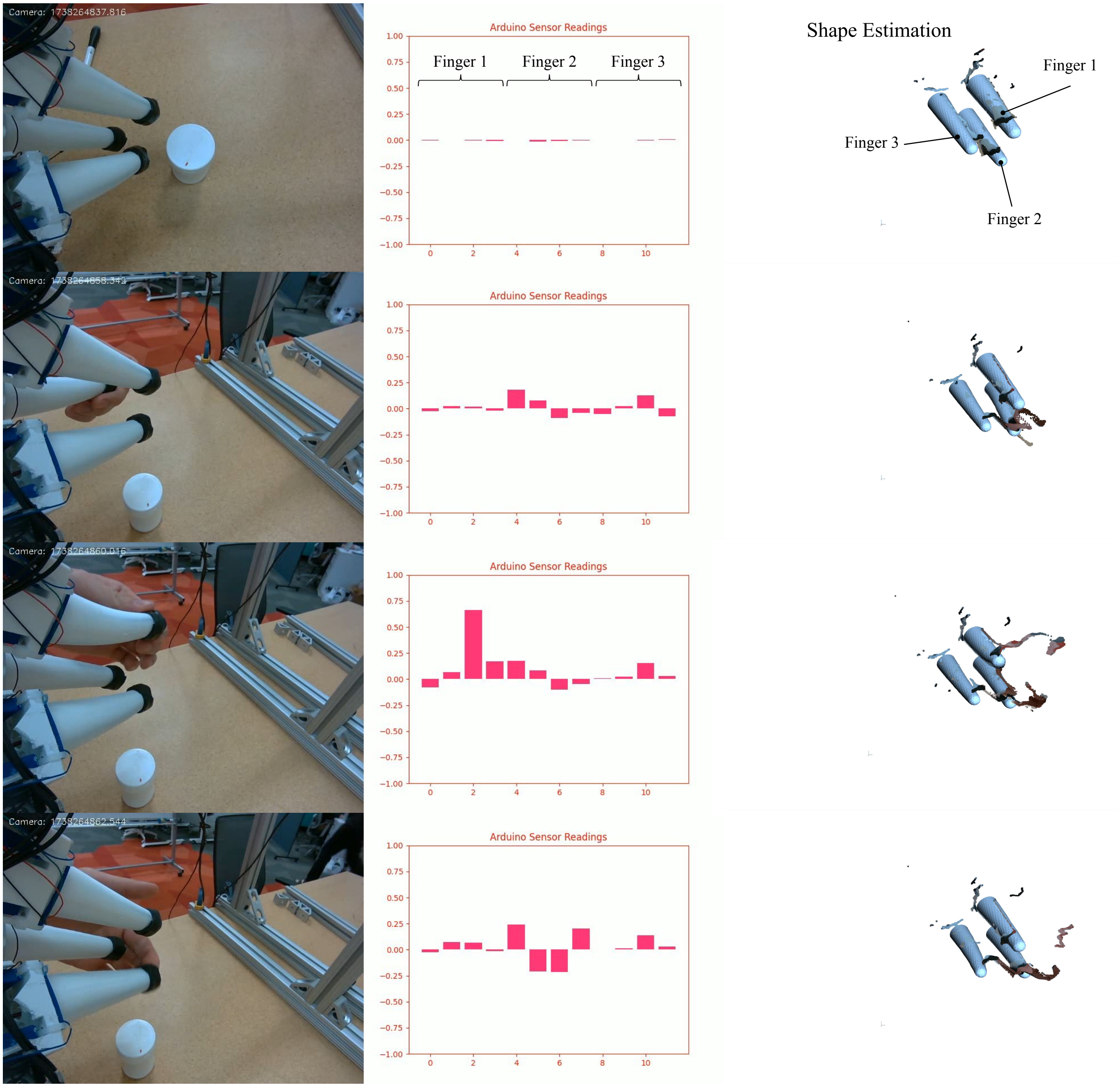

Fig. 9: **Sensor signals and corresponding shape estimation**

## Appendix B
## Other Task

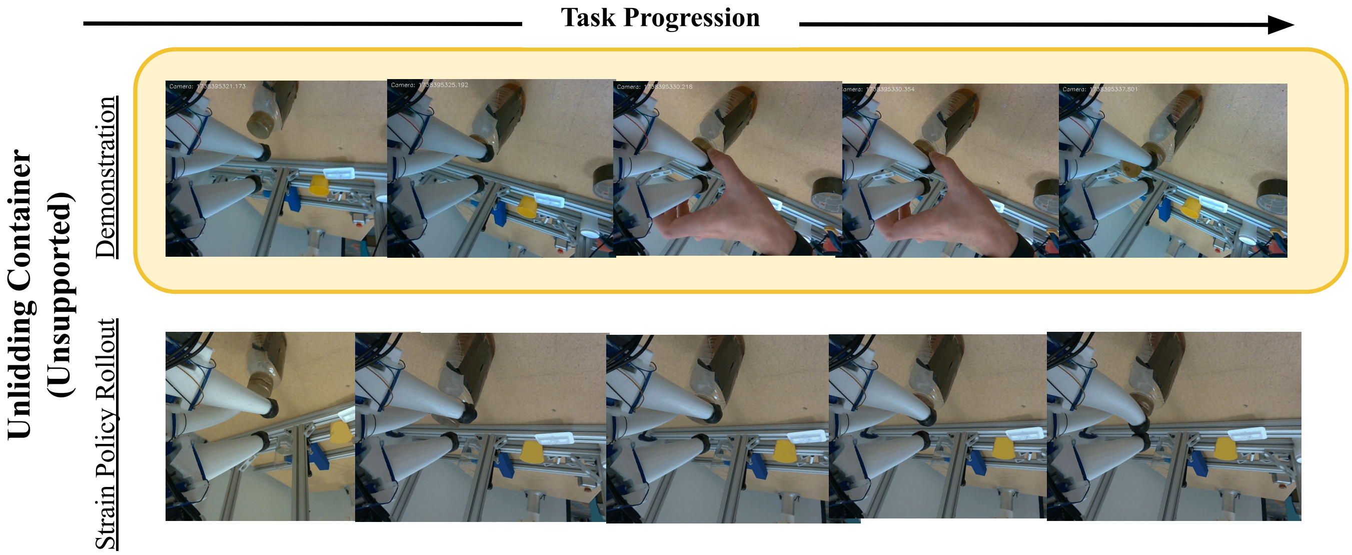

Fig. 10: **Strain tracking performance on a simple task**

Strain tracking can succeed in some tasks. Particularly with a weighted bottle opening task, where one finger could flick the lid open, the strain matching policy seemed to perform consistently.