Reactive Human-Robot Collaborative Manipulation of Deformable Linear Objects Using a New Topological Latent Control Model

Peng Zhou[a,c,*], Pai Zheng[b], Jiaming Qi[a], Chengxi Li[b], Hoi-Yin Lee[a], Anqing Duan[a], Liang Lu[c], Zhongxuan Li[c], Luyin Hu[c], David Navarro-Alarcon[a]

[a]*Department of Mechanical Engineering, The Hong Kong Polytechnic University, KLN, Hong Kong,*
[b]*Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, KLN, Hong Kong,*
[c]*Centre for Transformative Garment Production, The Hong Kong University, NT, Hong Kong,*

Abstract

Real-time reactive manipulation of deformable linear objects is a challenging task that requires robots to quickly and adaptively respond to changes in the object's deformed shape that result from external forces. In this paper, a novel approach is proposed for real-time reactive deformable linear object manipulation in the context of human-robot collaboration. The proposed approach combines a topological latent representation and a fixed-time sliding mode controller to enable seamless interaction between humans and robots. The introduced topological control model offers a framework for controlling the dynamic shape of deformable objects. By leveraging the topological representation, our approach captures the connectivity and structure of the objects' shapes within a latent space. This enables improved generalization and performance when handling complex deformable shapes. A fixed-time sliding mode controller ensures that the object is manipulated in real-time, while also ensuring that it remains accurate and stable during the manipulation process. To validate our proposed framework, we first conduct motor-robot experiments to simulate fixed human interaction processes, enabling straightforward comparisons with other approaches. We then follow up with human-robot experiments to demonstrate the effectiveness of our approach.

Keywords: Deformable Linear Objects; Reactive Manipulation; Latent Control Model; Human-Robot Collaboration.

1. Introduction

In recent years, there has been significant progress in Human-Robot Collaboration (HRC) [1, 2, 3, 4, 5] research and development, with the aim of achieving more efficient and effective collaboration between humans and robots in various domains, such as manufacturing [6], construction [7, 8], healthcare [9], and service industries [10]. One area of HRC that has received a lot of attention is the manipulation of objects [11, 5]. In particular, robots are increasingly being designed and developed to manipulate objects in various environments and conditions. However, most of the research in this area has focused on rigid objects, whose unchangeable geometry makes them easier to manipulate than deformable objects.

Deformable objects, on the other hand, can change their shape/configuration under the action of external forces, e.g., coming from a robotic gripper. Examples of deformable objects include fabrics [12], wires [13], cables [14], and soft tissues [15]. Manipulating deformable objects is more challenging than manipulating rigid objects because they have complex and nonlinear behaviors [16, 17, 18, 19]. Despite these challenges, there are many potential applications [20, 21, 22, 23] of deformable object manipulation for HRC in the manufacturing industry, such as fabric handling and sewing, food processing, assembly of flexible parts and so on. In addition to their complex dynamics, the manipulation of deformable objects presents several difficulties in the context of human-robot collaboration due to the unpredictability in the human's manipulation actions. [24, 25].

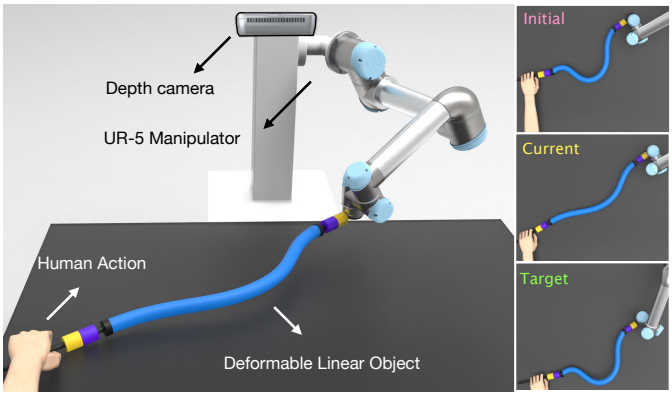
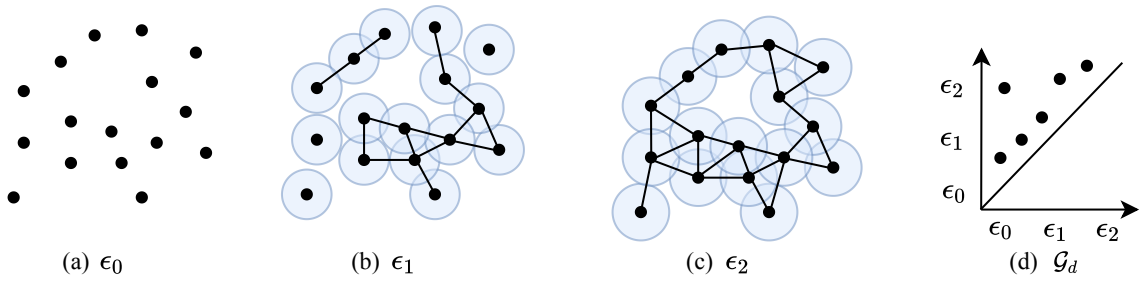


Fig. 1: Conceptual representation for a reactive deformable linear object manipulation in the context of human-robot collaboration, where the robot is adaptively deforming the linear object into the *initial* shape in response to the human partner's action in real-time.

The manipulation of deformable objects in the context of HRC (see Fig. 1 for a conceptual representation) has not been

*This work was supported by the Research Grants Council of the Hong Kong under Grant 15212721 and 15210222, and in part by the Hong Kong Polytechnic University under Grant UANS and G-UAMS.
*Corresponding author e-mail: jeffery.zhou@connect.polyu.hk (P. Zhou).



(a) ϵ_0 (b) ϵ_1 (c) ϵ_2 (d) $\mathcal{G}_d$

Fig. 2: (a)-(c) show at varying scales ϵ_0, ϵ_1, and ϵ_2, the Vietoris-Rips complex $\mathcal{V}_\epsilon(\boldsymbol{q})$ of a point cloud $\boldsymbol{q}$ changes its connectivity as the distance threshold ϵ is increased. (d) presents the d-th persistence diagram $\mathcal{G}_d$ captures the emergence and disappearance of d-dimensional topological features.

sufficiently studied; Most of the existing HRC research has focused on rigid object manipulation. Furthermore, existing approaches [26, 27] for the manipulation of deformable objects typically rely on analytical or numerical models [28, 29] that describe the object's dynamics and behavior. These models are often computationally expensive and may not accurately capture the complex nature of soft bodies. Additionally, these models may not be suitable for real-time control, which is essential for human-robot collaboration [29]. Therefore, this gap is a significant challenge that needs to be addressed to fully realize the potential of Deformable object manipulation (DOM) in various HRC applications [30].

To address these challenges, we propose a novel approach for real-time reactive deformable linear object manipulation in the context of human-robot collaboration. Our approach is named the topological latent control model, and it combines a topological latent representation and a fixed-time sliding controller to enable seamless interaction between humans and robots. The topological latent representation provides a generic approach for applying persistent homology to calculate topological signatures for both the original shape space and latent shape space to derive a topological loss term used for training an auto-encoder network. By leveraging this topological latent representation, our approach is able to capture the connectivity and structure of the objects' shapes within a latent space. This enables improved generalization and performance when handling complex deformable shapes. Besides, the application of a fixed-time sliding controller ensures that the object is manipulated in real-time, while also ensuring that it remains stable and safe during the manipulation process. Therefore, the proposed topological latent control model provides a framework for controlling the shape and motion of deformable objects based on their topological properties. This approach is highly efficient and computationally inexpensive, as it avoids the need for complex analytical or numerical models. In general, our research takes a further step in that direction by weaving computational topology principles into the controller design for human-robot collaboration tasks involving the manipulation of deformable linear objects.

Our proposed method is highly effective, as it provides a robust and reliable control strategy that can adapt to a wide range of deformable objects. To demonstrate the effectiveness of our method, we test it on a variety of deformable linear objects in the context of HRC. Our experiments show that our approach is effective in conducting the manipulation task, enabling seamless interaction between humans and robots in a wide range of real-world experiments. A video of the conducted experiments can be obtained from `https://sites.google.com/view/hrc-dom`. This paper provides a valuable contribution to the field of human-robot collaboration, offering a new approach for real-time reactive deformable linear object manipulation that is both effective and safe.

In summary, we present four key contributions in this paper:

- A novel method for real-time human-robot collaboration that allows the robot to adjust its actions in real time based on the behavior of the deformable object.
- A latent representation embedding topological structure to ensure an efficient and effective control for deformable linear objects.
- A controller that takes as input the latent topological features to support real-time human-robot collaboration during deformable object manipulation tasks.
- A detailed experimental validation of the proposed framework in which robot and unmodelled human partners collaborate to manipulate a deformable linear object.

The rest of the paper is organized as follows: Sect. 2 provides a detailed overview of related work in the field. Sect. 3 gives preliminaries of persistent homology used for constructing topological autoencoder. Sect. 4 present a general system description problem definition. Sect. 5 describes the proposed approach in detail, including the topological latent control model, topological latent representation, and fixed-time sliding controller. Sect. 6 presents the experimental results, and Sect. 7 concludes the paper and discusses future research directions.

2. Related Work

Deformable object manipulation (DOM) [16, 17, 18] is an emerging research problem in robotics that involves handling objects that can change their shape, such as cables, fabrics, and bags. DOM poses significant challenges due to the complex dynamics of the object and the real-time requirement for the manipulation. Several studies [31, 31, 32] have addressed different aspects of DOM, such as modeling, perception, planning,

and control. However, most of them do not consider a collaborative manipulation scenario with human partners, which can enhance the performance and efficiency of DOM tasks.

Some previous works have explored human-robot collaboration for manipulating deformable objects. For example, [20] proposed a method of collaborative manipulation of a deformable sheet between a person and a robot, where the robot follows the human motion to handle the cloth. However, their method relies on predefined motion primitives and does not account for the feedback from the object deformation. Other works [19, 33, 34, 35] have focused on learning-based approaches for DOM, such as DeformableRavens [36], which uses reinforcement learning to train a robot to manipulate cables, fabrics, and bags towards desired goal configurations. However, these approaches do not explicitly model the topological properties of the deformable object, such as knots and folds, which are crucial for some DOM tasks. Our work takes a further step in that direction by introducing concepts from computational topology into the controller design for deformable linear object manipulation tasks.

[20] considered collaborative manipulation of a deformable sheet between a person and a dual-armed robot. The proposed approach was capable of sensing contact force to maintain the tension of the sheet, and in turn comply with human motion, which is akin to handling a tablecloth with a partner but with one's eyes closed. [37] presented a model-based closed-loop control framework for seamless human-robot or multi-robot fabric co-manipulation. A mass-spring model is used for simulating ply distortion and generating optimal grasping points' spatial localization. The model is enhanced with real-time operator handling actions, as captured from the implemented perception system. The proposed sensor and model-based controlling framework incorporate robot motion planners either for operator support, through non-rigid object co-manipulation, or synchronization of cooperative robots within fully automated tasks. [38] presented a model-based motion planner for deformable object co-manipulation and the developed closed-loop controlling framework interprets manipulation inputs into appropriate handling actions by simulating fabric's distortion through a mass-spring mode. [39] presented the collaborative manipulation of rigid objects with deformable objects by introducing a novel framework comprising an Action Prediction Network (APN) and a Configuration Prediction Network (CPN) to model the complex pattern and stochasticity of soft-rigid body systems. Finally, they demonstrated the effectiveness of moving rigid objects to a target position with ropes connected to robotic arms.

In this paper, we introduce an innovative method for reactive manipulation of deformable linear objects (DLOs) within the context of human-robot collaboration. This method is capable of handling complex deformable linear objects and has the ability to react and adapt to the actions of the human partner in real time. Our approach leverages a topological latent space to capture the deformation state of the object and to generate appropriate control actions for the robot. This method provides a unique advantage over existing methods by offering a more comprehensive representation of DLOs and enabling more precise control over their manipulation. Through several collaborative DLO manipulation tasks, we evaluate the efficacy of our proposed method, showcasing its superiority over current methods in terms of deformation error, task response time, and robustness to human interventions. To our knowledge, this is the first work that integrates topological modeling with a latent control model for collaborative DLO manipulation. This pioneering approach opens new possibilities for more effective and adaptive human-robot collaboration in handling deformable linear objects.

3. Preliminaries

In computational topology, the method used for analyzing topological features of data across multiple scales is called *persistent homology* [40, 41]. To obtain the persistent homology of a space, it must first be represented as a *simplicial complex*, which seeks to generate a family of groups by using matrix reduction algorithms. These groups are called the homology groups denoted by $\mathcal{K}$, where d-dimensional topological features comprise the d-th homology group $\mathcal{H}_d(\mathcal{K})$ Typically, homology groups are summarized according to their ranks to obtain an invariant "signature" of the data manifold $\mathcal{M}$. Given an unknown manifold $\mathcal{M}$ over a point cloud $Q = \{q_1, \ldots, q_n\} \subseteq \mathbb{R}^3$ and a distance metric: $Q \times Q \to \mathbb{R}$ (i.e., the Euclidean distance), to keep track of changes in the homology groups across various scales of the metric, persistent homology employs the construction of a unique simplicial complex known as the Vietoris-Rips complex [42]. Let $\mathcal{V}\epsilon(Q)$ be the Vietoris-Rips complex of Q with a scale ϵ, and it has all simplices of the point cloud Q whose elements satisfy a distance criterion, namely $\text{dist}\left(e_i, e_j\right) \leq \epsilon$ for all i, j. As the Vietoris-Rips complex provides a nesting structure, $\mathcal{V}\epsilon_i(Q) \subseteq \mathcal{V}_{\epsilon_j}(Q)$ when $\epsilon_i \leq \epsilon_j$, it becomes possible to trace alterations in the homology groups when ϵ increases [43] (see Fig. 2 for a detailed illustration of this process).

Let $\mathcal{PH}\left(\mathcal{V}_\epsilon(Q)\right)$ represent the persistent homology of the point cloud Q's Vietoris-Rips complex, and it results in a tuple $(\{\mathcal{G}_1, \mathcal{G}_2, \ldots\}, \{\phi_1, \phi_2, \ldots\})$ $\mathcal{G}_i$ and ϕ_i denote the persistence diagrams and persistence pairings, respectively. In d-dimensional persistence diagram $\mathcal{G}_d$, we define a tuple of (a, b), where a denotes a scale ϵ at which a d-dimensional topological feature emerges, and b represents another scale ϵ' at which it disappears. The d-dimensional persistence consists of pairs of indices denoted as (i, j) that correspond to simplices $s_i, s_j \in \mathcal{V}\epsilon(Q)$ responsible for generating and annihilating topological features characterized by $(a, b) \in \mathcal{G}_d$, respectively. To compare the diagram $\mathcal{G}$ and $\mathcal{G}'$, we can use the bottleneck distance defined as: $\mathrm{d_b}\left(\mathcal{G}, \mathcal{G}'\right) := \inf_{\eta:\mathcal{G}\to\mathcal{G}'} \sup_{x\in\mathcal{G}} \|x - \eta(x)\|_\infty$, where $\eta : \mathcal{G} \to \mathcal{G}'$ is defined as a bijection between the diagram $\mathcal{G}$ and $\mathcal{G}'$, and $\|\cdot\|_\infty$ refers to the L_∞ norm. Finally, we define $\mathcal{G}^Q$ as the set of persistence diagrams for the point cloud Q, which can be obtained from the computation of $\mathcal{PH}\left(\mathcal{V}\epsilon(Q)\right)$.

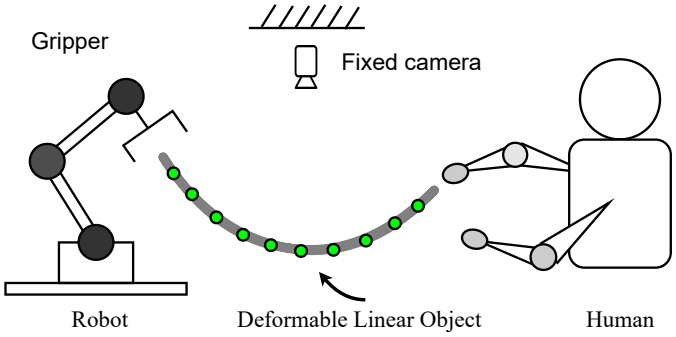


Fig. 3: The illustration of human-robot collaboration for deformable linear object manipulation.

4. Problem Formulation

In this article, we propose a novel human-robot collaborative system consisting of a robot manipulator, a human hand, and a deformable linear object, as depicted in Fig. 3. We assume that the deformable linear object is firmly held by both the robot arm and the human hand and there is no displacement between the manipulated object and the robotic end-effector or the object and the human hand, respectively. The human hand applies force on one side of the deformable linear object and leads to shape deformations (measured by a top-down depth camera), while the control system attempts to generate control commands for the robot arm to reactively recover its original shape. During the entire process, the human partner is leading the deformable object manipulation task by simply moving one side of the deformable linear object, so we define it as a "leader" role. On the other hand, the robot manipulator carries the other side of the manipulated linear object to achieve intelligent and reactive behavior to follow the leader's motion in real-time, which we refer to as "follower" (see Fig. 3 for details).

As shown in Fig. 3, a classic deformable object shape servoing task is reconsidered in the context of human-robot collaboration. Our objective is to develop a model-free reactive vision-based controller to respond to the movements of a human partner on deformable linear objects, without relying on any prior knowledge of the physical characteristics of elastic rods. Throughout the process, the controller instructs the robot to continually deform the linear object to maintain its initial shape in real time. In this task, the human partner leads the deformation first, then the robot controller follows the human action and manipulates the object into the initial shape. Therefore, we defined the human partner as the leader role, and the robot controller as the follower role in this human-robot collaboration.

Assumption 1. *The robot arm and human hand both securely grip the flexible linear object, and there is no motion between the manipulated object and the robot end-effector or between the object and the human hand.*

Consider a 6-degree-of-freedom (DOF) robot with revolute joints, we denote the joint-angle vector as $\boldsymbol{q} \in \mathbb{R}^6$, and the end-effector pose (3-DOF position and 3-DOF orientation) as $\boldsymbol{x} \in \mathbb{R}^6$, respectively. According to the classical kinematic equation of the manipulator, the differential relationship between $\boldsymbol{q}$ and $\boldsymbol{x}$ is given as follows:

$$\dot{\boldsymbol{x}} = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{q}}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{1}$$

where the matrix $\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{q}}(\boldsymbol{q}) \in \mathbb{R}^{6\times 6}$ represents the analytical kinematic equation of the robot. In this paper, the robot is assumed to be controlled with a kinematic interface, i.e., the robot can accurately operate the given velocity commands (e.g., the velocities of joint or end-effector).

Remark 1. *In this work, we present a controller that we have designed with the capacity to adapt to speed control across any given dimensionality. While it is typical for the dimensionality of the end-effector to be less than 6, our controller is not confined to this limitation.*

For the purpose of our experiments, and to provide a clear demonstration of performance, we have streamlined our focus to the utilization of 3D translation. This decision allows us to functionally validate our proposed framework for reactive deformable object manipulation in an efficient yet effective manner. It is important to note that our controller has been engineered with versatility in mind. Therefore, it can comfortably accommodate any rotational requirements that may arise, rendering it capable of operating under more complex conditions if necessary.

Remark 2. *In this paper, the speed control signal of the end-effector is designed, and by using* (1) *the angular joint velocity command of the manipulator can be calculated, accordingly. Note that in real physical experiments, this joint velocity command typically suffers a saturation effect.*

In this paper, a depth camera within an eye-to-hand configuration to observe the shape of the elastic cable. For simplicity, we use the commonly used center keypoints based splines to represent the object's shape, with the following definitions:

$$\boldsymbol{s} = [\, c_1, \ldots, c_N \,]^\top \in \mathbb{R}^{3N} \tag{2}$$

where N denotes the number of total center key points constituting the spline of the linear object, $c_i = [x_i, y_i, z_i] \in \mathbb{R}^3$ is the Cartesian coordinates of the i-th centerline point. Though B-splines provide a powerful tool for representing DLOs, in this study, we have chosen to utilize splines. This choice was made based on the simplicity and suitability of splines for our particular control algorithms, and we found that splines offered a sufficient balance between complexity and performance for our application.

In this study, our focus is on a shape servoing task, where the pose $\boldsymbol{x}$ of the end-effector definitely influences the shape $\mathbf{s}$ of the elastic cable. We assume the material properties of the objects and human movements remain relatively stable throughout the manipulation process. Consequently, within the scope of local deformation [28], the shape of the deformable object can be represented by an unknown nonlinear function:

$$\boldsymbol{s} = f_S(\boldsymbol{x}) \tag{3}$$

Remark 3. *It is important to note that the poses of a human hand or a gripper do not unambiguously define the configuration of a deformable linear object. According to [44], the placements of both ends of the object can result in multiple static equilibrium configurations. This essentially means that the shape of the object is not uniquely determined by the positions of its ends. During the shape servoing tasks we consider, that minor movements by the human operator lead to correspondingly small deformations. This gradual change allows the robot sufficient time to adapt to and manage these local deformations effectively.*

Then the kinematic model of first-order can be obtained by calculating the time derivative of (3), resulting in the following equation:

$$\dot{\boldsymbol{s}} = \frac{\partial f_S}{\partial \boldsymbol{x}} \dot{\boldsymbol{x}} = \boldsymbol{J}_s(\boldsymbol{x}) \dot{\boldsymbol{x}} \tag{4}$$

where $\boldsymbol{J}_s(\boldsymbol{x})$ is the deformation Jacobian matrix (DJM) [15], which describes the kinematic relationship between the robot and original shape feature of the manipulated deformable linear object.

Assumption 2. *The deformation Jacobian matrix (DJM) is able to describe the kinematic relationship between the robot and the manipulated deformable linear object.*

However, the large dimension of the original shape $\boldsymbol{s}$ is $3N$, so it is inefficient to be directly used as the inputs of the controller since not all the dimension of the shape data space is necessary for the controller solving the manipulation tasks and some of the information are redundant during the task. In our approach, we design a feature extraction method to construct a low-dimensional feature vector $\boldsymbol{z} \in \mathbb{R}^k (k \ll 3N)$ to represent $\boldsymbol{s}$, which characterizes the original shape $\boldsymbol{s}$ but with significantly fewer-dimensional feedback vector. Theoretically, the feature $\boldsymbol{z}$ has a one-to-one mapping relationship with $\boldsymbol{s}$, i.e., $\boldsymbol{z} = f_Z(\boldsymbol{s})$. Thus, the latent shape feature $\boldsymbol{z}$ can be obtained as below:

$$\boldsymbol{z} = f_Z(\boldsymbol{s}) = f_Z(f_S(\boldsymbol{x})) \tag{5}$$

The initial kinematic model of first-order can be obtained by calculating the time derivative of (5), resulting in the following equation:

$$\dot{\boldsymbol{z}} = \frac{\partial f_Z}{\partial \boldsymbol{x}} \dot{\boldsymbol{x}} = \boldsymbol{J}_z(\boldsymbol{x}) \dot{\boldsymbol{x}} \tag{6}$$

where $\boldsymbol{J}_z(\boldsymbol{x})$ is the latent deformation Jacobian matrix (LDJM), which describes the kinematic relationship between the robot and the low-dimensional shape feature of the manipulated deformable linear object. As the physical information of the object is usually unknown and difficult to obtain through identifications, the DJM often needs to be estimated numerically. It should be noted that the deformations of the DLO depend solely on its potential energy, the force of contact between the manipulator and the DLO, as well as the force of contact with the human hand. The quasi-static configuration (6) holds when the materials properties of objects and the human motions do not change significantly during the manipulation process, as LDJM $\mathbf{J}_z(\mathbf{x})$ captures the velocity mapping between latent shapes and the robot motions.

Assumption 3. *The DJM can be separated into two parts:* $\mathbf{J}(\boldsymbol{x}) = \hat{\mathbf{J}} + \tilde{\mathbf{J}}$, *where* $\tilde{\mathbf{J}}$ *is the approximation error and* $\hat{\mathbf{J}}$ *is the estimated* $\mathbf{J}(\mathbf{x})$.

Assumption 4. *A bound exists for the approximation error* $\tilde{\mathbf{J}}$, $\|\tilde{\mathbf{J}}\|^2 \leq \eta$, *for* η *as an unknown positive constant.*

5. Methodology

In this section, we propose a novel framework of human-robot collaboration for reactive deformable linear object manipulation as shown in Fig. 4, which is mainly composed of three components, namely, a deformable object state estimator, a topological-aware latent shape space, and a fixed-time sliding model-based controller. The deformable shapes of the manipulated linear object are represented by point cloud data. With the Gaussian Mixture Model, the deformable shapes are perceived as the corresponding centroids along the centerline of the object. Followed by a topological auto-encoder, the deformable centerlines are compressed into a low-dimensional latent space. At last, a fixed-time sliding model-based controller is used to command the robot action to follow the human action for achieving the shape servoing task in a reactive manner.

In this process, we start with an initial deformable linear shape $\boldsymbol{s}_0$, and after a series of human-robot interactions, the current shape becomes $\boldsymbol{s}_i$. Our goal is to command the robot manipulator to apply force to one side of the deformable object and deform it into the desired shape $\boldsymbol{s}_d$. This target shape is regarded as a similar shape to $\boldsymbol{s}_0$. The procedure begins with the state estimator, which is able to represent the original shapes perceived by point clouds $\{\boldsymbol{q}_i\}$ as a set of centerline points $\{\boldsymbol{c}_i\}$. This step is performed at every timestep. Next, a topological-aware auto-encoder $f_h : C \rightarrow \mathcal{Z}$ is trained once in a pre-training phase. This is achieved by combining the reconstruction loss $\mathcal{L}_{rec}$ and topological loss $\mathcal{L}_{topo}$. The auto-encoder is used to encode the centerline-based shapes $\boldsymbol{c}_i$ from the high-dimensional shape space C into a low-dimensional latent shape space $\mathcal{Z}$ during each timestep. With the constructed latent shape space, a deep neural network-based latent shape predictor is used at each timestep to predict the desired latent shape $\boldsymbol{z}_d$. Together with $\boldsymbol{z}_i$, these are regarded as the inputs for the designed fixed-time sliding model to propose robot commands in each timestep. By doing so, the robot agent is able to accomplish the shape servoing task reactively during human-robot collaborations.

5.1. Deformable Linear Object State Estimation

It is crucial to estimate the state when performing reactive manipulations of deformable linear objects during human-robot collaborations. As shown in Fig. 4, depth sensors or stereo cameras can be used to represent the state of a deformable object $\boldsymbol{st}$ at time step t as a dense, noisy, and occluded point

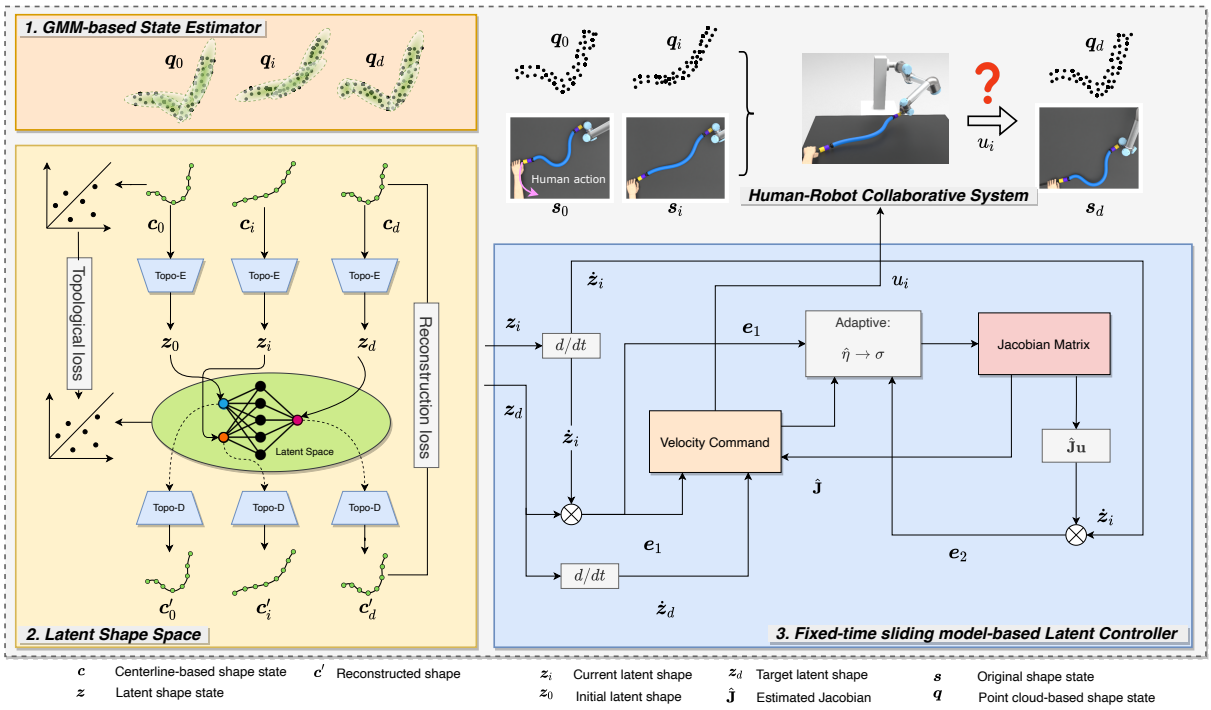


Fig. 4: Conceptual representation of human-robot collaboration for reactive deformable linear object manipulation. The goal of our proposed framework is to design a human-robot collaborative system to automatically execute robot action $\boldsymbol{u}_i$ to reach the desired shape state $\boldsymbol{s}_d$ after the intermediate shape state $\boldsymbol{s}_i$ under action from a human partner given its beginning shape state $\boldsymbol{s}_0$. The proposed framework is composed of three components, namely, (1) a Gaussian Mixture Model (GMM)-based state estimator for the deformable linear object (top left), (2) a latent shape space built upon topological loss using the topological auto-encoder (bottom left), and (3) a fixed-time sliding model-based controller for reactive controls on the manipulated linear object (bottom right).

cloud $\boldsymbol{qt} = q1^t, q2^t, \ldots, q_M^t \in \mathbb{R}^{M\times D}$, where M is the point cloud resolution and D is the point dimension. The goal of the state estimator is to estimate a concise and simplified representation of the state denoted by a series of centerline points $C = \{c_1^t, c_2^t, \ldots, c_N^t\} \in \mathbb{R}^{N\times D}$ at time step t, where $c_i^t \in \mathbb{R}^{1\times 3}$ represents the 3D coordinate of the i-th key point at time step t. Structure preserved registration (SPR) [45] consider that the perceived point clouds $\boldsymbol{q}_t$ are sampled $\boldsymbol{c}_t$ from a Gaussian Mixture Model (GMM), and centroids of the point cloud represent the key points of the deformable linear object shape $\boldsymbol{s}_t$. Based on Bayes' theorem, the probability of a point q_m^t sampled from the mixture model can be defined as below:

$$p\left(q_m^t\right) = \sum_{n=1}^{N+1} p(n)p\left(q_m^t \mid n\right) \tag{7}$$

where $p(n)$ denote the weight of the n-th mixture component, and $p\left(q_m^t \mid n\right)$ denote the probability of sampling q_m^t from the n-th mixture component. Assuming that all Gaussians have equal weight, a uniform distribution used for handling noise and outliers can be expressed as below:

$$p(n) = \begin{cases} (1-\mu)\frac{1}{n}, & n = 1, \ldots, N \\ \mu, & n = N+1 \end{cases} \tag{8}$$

$$p\left(q_m^t \mid n\right) = \begin{cases} \mathcal{N}\left(q_m^t; c_n^t, \sigma^2\mathbf{I}\right), & n = 1, \ldots, N \\ \frac{1}{M}, & n = N+1 \end{cases} \tag{9}$$

The main objective is to maximize the log-likelihood $\mathcal{L}$ sampled from the point cloud $\boldsymbol{q}_t$, which can be formulated as a problem of Maximum Likelihood Estimation (MLE). The optimization of mixture centroids for maximizing the log-likelihood function $\mathcal{L}$ is non-convex due to the summation inside $\log(\cdot)$, making direct optimization infeasible. Thus, we construct another log-likelihood function $\mathcal{O}$ having a lower bound of $\mathcal{L}$. The maximization of $\mathcal{O}$ through the EM algorithm [46] involves the E-step (expectation step) and M-step (maximization step), that iteratively estimate $\left(c_n^t, \sigma^2\right)$ by maximizing $\mathcal{O}$. The formula for $\mathcal{O}$ is given as:

$$\mathcal{O}\left(c_n^t, \sigma^2\right) = \sum_{m=1}^{M}\sum_{n=1}^{N+1} p\left(n \mid q_m^t\right) \log\left(p(n)p\left(q_m^t \mid n\right)\right) \tag{10}$$

It is worth noting that by moving the inside summation of $\log(\cdot)$ to the front, it becomes more convenient for further computation. The optimization of $\mathcal{O}$ is aimed to increase the value of $\mathcal{L}$ except at local optima, and The use of Jensen's inequality [47] can demonstrate that function $\mathcal{O}$ serves as a lower bound for function $\mathcal{L}$. As a result, elevating the value of $\mathcal{O}$ will inevitably lead to an increase in the value of $\mathcal{L}$ unless it has already reached a local optimum. By comparing the structures of $\mathcal{O}$ and $\mathcal{L}$, we see that the summation inside the logarithm in $\mathcal{L}$ has been moved to the front in $\mathcal{O}$, providing computational convenience. The EM algorithm [46] can be used with the definition of the complete log-likelihood function to iteratively estimate $\left(c_n^t, \sigma^2\right)$ by maximizing $\mathcal{O}$ through the E-step and M-step.

By utilizing the definition of the complete log-likelihood function, the EM algorithm [46] can be employed to iteratively estimate $\left(c_n^t, \sigma^2\right)$ by maximizing O through E-step and M-step.

Algorithm 1: Persistent Diagram Calculation for Deformable Shapes with Vietoris-Rips Complex

Input: Distance matrix $\mathbf{D}^C$ on the deformable shape space C, Maximum scale parameter $\varepsilon_{\max}$
Output: Persistence diagram $\mathcal{G}^C$ on the deformable shape space C

Initialize an empty set of simplices S ;
Initialize an empty list of birth-death pairs $\mathcal{G} = []$;
for *each entry* (i, j) *in* $\mathbf{D}$ **do**
  if $\mathbf{D}_{ij} \leq \varepsilon_{\max}$ **then**
    Add the edge e_{ij} to S ;
  end
end
Sort S in non-decreasing order of edge length ;
for *each edge* e_{ij} *in* S **do**
  Add (i, j) to the Vietoris-Rips complex $\mathcal{V}$;
  Compute the connected components of $\mathcal{V}$;
  if *adding pairing* (i, j) *created a new connected component* **then**
    $\phi^C = (i, j)$;
    Add the birth time of the new component to $\mathcal{G}$ as $(\mathbf{D}_{ij}, \infty)$;
  else
    Update the death time of the older component in $\mathcal{G}$ to $\mathbf{D}_{ij}$;
  end
end
return $\mathcal{G}$;

5.2. Latent Shape Space

Due to the infinite configurations and complex dynamics of deformable objects, it is difficult to characterize the shapes with topological features. In addition, the use of centerline-based shape representation poses challenges when directly incorporated into a controller. Due to its high dimensionality, it complicates the process of identifying the optimal solution for control actions. This complexity can potentially lead to instability if the controller struggles to quickly ascertain a suitable solution, or if the derived solution ends up being suboptimal [48]. Therefore, designing an effective low-dimensional representation for the deformable objects to reduce the feature dimension and preserve topological structure is necessary. In this article, by using persistent homology, we propose a generic approach that applies topological auto-encoders [49] to calculate topological signatures for both the original shape space and latent shape space to derive a topological loss term when training an auto-encoder network. Fig. 6 (also see the second component in Fig. 4) depicts an overview of our method, and we divide this learning process into three individual steps in the following.

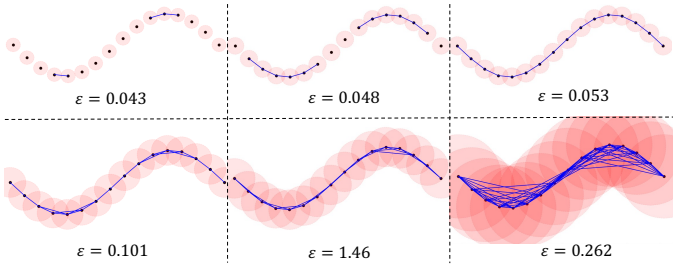


Fig. 5: Conceptual representation of Vietoris-Rips complex computation on an "S"-shaped deformable linear object.

5.2.1. Vietoris-Rips Complex Calculation

To begin, we employ the distance matrix $\mathbf{D}^C$ to compute the persistent homology of the Vietoris-Rips complex of a deformable shape space C (each shape in the shape space is represented as a set of centerline points. In this work, we choose to use the Euclidean distance for the calculation of $\mathbf{D}^C$, but other distances can be used as well. We then determine $\epsilon := \max \mathbf{D}^C$ and construct the corresponding Vietoris-Rips complex, denoted by $\mathcal{V}\epsilon\left(\mathbf{D}^S\right)$. As illustrated in Fig. 5, we provide a detailed example of Vietoris-Rips complex computation on a "S"-shaped deformable linear object where birth time and death time are updated after adding each pairing (i, j). For a dimension $d \in \mathbb{N}> 0$, a set of persistence diagrams $\mathcal{G}^C$ and a set of persistence pairings ϕ^C can be obtained. The persistence pairing ϕ_d^C for dimension d consists of the indices of simplices that participate in the emergence and disappearance of topological characteristics in d dimensions. Persistent homology computation identifies a set of edge indices that are deemed "topologically significant," and each of these sets is represented by a persistence pairing. Alg. 1 shows the detailed computation process of persistent diagram for Deformable Shapes with Vietoris-Rips Complex.

5.2.2. Selecting indices from pairings

In this section, our objective is to choose indices from the persistence pairing and transform them into a distance metric between two vertices. We modify this distance to align the topological characteristics of the input space and the latent space. For 0-dimensional topological features, we only need to examine the indices of edges, which are the " "destroyer" simplices, in the pairing ϕ_0^C. Our preliminary experiments suggest that utilizing 1-dimensional topological features only prolongs the computation time. As a result, subsequent experiments will exclusively concentrate on 0-dimensional persistence diagrams. Hence, we denote the 0-dimensional persistence diagram and pairing of C as $\left(\mathcal{G}^C, \phi^C\right)$.

5.2.3. Topological Autoencoder

We start by considering a mini-batch $\boldsymbol{c}$ consisting of l points from the shape data space C (i.e., a set of centerline points). We then construct an autoencoder using a composite function $f_h \circ f_g$, where $f_h : C \to \mathcal{Z}$ is the encoder function and $f_g : \mathcal{Z} \to C$ is the decoder function. Here, z denotes the latent codes obtained

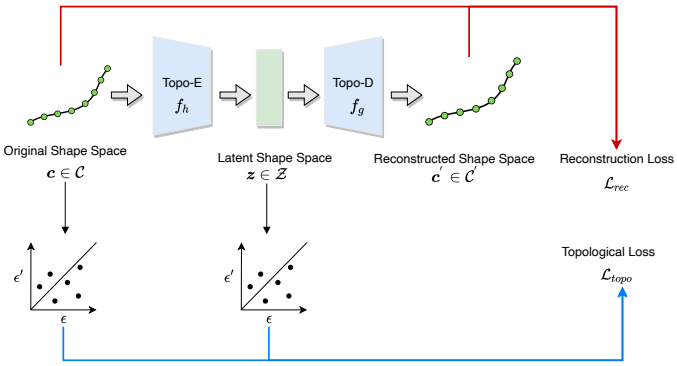


Fig. 6: An overview of the topological-aware latent shape space construction approach. Given a mini-batch shape represented by a spline denoted by a set of centerline key points $\boldsymbol{c}$ in original shape space C, we apply the topological auto-encoder to reconstruct $\boldsymbol{c}$, resulting in a reconstruction $\boldsymbol{c}'$. In addition to the usual reconstruction loss, we calculate our *topological loss* based on the topological structure differences between persistence diagrams computed from $\boldsymbol{c}$ and its latent variable $z \in \mathcal{Z}$. The goal of the topological loss is to guide the auto-encoder to preserve the topological features of the original shape space into the latent representations.

by applying the encoder function to the mini-batch $\mathbf{c}$, i.e., $\mathbf{z} = f_h(\mathbf{c})$. In a forward pass, we compute the persistent homology in both the original shape space and the generated latent space, as below:

$$\begin{aligned}(\mathcal{G}^s, \phi^s) &:= \mathcal{H}\left(\mathcal{V}_\epsilon(\boldsymbol{c})\right) \\ (\mathcal{G}^z, \phi^z) &:= \mathcal{H}\left(\mathcal{V}_\epsilon(z)\right)\end{aligned} \tag{11}$$

To obtain the persistence diagram values, we use the edge indices provided by the persistence pairings to subset the distance matrix. We can represent the persistence diagram as a set that contains the same information as the distances retrieved with the pairing, denoted as $\mathcal{G}^{\mathbf{c}} \simeq \mathbf{D}^{\mathbf{c}}[\phi^{\mathbf{c}}]$. We treat $\mathbf{D}^{\mathbf{c}}[\phi^{\mathbf{c}}]$ as a vector in $\mathbb{R}^{|\phi^{\mathbf{c}}|}$. By comparing the persistence diagrams obtained from the data space and latent space, we can construct a topological regularization term $\mathcal{L}_{topo}$, which is added to the reconstruction loss of an autoencoder. The overall loss function is then given by:

$$\mathcal{L} = \mathcal{L}_{rec}(\mathbf{c}, f_g(f_h(\mathbf{c}))) + \lambda \mathcal{L}_{topo} \tag{12}$$

where $\mathcal{L}_{rec}$ is the reconstruction loss, f_h and f_g are the encoder and decoder functions respectively, and λ is a regularization parameter that controls the strength of the regularization.

Let us consider how to express $\mathcal{L}_{topo}$. We select edge indices from $\pi^{\boldsymbol{c}}$ and $\pi^{\boldsymbol{z}}$ to calculate the $\mathcal{V}$ value, which represents topologically relevant distances from the distance matrix. Each persistence diagram entry indicates a distance between two data points. To ensure unbiased estimation and efficient training, we take into account the union set arising from selected edges in $\boldsymbol{c}$ and $\boldsymbol{z}$. The topological loss term of the autoencoder consists of two parts that tackle the "directed" loss that arises when topological characteristics in one of the two spaces remain unchanged. Thus, $\mathcal{L}_{topo} = \mathcal{L}_{C\to\mathcal{Z}} + \mathcal{L}_{\mathcal{Z}\to C}$, where

$$\mathcal{L}_{C\to\mathcal{Z}} := \frac{1}{2}\left\|\mathbf{D}^{\boldsymbol{c}}\left[\phi^{\boldsymbol{c}}\right] - \mathbf{D}^{\boldsymbol{z}}\left[\phi^{\boldsymbol{c}}\right]\right\|^2$$

and

$$\mathcal{L}_{\mathcal{Z}\to C} := \frac{1}{2}\left\|\mathbf{D}^{\boldsymbol{z}}\left[\phi^{\boldsymbol{z}}\right] - \mathbf{D}^{\boldsymbol{c}}\left[\phi^{\boldsymbol{z}}\right]\right\|^2,$$

By considering the union set arising from selected edges, an informative loss can be determined by at least $|\boldsymbol{c}|$ distances. Our formulation aims to align the distances between $\boldsymbol{c}$ and $\boldsymbol{z}$, which in turn leads to an alignment of distances between C and $\mathcal{Z}$.

If the two spaces are perfectly aligned, then $\mathcal{L}_{C\to\mathcal{Z}}$ and $\mathcal{L}_{\mathcal{Z}\to C}$ are both equal to zero, as the pairings and corresponding distances coincide. However, if $\mathcal{L}_{topo} = 0$, it does not necessarily mean that the persistence pairings and diagrams are identical. To calculate the gradient, we use $\boldsymbol{\omega}$ to represent the encoder parameters, and $\boldsymbol{\delta} := (\mathbf{D}^{\boldsymbol{c}}\left[\phi^{\boldsymbol{c}}\right] - \mathbf{D}^{\boldsymbol{z}}\left[\phi^{\boldsymbol{c}}\right])$. The partial derivative of $\mathcal{L}_{C\to\mathcal{Z}}$ with respect to $\boldsymbol{\omega}$ can be obtained as follows:

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{\omega}}\mathcal{L}_{C\to\mathcal{Z}} &= \frac{\partial}{\partial \boldsymbol{\omega}}\left(\frac{1}{2}\left\|\mathbf{D}^{\boldsymbol{c}}\left[\phi^{\boldsymbol{c}}\right] - \mathbf{D}^{\boldsymbol{z}}\left[\phi^{\boldsymbol{c}}\right]\right\|^2\right) \\ &= -\boldsymbol{\delta}^\top\left(\frac{\partial \mathbf{D}^{\boldsymbol{z}}\left[\phi^{\boldsymbol{c}}\right]}{\partial \boldsymbol{\omega}}\right) \\ &= -\boldsymbol{\delta}^\top\left(\sum_{i=1}^{|\phi^{\boldsymbol{c}}|}\frac{\partial \mathbf{D}^{\boldsymbol{z}}\left[\phi^{\boldsymbol{c}}\right]_i}{\partial \boldsymbol{\omega}}\right)\end{aligned}$$

In the above equation, the size of a persistence pairing is denoted by $|\phi^s|$, while $\mathbf{D}^{\boldsymbol{z}}\left[\phi^{\boldsymbol{c}}\right]_i$ indicates the ith component of the vector of paired distances. An analogous derivation applies to $\mathcal{L}_{\mathcal{Z}\to C}$, where $\phi^{\boldsymbol{c}}$ is substituted with $\phi^{\boldsymbol{z}}$. Furthermore, since the distances between input samples are independent of the encoder network, the derivative of $\mathbf{D}^{\boldsymbol{c}}$ with respect to $\boldsymbol{\omega}$ must be zero. Given the stability of the persistence diagram, where small shifts in the function result in only minor modifications to the diagram as outlined by Cohen [50], the diagram remains robust even against infinitesimal alterations of its entries (please refer to the associated definition and theorem detailed in Section 5.3). Consequently, our topological loss maintains differentiability at each update step throughout the training process.

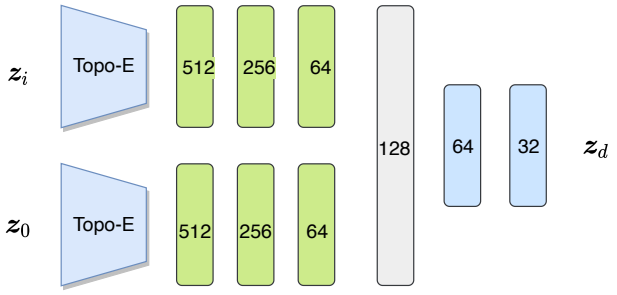


Fig. 7: Conceptual representation of the designed neural network that takes the initial latent shape z_0 and current latent shape z_i as inputs and predicts the target latent shape z_d.

With the built latent shape space, we train a neural network as shown in Fig. 7 that takes as inputs current latent shape z_i and initial latent shape z_0, and outputs its desired latent shape z_d. The dimension of the latent shape space is set to 16 in the following experiments, and we train this neural network by iteratively collecting a data set composed of tuples $\{(z_0, z_i, z_d)\}$ with the bimanual manipulation algorithms in [14].

5.3. Controller Design

Mathematical Properties Some necessary lemmas, assumptions, and definitions related to mathematical properties are given as follows:

Lemma 1. *In [51], the function* $\mathrm{sig}^k(x) = |x|^k \operatorname{sgn}(x)$ *is defined, where* $x \in \mathbb{R}$, $k > 0$, *and* sgn *denotes the standard sign function.*

Lemma 2. *[51]* $\left(\sum_{i=1}^{n} |x_i|\right)^p \leq \sum_{i=1}^{n} |x_i|^p$ *holds for any* $x_i \in \mathbb{R}, i = 1, 2, \ldots, n$, *where* p *is a real number satisfying* $0 < p < 1$.

Lemma 3. *[51]* $n^{1-p}\left(\sum_{i=1}^{n} |x_i|\right)^p \leq \sum_{i=1}^{n} |x_i|^p$ *holds for any* $x_i \in \mathbb{R}, i = 1, 2, \ldots, n$, *and* $p > 1$.

Lemma 4. *[52] For any* $x \in \mathbb{R}$ *and* $\delta > 0$, *we have the inequality satisfies:* $0 \leq |x| - x\tanh(x/\delta) \leq \kappa\delta$ *where* $\kappa = 0.2785$ *with satisfying* $\kappa = e^{-(\kappa+1)}$.

Lemma 5. *[52] For* $h > 0$ *and* $x \geq 0, y > 0$, *the following inequality holds:* $x^h(y - x) \leq (y^{1+h} - x^{1+h})/(1 + h)$.

Lemma 6. *[52] For* $h > 1, x > 0, y \leq x$ *and* $y \in \mathbb{R}$, *it holds that:* $(x - y)^h \geq y^h - x^h$.

Lemma 7. *[53] For a continuous positive definite and radially unbounded function* $V(x) : \mathbb{R}^n \to \mathbb{R}$ *which satisfies the following inequality as shown:*

$$\dot{V}(x) \leq -\alpha_1 V^{a_1}(x) - \beta_1 V^{a_2}(x) + \rho_1 \tag{13}$$

where α_1, β_1, *and* ρ_1 *are all positive constants and the parameters* $a_1 \in (0, 1)$, *and* $a_2 \in (1, \infty)$, *then the trajectory of the system* $\dot{x}(t) = f(x)$ *is practical fixed-time stable. The final convergence domain of the system can be expressed as follows:*

$$\Omega_1 = \{x | V(x) \leq \min\{(\frac{\rho_1}{\alpha_1(1-\theta)})^{\frac{1}{a_1}}, (\frac{\rho_1}{\beta_1(1-\theta)})^{\frac{1}{a_2}}\}\} \tag{14}$$

where $\theta \in (0, 1)$ *is a positive constant. The settling time in such a system to reach the residual set can be expressed as follows:*

$$T \leq T_{\max} = \frac{1}{\alpha_1\theta(1 - a_1)} + \frac{1}{\beta_1\theta(a_2 - 1)} \tag{15}$$

Definition 1. *Let* X *and* Y *be two non-empty subsets of a metric space* (M, d), *the Hausdorff distance* $d_{\mathrm{H}}(X, Y)$ *and the bottleneck are defined as:*

$$\begin{aligned} d_{\mathrm{H}}(X, Y) &= \max\left\{\sup_{x \in X} d(x, Y), \sup_{y \in Y} d(X, y)\right\} \\ d_{\mathrm{B}}(X, Y) &= \inf_{\varphi: X \to Y} \sup_{x \in X} \|x - \varphi(x)\|_q \end{aligned} \tag{16}$$

where sup *represents the supremum and* $d(a, B) = \inf_{b \in B} d(a, b)$ *(*inf *denotes the infimum) quantifies the distance from a point* $a \in X$ *to the subset* $B \subseteq X$, *and* $q \leq \infty$ *and* φ *ranges over bijections between* X *and* Y.

Theorem 1. *The* ***stability of the persistence diagram****: Given two functions,* f *and* g, *and their corresponding persistence diagrams,* $\mathrm{D}(f)$ *and* $\mathrm{D}(g)$, *on a topological space, the bottleneck distance between the of the persistence diagrams bounds the* L_∞*-norm between the two functions:*

$$d_{\mathrm{B}}(\mathrm{D}(f), \mathrm{D}(g)) \leq \|f - g\|_\infty \tag{17}$$

The assumptions required for this result are mild and are satisfied by Morse functions on compact manifolds, piecewise linear functions on simplicial complexes, and more. The bottleneck distance is based on a bijection between the points and is therefore always at least the Hausdorff distance between the two diagrams.

Definition 2. *In [54], we can find the following vectorial power definitions for any arbitrary vector* $\mathbf{x} \in \mathbb{R}^n$*:*

$$\begin{aligned} \mathrm{sig}^k(\mathbf{x}) &= \left[\mathrm{sig}^k(x_1), \ldots, \mathrm{sig}^k(x_n)\right]^\mathsf{T} \in \mathbb{R}^n \\ |\mathbf{x}|^k &= \mathrm{diag}\left\{|x_1|^k, \ldots, |x_n|^k\right\} \in \mathbb{R}^{n \times n} \end{aligned}$$

The fixed-time sliding mode control is used to control the shape of the elastic rod. Throughout this paper, we denote the velocity motion of the robot as $\mathbf{u} = \dot{\mathbf{r}}$ for simplicity. The shape-motion relationship considering Assumption 3 satisfies:

$$\dot{\mathbf{s}} = \mathbf{J}\mathbf{u} = \hat{\mathbf{J}}\mathbf{u} + \check{\mathbf{J}}\mathbf{u} \tag{18}$$

Two error variables are defined:

$$\mathbf{e}_1 = \mathbf{s} - \mathbf{s}_d, \quad \mathbf{e}_2 = \dot{\mathbf{s}} - \hat{\mathbf{J}}\mathbf{u} \tag{19}$$

and its derivative with respect to time is:

$$\dot{\mathbf{e}}_1 = \dot{\mathbf{s}} - \dot{\mathbf{s}}_d, \quad \dot{\mathbf{e}}_2 = \ddot{\mathbf{s}} - \dot{\hat{\mathbf{J}}}\mathbf{u} - \hat{\mathbf{J}}\dot{\mathbf{u}} \tag{20}$$

Combining with (18), it yields

$$\dot{\mathbf{e}}_1 = \hat{\mathbf{J}}\mathbf{u} + \tilde{\mathbf{J}}\mathbf{u} - \dot{\mathbf{s}}_d \tag{21}$$

The velocity control input can be defined as below:

$$\mathbf{u} = \hat{\mathbf{J}}^+\left(\dot{\mathbf{s}}_d - a_{11}\,\mathrm{sig}^{2b_{11}-1}(\mathbf{e}_1) - a_{12}\,\mathrm{sig}^{2b_{12}-1}(\mathbf{e}_1)\right) \tag{22}$$

where $\hat{\mathbf{J}}^+$ is the pseudo-inverse of the Jacobian matrix $\hat{\mathbf{J}}$. $a_{11} > 0, a_{12} > 0, 0 < b_{11} < 1, b_{12} > 1$ are design parameters specifying the convergence speed of the controller (22) and the system stability indirectly. In order to measure the error of shape tracking, a quadratic function is introduced:

$$V_1(\mathbf{e}_1) = \frac{1}{2}\mathbf{e}_1^\mathsf{T}\mathbf{e}_1 \tag{23}$$

Time differentiation of (23) yields

$$\dot{V}_1(\mathbf{e}_1) = \mathbf{e}_1^\mathsf{T}\dot{\mathbf{e}}_1 = \mathbf{e}_1^\mathsf{T}(\hat{\mathbf{J}}\mathbf{u} - \dot{\mathbf{s}}_d) + \mathbf{e}_1^\mathsf{T}\tilde{\mathbf{J}}\mathbf{u} \tag{24}$$

Substituting the controller (22) into (24), one can get

$$\dot{V}_1 = \mathbf{e}_1^\mathsf{T}\left(-a_{11}\,\mathrm{sig}^{2b_{11}-1}(\mathbf{e}_1) - a_{12}\,\mathrm{sig}^{2b_{12}-1}(\mathbf{e}_1)\right) + \mathbf{e}_1^\mathsf{T}\tilde{\mathbf{J}}\mathbf{u}$$

Considering Lemma 2 and Lemma 3, it yields

$$\dot{V}_1 \leq -a_{11}\|\mathbf{e}_1\|^{2b_{11}} - a_{12}p^{1-b_{12}}\|\mathbf{e}_1\|^{2b_{12}} + \mathbf{e}_1^{\intercal}\tilde{\mathbf{J}}\mathbf{u} \tag{25}$$

By considering Young's inequality, it can obtain the inequality as follows:

$$\mathbf{e}_1^{\intercal}\tilde{\mathbf{J}}\mathbf{u} \leq \|\mathbf{e}_1\|^2/4 + \eta\|\mathbf{u}\|^2 \tag{26}$$

Substituting (26) into (25) obtains:

$$\dot{V}_1 \leq -a_{11}\|\mathbf{e}_1\|^{2b_{11}} - a_{12}p^{1-b_{12}}\|\mathbf{e}_1\|^{2b_{12}} + \frac{1}{4}\|\mathbf{e}_1\|^2 + \eta\|\mathbf{u}\|^2$$

The adaptation rule of the DJM can be defined as:

$$\begin{aligned}\dot{\hat{\mathbf{J}}} &= (a_{21}\,\mathrm{sig}^{\,2b_{21}-1}(\mathbf{e}_2) + a_{22}\,\mathrm{sig}^{\,2b_{22}-1}(\mathbf{e}_2) + \ddot{\mathbf{s}} - \hat{\mathbf{J}}\dot{\mathbf{u}} + \sigma)\mathbf{u}^{+}\\ \boldsymbol{\sigma} &= \mathbf{e}_2^{\intercal+}(\hat{\eta}\tanh(\frac{\|\mathbf{u}\|^2}{\delta})\|\mathbf{u}\|^2 + \frac{1}{4}\|\mathbf{e}_1\|^2)\end{aligned} \tag{27}$$

where $a_{21} > 0, a_{22} > 0, 0 < b_{21} < 1, b_{22} > 1$ are design parameters determining the convergence speed of the approximation of the Jacobian matrix. The adaptive rule of $\hat{\eta}$ is designed as follows:

$$\dot{\hat{\eta}} = \tanh(\|\mathbf{u}\|^2/\delta)\|\mathbf{u}\|^2 - a_{31}\hat{\eta}^{2b_3-1} - a_{32}\hat{\eta}^{2b_3+1} \tag{28}$$

where δ is a positive constant. $a_{31} > 0, a_{32} > 0$, $\frac{1}{2} < b_3 < 1$ are design parameters. Define the quadratic function $V_2(\mathbf{e}_2) = \frac{1}{2}\mathbf{e}_2^{\intercal}\mathbf{e}_2$, and differentiating V_2 with respect to time and using (20) gains

$$\dot{V}_2 = \mathbf{e}_2^{\intercal}\dot{\mathbf{e}}_2 = \mathbf{e}_2^{\intercal}(\ddot{\mathbf{s}} - \dot{\hat{\mathbf{J}}}\mathbf{u} - \hat{\mathbf{J}}\dot{\mathbf{u}}) \tag{29}$$

Substituting (27) into (29), and considering Lemma 2 and Lemma 3, it yields

$$\begin{aligned}\dot{V}_2 &= -a_{21}\mathbf{e}_2^{\intercal}\,\mathrm{sig}^{\,2b_{21}-1}(\mathbf{e}_2) - a_{22}\mathbf{e}_2^{\intercal}\,\mathrm{sig}^{\,2b_{22}-1}(\mathbf{e}_2) - \mathbf{e}_2^{\intercal}\boldsymbol{\sigma}\\ &\leq -a_{21}\|\mathbf{e}_2\|^{2b_{21}} - a_{22}p^{1-b_{22}}\|\mathbf{e}_2\|^{2b_{22}} - \mathbf{e}_2^{\intercal}\boldsymbol{\sigma}\end{aligned} \tag{30}$$

Proposition 1. *Assuming the dynamic system* (18) *is in closed-loop with the controller* (22) *under the conditions of Assumptions 3 and 4, with the Jacobian approximation* (27) *and the adaptive update rule* (28)*, two conclusions can be drawn: 1) all signals within the closed-loop system remain uniformly ultimately bounded (UUB); 2) the deformation error* $\mathbf{e}_1$ *converges to a compact set near zero within a fixed time frame, with no occurrence of any singularities during the task.*

Consider the energy-like function:

$$V = V_1 + V_2 + \frac{1}{2}\tilde{\eta}^2 \tag{31}$$

where $\tilde{\eta} = \eta - \hat{\eta}$ is the estimation error, with $\hat{\eta}$ being the estimation of η. With $\dot{V}_1$ and $\dot{V}_2$, time differentiation of (31) yields

$$\begin{aligned}\dot{V} \leq\ & -a_{11}\|\mathbf{e}_1\|^{2b_{11}} - a_{21}\|\mathbf{e}_2\|^{2b_{21}} - a_{12}p^{1-b_{12}}\|\mathbf{e}_1\|^{2b_{12}} - \tilde{\eta}\dot{\hat{\eta}}\\ & - a_{22}p^{1-b_{22}}\|\mathbf{e}_2\|^{2b_{22}} + \frac{1}{4}\|\mathbf{e}_1\|^2 + \eta\|\mathbf{u}\|^2 - \mathbf{e}_2^{\intercal}\boldsymbol{\sigma}\end{aligned} \tag{32}$$

With the adaptive update rule (28) and Lemma 4, we can get the following inequality:

$$\begin{aligned}&\tfrac{1}{4}\|\mathbf{e}_1\|^2 + \eta\|\mathbf{u}\|^2 - \mathbf{e}_2^{\intercal}\boldsymbol{\sigma} - \tilde{\eta}\dot{\hat{\eta}}\\ &= \eta\|\mathbf{u}\|^2 - \hat{\eta}\tanh\left(\tfrac{\|\mathbf{u}\|^2}{\delta}\right)\|\mathbf{u}\|^2 - \tilde{\eta}\dot{\hat{\eta}}\\ &\leq \eta\kappa\delta + \tilde{\eta}\left(\|\mathbf{u}\|^2\tanh\left(\tfrac{\|\mathbf{u}\|^2}{\delta}\right) - \dot{\hat{\eta}}\right)\\ &\leq \eta\kappa\delta + a_{31}\tilde{\eta}\hat{\eta}^{2b_3-1} + a_{32}\tilde{\eta}\hat{\eta}^{2b_3+1}\end{aligned} \tag{33}$$

And, considering Lemma 5 and Lemma 6, we have:

$$\tilde{\eta}\hat{\eta}^{2b_3-1} \leq \frac{2\eta^{2b_3} - \tilde{\eta}^{2b_3}}{2b_3}, \tilde{\eta}\hat{\eta}^{2b_3+1} \leq \frac{2\eta^{2b_3+2} - \tilde{\eta}^{2b_3+2}}{2b_3+2} \tag{34}$$

Substituting (33) and (34) into (32), it yields

$$\begin{aligned}\dot{V} =\ & -(a_{11}\|\mathbf{e}_1\|^{2b_{11}} + a_{21}\|\mathbf{e}_2\|^{2b_{21}})\\ & - (a_{12}p^{1-b_{12}}\|\mathbf{e}_1\|^{2b_{12}} + a_{22}p^{1-b_{22}}\|\mathbf{e}_2\|^{2b_{22}})\\ & - \frac{a_{31}}{2b_3}\tilde{\eta}^{2b_3} - \frac{a_{32}}{2b_3+2}\tilde{\eta}^{2b_3+2}\\ & + (\eta\kappa\delta + \frac{a_{31}}{b_3}\eta^{2b_3} + \frac{a_{32}}{b_3+1}\eta^{2b_3+2})\\ \leq\ & -a_1V^{b_1} - a_2V^{b_2} + \Omega\end{aligned} \tag{35}$$

where the coefficients are:

$$\begin{aligned}a_1 &= \min(2a_{11}, 2a_{21}, \frac{a_{31}}{b_3})\\ a_2 &= \min(2a_{12}p^{1-b_{12}}, 2a_{22}p^{1-b_{22}}, \frac{a_{32}}{b_3+1})\\ b_1 &= \min(b_{11}, b_{21}, b_3),\ b_2 = \min(b_{12}, b_{22}, b_3+1)\\ \Omega &= \eta\kappa\delta + \frac{a_{31}}{b_3}\eta^{2b_3} + \frac{a_{32}}{b_3+1}\eta^{2b_3+2}\end{aligned} \tag{36}$$

By selecting appropriate parameters that ensure $a_1 > 0, a_2 > 0, b_1 \in (0,1), b_2 \in (1,+\infty)$, and referring to (35) and Lemma 7, V converges to the compact set:

$$\lim_{t\to T_{\max}} V \leq V_m = \min\{(\frac{\Omega}{a_1(1-\varpi)})^{\frac{1}{b_1}}, (\frac{\Omega}{a_2(1-\varpi)})^{\frac{1}{b_2}}\} \tag{37}$$

where $\varpi \in (0,1)$ is a user-define constant, within the fixed convergence time $T_{\max}$ calculated as [53]:

$$T_{\max} \leq \frac{1}{a_1\varpi(1-b_1)} + \frac{1}{a_2\varpi(b_2-1)} \tag{38}$$

It implies that all states of the closed-loop system are bounded. Define the augmented variable $\mathbf{e} = [\mathbf{e}_1^{\intercal}, \mathbf{e}_2^{\intercal}]^{\intercal}$, and further from the construction of V (given in (31)), $\frac{1}{2}\mathbf{e}^{\intercal}\mathbf{e} \leq V_m$ can be derived. Then, we have:

$$\Omega_{\mathbf{e}} = \{\mathbf{e} \| \|\mathbf{e}\| \leq \sqrt{2V_m}, t \geq T_{\max}\} \tag{39}$$

By adjusting $a_i, b_i, i = 1, 2$, we can make the convergence range of V smaller. The above analysis implies that the shape error $\mathbf{e}_1$ and motion estimation error $\mathbf{e}_2$ converge to a compact set around zero within a fixed-time [53]. The above results demonstrate that the DLO can be deformed into the target configuration within a fixed time and all signals as practical fixed-time stable.

The proposed controller (35) is the overdetermined format, which means that the error $\mathbf{e}_1$ can only converge to a local optimum whose size depends on the accessibility of the desired shape $\mathbf{s}_d$. A local minimum is unavoidable for this class of DJM form-based manipulation tasks.

Remark 4. *The fixed-time SMC controller in this paper aims to regulate the dynamic performance of the system without considering the initial value of the system, rather than focusing only on the system acceleration. There are times when low-speed manipulation of DLOs is equally important in industry.*

Remark 5. *Discrete difference [55] and Levant differentiator [56] are used to obtain* $\dot{\mathbf{s}}$ *and* $\ddot{\mathbf{s}}$ *in the experiments. In addition, the tactile sensor can be used to measure friction between the gripper (e.g., Robotiq-85) and objects. Then it monitors whether grippers and objects are connected in real time.*

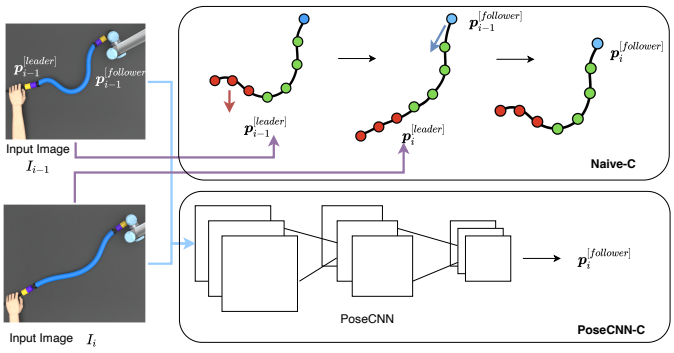


Fig. 8: Conceptual representation of our two baseline models for solving reactive shape servoing tasks of the deformable linear object under human-robot collaboration. The upper **Naïve-C** model employs the transformation of the leftmost k centerline points to compute the target desired pose, while the lower **PoseCNN-C** model utilizes the PoseCNN framework to predict the transformation of the deformable object from the leader side for the desired pose calculation.

5.4. Comparison with Existing Methods

Our work is pioneering in its approach to visual servoing tasks for deformable linear objects in the Human-Robot Collaboration (HRC) context. As it's the first to consider deformable linear manipulation in this context, finding a perfectly matching SOTA method for comparison poses a significant challenge. Despite this, we have made an extensive effort to compare our approach with SOTA methods in each related domain: Visual Servoing (VS), Deformable Object Manipulation (DOM), and Human Robot Collaboration (HRC). For VS, we selected a technique for adaptively manipulating deformable objects using model-free visual servoing [57]. This technique stands out as a SOTA method within VS. Please note that our approach is model-free, hence model-based visual servoing methods were not considered for comparison. For DOM, we chose a similar latent shape control (LSC) model [58], referred to as AutoLSC, which combines a naive auto-encoder with a normal sliding model for the deformable linear object manipulation task. This method has demonstrated remarkable performance in DOM and thus represents the SOTA in this field. We also introduced an additional LSC model, referred to as AutoLSC$_2$, that replaces our method with a naive auto-encoder network. This ablation analysis helps to underline the importance of considering the topological property in handling complex deformable objects in the HRC domain.

For HRC, we analyzed our task carefully and found a pattern. Essentially, we can solve the task by moving the robot agent towards the new pose based on the transformation between the previous and current poses on the leader side (human). By leveraging this regularity, we propose two baseline approaches to implement robot controllers for visual image-based robotic deformable linear object manipulation tasks. The first approach is to directly predict the new pose of the follower (the robot agent) by estimating the transformation between the previous and current poses on the leader side with leftmost k centerline points. This is based on the regularity that the relative transformations of the leader side and follower side are identical, which can be denoted as below:

$$\begin{aligned} \boldsymbol{p}_i^{[follower]} &= \boldsymbol{p}_{i-1}^{[follower]} * \mathbf{T}_{\boldsymbol{p}_0^{[leader]}}^{\boldsymbol{p}_i^{[leader]}} \\ \mathbf{T}_{\boldsymbol{p}_j^{[leader]}}^{\boldsymbol{p}_i^{[leader]}} &= \mathbf{T}_{\boldsymbol{p}_j^{[follower]}}^{\boldsymbol{p}_i^{[follower]}} \quad (j \leq i) \end{aligned} \tag{40}$$

where $\mathbf{T}_{\boldsymbol{p}_0^{[leader]}}^{\boldsymbol{p}_i^{[leader]}}$ is the transformation of the leader side of the deformable linear object between time step 0 and time step i. This transformation can be roughly estimated by the leftmost k centerline points with ICP algorithms. Since this approach is simple and straightforward for solving deformable object manipulation tasks, we name it **Naïve-C** controller and its main process is illustrated in the upper part of Fig. 8. The leftmost k centerline points of the leader side are marked with red color (here $k = 3$ for illustration). After the motion of the leader side of the deformable linear object, the leader poses will become $\boldsymbol{p}_i^{[leader]}$ from $\boldsymbol{p}_{i-1}^{[leader]}$, then **Naïve-C** estimates a transformation $\mathbf{T}_{\boldsymbol{p}_0^{[leader]}}^{\boldsymbol{p}_i^{[leader]}}$ to finally generate the desired pose for the follower side using the Equ. 40. Nonetheless, this approach has to compute an optimal transformation before sending the control commands to the manipulator, which is not quite efficient since its computing process involves an iterative calculation. Besides, it also needs to look for an appropriate trade-off between the shape manipulation accuracy and system response time. Because the expansive optimization-solving process will always reduce the system response time and perform poorly during the human-robot interaction process. The second baseline approach is to directly estimate the $\boldsymbol{p}_i^{[follower]}$ based on a visual observation using PoseCNN to implement a robot controller (indicated by **PoseCNN-C**) for solving the manipulation task. This approach requires collecting a large dataset composed of a series of tuples $\{(\boldsymbol{p}_0, \boldsymbol{p}_i, \boldsymbol{p}_d)\}$. Nevertheless, this approach solely relies on data, and hence, lacks the ability to comprehend the impact of the deformed shape on the behavior of the robot manipulator. Moreover, the absence of modeling the deformable object's geometry could potentially hinder the model's applicability and generalizability.

Table 1: Performance of Latent Representation Approaches on Different Shape Categories

Category	ℓ-Trust				ℓ-Cont				RMSE			
	PCA	TSNE	AE	TAE	PCA	TSNE	AE	TAE	PCA	TSNE †	AE	TAE
Line-shaped	0.907	0.953	0.947	**0.982**	0.859	**0.903**	0.897	0.902	1.572	-	1.770	**0.968**
Pos. Arch-shaped	0.868	0.910	0.884	**0.939**	0.846	0.890	0.884	**0.891**	2.060	-	2.199	**1.279**
Neg. Arch-shaped	0.859	0.938	0.909	**0.947**	0.857	0.885	0.879	**0.887**	1.952	-	2.212	**1.256**
Pos. S-shaped	0.788	0.888	0.837	**0.891**	0.837	0.869	0.845	**0.872**	2.546	-	2.380	**1.492**
Neg. S-shaped	0.838	0.865	0.857	**0.886**	0.843	**0.873**	0.862	0.871	2.775	-	2.486	**1.517**
Pos. Helix	0.824	0.863	0.851	**0.887**	0.829	**0.877**	0.854	0.874	2.894	-	2.671	**1.623**
Neg. Helix	0.832	0.872	0.838	**0.879**	0.836	0.874	0.858	**0.876**	3.127	-	2.733	**1.795**

† The absence of Root Mean Squared Error (RMSE) metric for the t-SNE method in the above table is due to two main reasons: (1) t-SNE is an unsupervised technique used for exploration and visualization of high-dimensional data in two or three dimensions, and thus, it might not be appropriate or meaningful to calculate RMSE for the transformed data. (2) t-SNE does not preserve distances between data points from the high-dimensional space in the lower-dimensional space since it preserves the probability distribution of pairwise similarities of points. Hence, calculating an error metric like RMSE, which is based on distances, might not provide a relevant or meaningful evaluation of the t-SNE transformation.

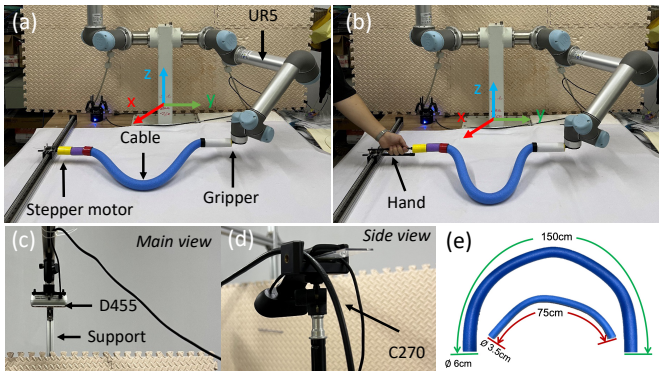


Fig. 9: Experimental setup for reactive deformable object manipulation tasks in the context of human-robot collaboration, which includes an elastic cable to be co-manipulated by a stepper motor and robot end-effector (shown in (a)), and by human hand and robot hand in (b), a depth camera D455 (eye-to-hand configuration) to measure the object's state in (c), a single-arm robot (UR5) to manipulate the cable to maintain its *origin* shape configuration, and two different elastic cables to be considered for validating generalizability.

6. Experiments

In this section, we detail the experimental setup used to validate the efficacy of our proposed human-robot collaboration approach for the task of reactive manipulation of deformable linear objects, specifically focusing on shape servoing tasks. The setup description is followed by an overview of the shape sensory data processing pipeline, which employs a Gaussian Mixture Model (GMM)-based shape state estimator. We then delve into the implementation specifics of the latent shape representation utilizing a topological auto-encoder. We contrast its performance with other prevalent low-dimensional representation approaches across different categories of deformable linear object shapes. In order to replicate stable human-robot interaction during Human-Robot Collaboration (HRC), we initially conducted motor-robot experiments. In these experiments, the left side of the Deformable Linear Object (DLO) was controlled by a stepping motor. This allowed us to compare our proposed method, denoted as **TopoLSC**, with traditional methods including **VS**, **AutoLSC**, and **AutoLSC**$_2$. Following this, we replaced the stepping motor with an actual human hand to conduct human-robot experiments. These trials tested the real-world human-robot collaboration performance using different methodologies. In the human-robot experiments, we compared our proposed method with regularity-based human-robot collaboration approaches (**Naïve-C** and **PoseCNN-C**), as described in Section 5.4, as well as the ablation method (**AutoLSC**$_2$). The superiority of our proposed latent shape controller is substantiated through both quantitative and qualitative measurements of various reactive deformable object manipulation tasks. These tasks were performed in both motor-robot and human-robot experiment contexts.

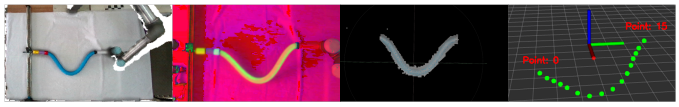
6.1. Experiment Setup

Fig. 9 shows the experiment setup for our approach, where a RealSense RGB-D camera (D455) is used to observe the deformable object manipulation process from a top-down perspective, namely, the main view. Besides, we also consider providing a side view with a commonly used Logi RGB camera (C270) to have a better overview of the entire manipulation process from a third-person perspective. During the process, an elastic sponge bar (viz. a deformable linear object) is manipulated with a UR-5 robot while the other end of the linear object is controlled by a stepping motor or a human hand. Both ends of the elastic sponge bar are connected with a 3D-printed gripper between the robot arm or the human hand. We employ a stepping motor to generate four standard trajectories for a quantitative and effortless measurement of the performance of our proposed approach compared with other advanced approaches (see Fig. 9(a)). Furthermore, a real-time human-robot collaboration is conducted to examine the overall performance of our proposed framework. The robotic manipulation task considered in the context of human-robot collaboration is a shape servoing task in a reactive manner. As shown in Fig. 9, the left side of the deformable linear object is directly manipulated by a human hand, and the other side is manipulated by the UR-5 robot connected with a 3D printed gripper. The robot executes the action to recover the shape of the deformable linear object in real-time after resulting deformations caused by the human hand motions performed on the left side of the object. In our experiments, we set the initial configuration as our final target shape that the robot is trying to reach in real time. For safe operation, there is

the saturation limit for each axis direction, i.e., $|u_i| \leq 0.04m/s$. The proposed algorithm is implemented on ROS/URX running within a servo-control loop of around 20Hz. Utilizing deformation error and velocity convergence as our primary metrics, we employed grid search to refine the parameters of the Sliding Mode Control (SMC). This was initiated with an estimated set of parameters derived from several preliminary experiments. Detailed settings for these parameters can be found in Table 6.1. A video of the conducted experiments can be obtained from `https://sites.google.com/view/hrc-dom`.

Table 2: Experimental Parameters of Fixed-time SMC

Parameter	a_{11}	a_{21}	a_{31}	a_{12}	a_{22}	a_{32}	b_{11}	b_{21}	b_{12}	b_{22}	b_3
Value	4.26	5.52	4.21	4.40	237	259	227	246	246	246	246



(a) RGBD (b) HSV (c) Point cloud (d) 3D Coordinate

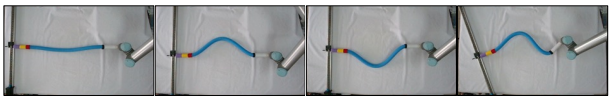
Fig. 10: 3D shape processing pipeline for the reactive deformable object manipulation tasks. Fig. 10a is an aligned RGBD image frame of the working space. Fig. 10b is to extract the deformable object region by using an appropriate HSV color filter. Fig. 10c presents the point cloud of the deformable linear object after computing with the Open3D library. Fig. 10d shows the final 3D centerline to represent the 3D shapes of the manipulated deformable object.

6.2. Shape Estimation

In our experimental platform, the depth camera is set in an eye-to-hand configuration (i.e., fixed pose relative to the robot) and receives the video stream, then we compute the 3D shapes by using the OpenCV and RealSense libraries. Fig. 10 shows the extraction flow of the elastic cable. In the first place, the RGB frame is aligned with the depth image by using RealSense SDK to produce an RGBD frame as shown in Fig. 10a. Then, we mask the deformable object region by designing an appropriate HSV color filter to compute the point cloud of the manipulated object based on camera parameters with the Open3D library (see Fig. 10b). After that, a GMM-based estimator is performed to further extract a fixed number of centerlines (Note that the sequence is still disordered). Therefore, we define the leftmost centerline point as the beginning point to sort the centerline point set. Finally, the visual pipeline ended up with a sequence of fixed and ordered centerline points to represent the 3D shapes of the deformable linear objects.

6.3. Validation of Latent Shape Representation

To validate the performance of the topological auto-encoder on shape representation, we compare our Topological Autoencoder (TAE) with three commonly used representation learning approaches including Principal Component Analysis (PCA), t-distributed Stochastic Neighbor Embedding (TSNE), Autoencoder (AE) on various deformed shapes collected from our built experimental setup. All shapes are stored in the format of 3D centerline points with the visual shape estimator described in



(a) *Line* (b) $Arch_+$ (c) $Arch_-$ (d) S_+

(e) S_- (f) $Helix_+$ (g) $Helix_-$

Fig. 11: Seven different shape categories to measure the latent representation performance for the deformable linear object, namely, *Line*, $Arch_+$, $Arch_-$, S_+, S_-, $Helix_+$ and $Helix_-$ categories.

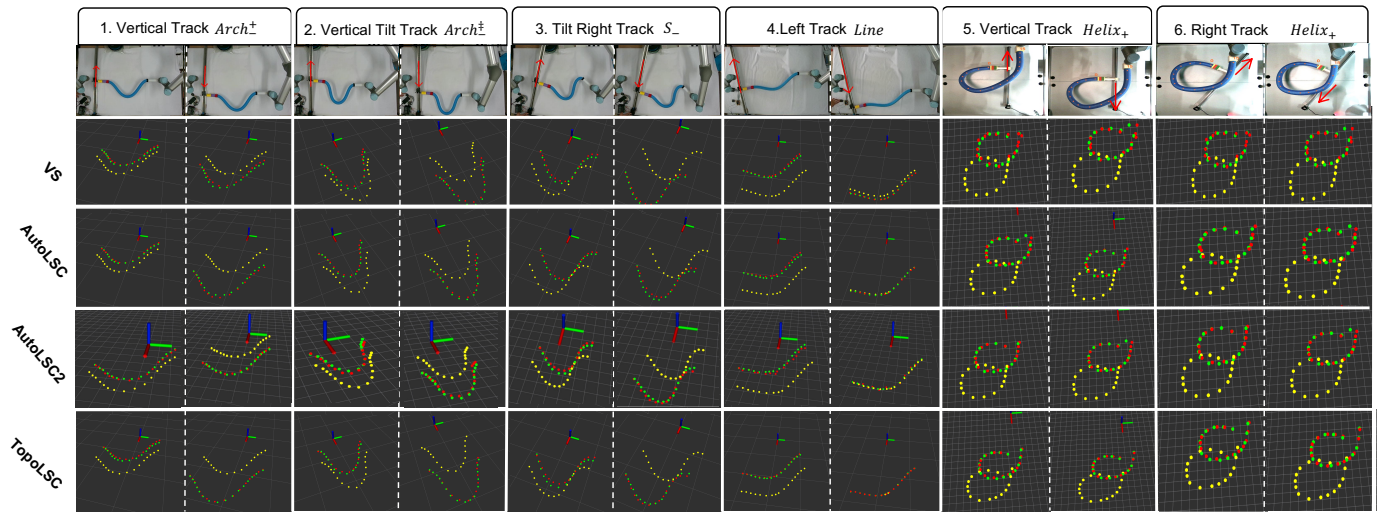
the proceeding section. After examining the collected shapes, we classify them into five different shape categories, namely, **Line**, **Pos. Arch**, **Neg. Arch**, **Pos. S**, **Neg. S**, **Pos. Helix** and **Neg. Helix** class as shown in Fig. 11. We evaluate the reconstruction errors between the input shape c_i and reconstructed shape $\overline{c}_i$ with *root mean square error* (RMSE).

$$\text{RMSE}_{rec} = \|c_i - \overline{c}_i\|^2 \tag{41}$$

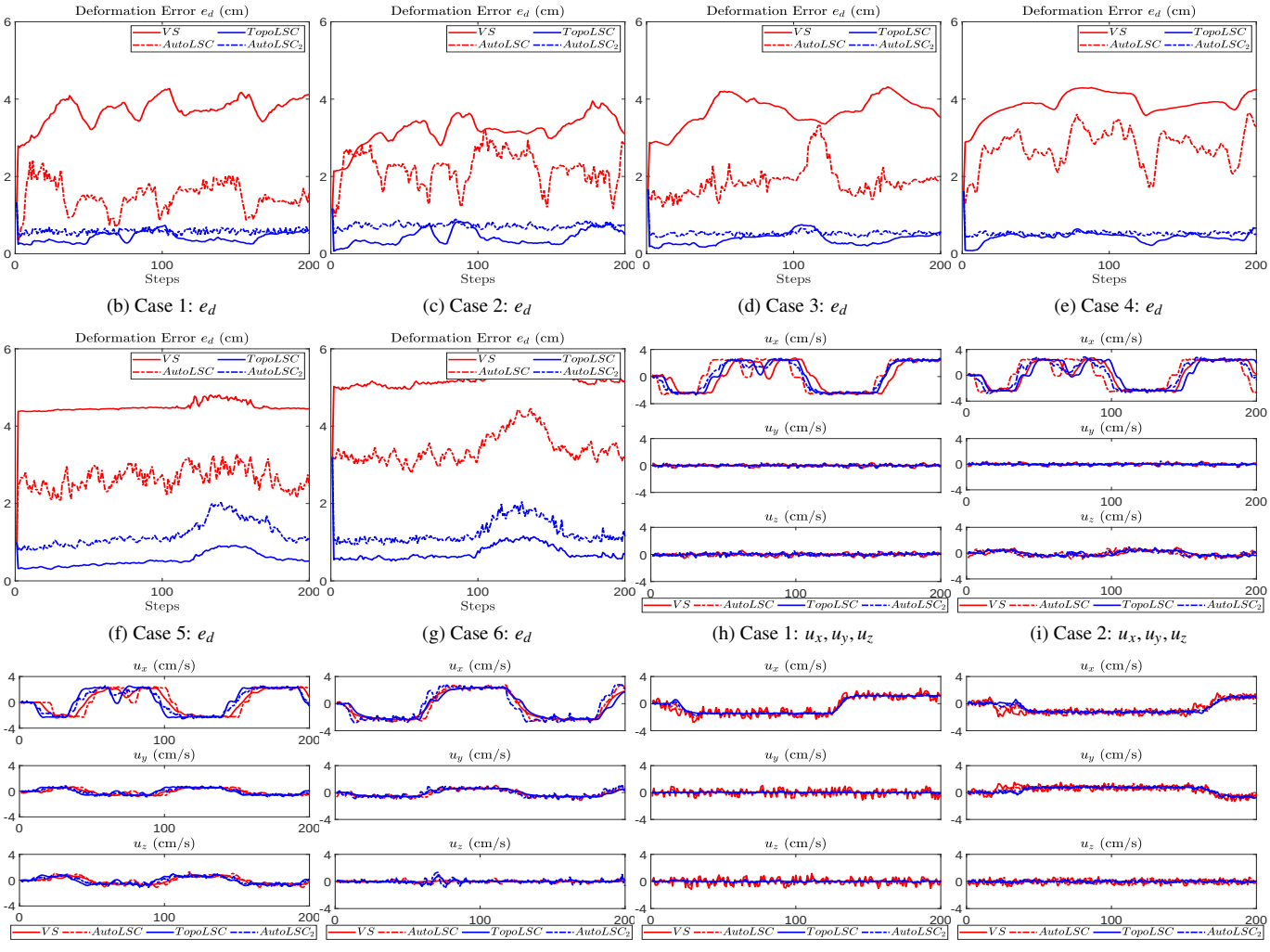
Furthermore, to evaluate the quality of latent representations, we also introduce another two metrics to measure the dimensionality reduction quality between input data and latent codes (as indicated by the ℓ in the abbreviations). Specifically, the first is called *trustworthiness* (ℓ-Trust), which evaluates the extent to which the k nearest neighbors of a point are conserved during the transition from the original space to the latent space. The second measure is called *continuity* (ℓ-Cont), which assesses the degree to which neighbors are maintained during the transition from the latent space to the original space. To enable a fair comparison, we set the same dimension ($n = 16$) of the latent space for different approaches. Experimental quantitative results can be found in Table 1, the TopoAE achieves the highest ℓ-Trust and lowest RMSE over all shape categories and shows a large improvement over other approaches. With respect to ℓ-Cont, the TopoAE presents a very competitive performance compared to the TSNE representations. As can be observed, the TopoAE not only can reconstruct the latent shapes back into the original shape space accurately but also can preserve the structural information on topological features in this built latent space.

6.4. Evaluation of Motor-robot Experiments

To quantitatively analyze the performance of our proposed sensorimotor model on reactive deformable object manipulation tasks, we program a stepping motor in a fixed trajectory to move the left side of the elastic cable. By doing so, we are able to produce the same interaction pattern for the deformable objects to imitate the human-robot collaboration, which is vital to fairly compare our proposed model with other advanced solutions. In these motor-robot experiments, three advanced approaches are selected to compare with our model, one is a traditional technique for adaptively deformable object manipulation using a model-free visual servoing [57] (referred as **VS** in the



(a) Qualitative experimental results in motor-robot experiments.



(b) Case 1: e_d (c) Case 2: e_d (d) Case 3: e_d (e) Case 4: e_d

(f) Case 5: e_d (g) Case 6: e_d (h) Case 1: u_x, u_y, u_z (i) Case 2: u_x, u_y, u_z

(j) Case 3: u_x, u_y, u_z (k) Case 4: u_x, u_y, u_z (l) Case 5: u_x, u_y, u_z (m) Case 6: u_x, u_y, u_z

Fig. 12: Qualitative and quantitative results in motor-robot experiments. (a) shows qualitative results of six motor-robot experiments (Vertical Track + $Arch_{-}^{\dagger}$, Vertical Tilt Track + $Arch_{-}^{\ddagger}$, Tilt Right Track + S_{-}, Left Track + *Line*, Vertical track + $Helix_{+}$, and Right track + $Helix_{+}$) by using three different approaches, where red arrows represent the motion direction of the motor (first rows), red, yellow, and green points represent the current, initial, and ground truth centerline points of the deformable linear object co-manipulated by the motor and robot, respectively. (b)-(g) and (h)-(m) show quantitative results of deformation error e_d and velocity curve along the x, y, and z-axis within the world frame in corresponding motor-robot experiments, respectively.

Table 3: Performance of Different Sensorimotor Models on Different Tasks for Motor-robot Experiments

Method	Shape Accuracy (cm)						Response Time (ms)					
	Case (a)	Case (b)	Case (c)	Case (d)	Case (e)	Case (f)	Case (a)	Case (b)	Case (c)	Case (d)	Case (e)	Case (f)
VS [57]	3.65 ± 0.07	3.15 ± 0.09	3.74 ± 0.06	3.90 ± 0.03	4.92 ± 0.08	5.25 ± 0.09	120 ± 52	197 ± 59	113 ± 63	124 ± 41	184 ± 68	192 ± 77
AutoLSC [58]	1.81 ± 0.73	2.21 ± 0.65	1.86 ± 0.59	2.76 ± 0.74	3.13 ± 1.07	3.26 ± 1.29	87 ± 34	82 ± 22	83 ± 28	77 ± 21	89 ± 33	93 ± 35
AutoLSC$_2$	0.68 ± 0.44	0.71 ± 0.36	0.67 ± 0. 46	0. 55 ± 0.33	1.05 ± 0.56	1.11 ± 0.62	55 ± 13	56 ± 20	60 ± 16	52 ± 12	57 ± 19	60 ± 18
TopoLSC (Ours)	0.44 ± 0.09	0.35 ± 0.06	0.42 ± 0.07	0.47 ± 0.08	0.56 ± 0.12	0.59 ± 0.11	47 ± 12	46 ± 8	43 ± 9	45 ± 13	49 ± 14	52 ± 17

following), and the other is a latent shape control (LSC) [58] model with naive auto-encoder (**AutoLSC**). To measure the performance of different approaches in the context of human-robot collaboration, we designed two metrics to measure and analyze the model performance: (1) shape accuracy during the entire human-robot collaboration process; (2) the response time of the manipulator starting to deform the shape after each leader's action. The shape accuracy is defined as the RMSE between the current shape c_i and its desired target shape c_i^* as below:

$$\text{RMSE}_{dom} = \left\| c_i - c_i^* \right\|^2 \tag{42}$$

where the desired target shape is computed based on the transformation $\mathbf{T}_{\boldsymbol{p}_0}^{\boldsymbol{p}_t}$ between the beginning leader pose $\boldsymbol{p}_0$ and current leader pose $\boldsymbol{p}_t$. As for the system response time, we intend to include both the time required for computing the desired latent shape vector and the control action using our method (computation time), and the time required for the control action to take effect (controller response time). We opted not to separate computation time from system response time in our assessments. This decision was made as our aim was not to provide an absolute measure of performance, but instead to facilitate a relative comparison among various methods under analogous conditions. By calibrating the transformation between the stepping motor device and the robot base, the current leader pose $\boldsymbol{p}_t$ is easy to obtain in the global coordinate system. Finally, the desired target shape is obtained as below:

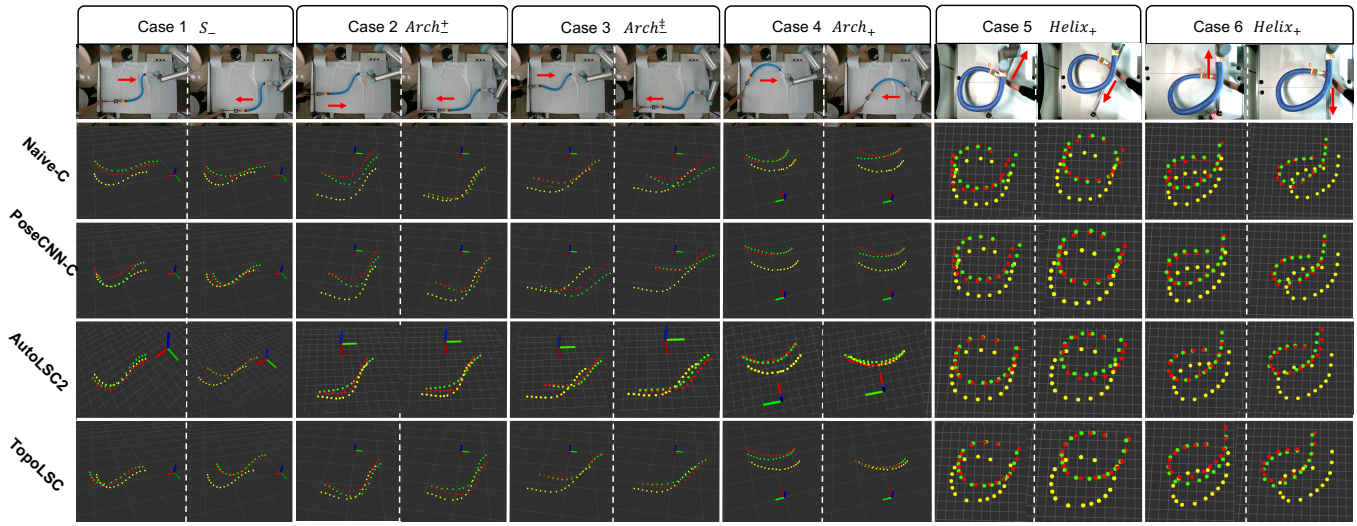
$$c_i^* = c_0 * \mathbf{T}_{\boldsymbol{p}_0}^{\boldsymbol{p}_t} \tag{43}$$

where c_0 is the beginning shape of the deformable object represented as the centerline points.

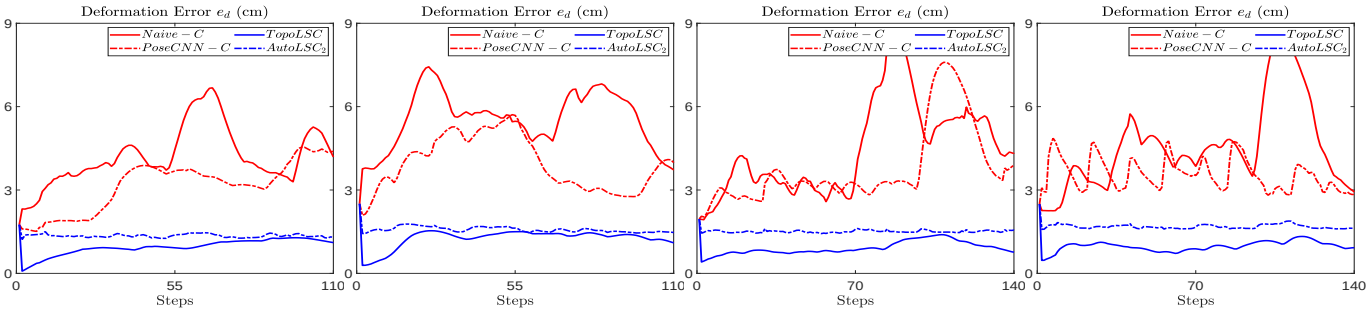
Table 3 and Fig. 12 show the quantitive and qualitative performance comparison of different sensorimotor models on various tasks for motor-robot experiments, respectively. The performance is measured in terms of Shape Accuracy (in *cm*) and Response Time (in *ms*) for six different cases, namely, 1) Vertical Track for shape $Arch^{\dagger}$; 2) Vertical Tilt Track for $Arch^{\ddagger}$; 3) Tilt Right Track for S, 4) Left Track for *Line*, 5) Vertical Track for $Helix_{\dagger}$ and 6) Vertical Track for $Helix$. We ran 10 experiments for each different experimental shape case. From the table, it is evident that our proposed method,**TopoLSC**, consistently outperforms other methods, including **VS**, **AutoLSC**, and **AutoLSC**$_2$, across all cases in terms of Shape Accuracy. In all cases, **TopoLSC** produced the lowest shape error, thereby indicating its superior precision in preserving the shape during manipulation tasks. For instance, in Case (a), the Shape Accuracy of **TopoLSC** was 0.44 ± 0.09 *cm*, compared to 3.65 ± 0.07 *cm*, 1.81 ± 0.73 *cm*, and 0.68 ± 0.44 *cm* for VS, AutoLSC, and AutoLSC$_2$, respectively. In terms of Response Time, TopoLSC also exhibits competitive performance. While the AutoLSC$_2$ method has the shortest response time in some cases, TopoLSC performs comparably well. For example, in Case (a), the response times for TopoLSC and AutoLSC$_2$ were 47 ± 12 ms and 55 ± 13 ms, respectively. In conclusion, the TopoLSC method outperforms other methods in terms of Shape Accuracy across all cases and shows competitive performance in Response Time. This indicates that our proposed method can effectively and efficiently manipulate deformable linear objects in motor-robot experiments, proving its efficacy and robustness. The results validate the advantage of using a topological latent shape representation in sensorimotor models for deformable object manipulation tasks.

6.5. Evaluation of Human-Robot Experiments

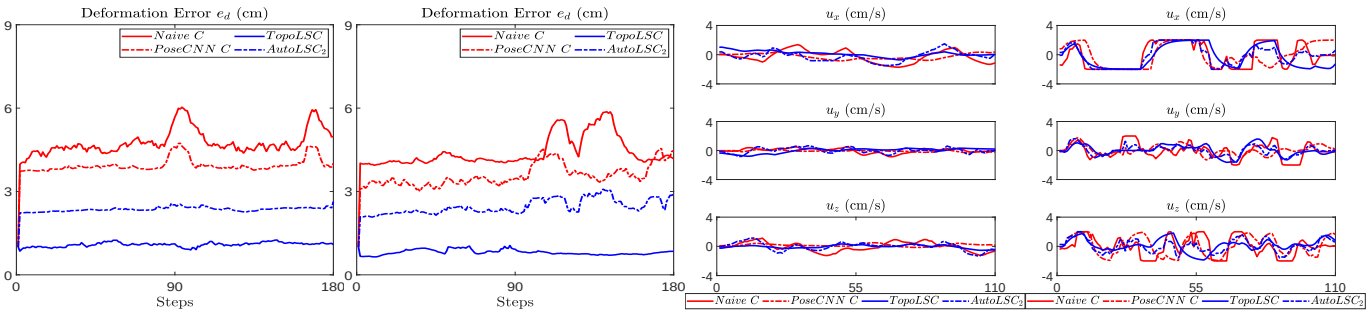
To further measure the overall performance of our proposed approach, a series of human-robot experiments are also conducted on various reactive shape manipulation tasks. Table 4 and Fig. 13 show the quantitative and qualitative performance comparison of different human-robot collaboration approaches (namely, **Naïve-C**, **PoseCNN-C**, and **TopoLSC**) on six reactive shape servoing tasks. Similarly, the performance is evaluated based on the same metrics: Shape Accuracy (measured in *cm*) and Response Time (measured in *ms*), we ran 10 trials for each task, and for each task, the Shape Accuracy and Response Time are given as a mean value plus/minus a standard deviation, which indicates the average performance and variability of each method for each task. Looking at the Shape Accuracy, we can see that the **TopoLSC** method outperforms the other methods in all six tasks, with the lowest average errors ranging from 0.72 cm to 1.23 cm. The **AutoLSC**$_2$ method comes in second, followed by the PoseCNN-C, and finally the Naïve-C method with the highest average shape accuracy errors. Although **AutoLSC**$_2$ demonstrates competitive performance on simpler tasks (Task 1-4), it underperforms compared to our **TopoLSC** approach on more complex helix shape servoing tasks (Task 5 and 6). This is primarily due to **AutoLSC**$_2$'s insufficient representation of the interconnected structure and overall shape. **Naïve-C**, in contrast, performs the worst out of all, as it only considers the leftmost k points to compute the transformation. This approach fails to provide a comprehensive understanding of the entire deformable object shape, leading to poorer performance. When comparing the performance of **TopoLSC** in human-robot interactions to its performance in motor-robot experiments, a relative decrease is observed. This is likely because human-robot interactions are more dynamic



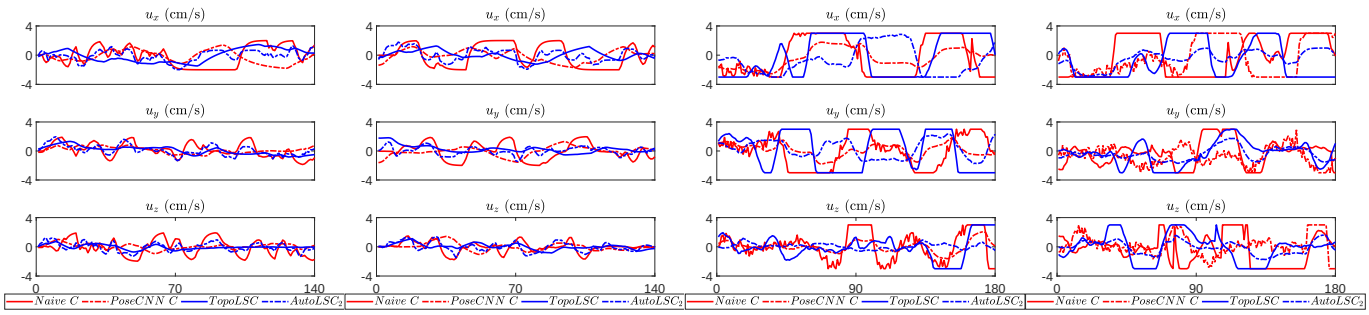
(a) Qualitative experimental results in human-robot experiments.



(b) Case 1: e_d (c) Case 2: e_d (d) Case 3: e_d (e) Case 4: e_d



(f) Case 5: e_d (g) Case 6: e_d (h) Case 1: u_x, u_y, u_z (i) Case 2: u_x, u_y, u_z



(j) Case 3: u_x, u_y, u_z (k) Case 4: u_x, u_y, u_z (l) Case 5: u_x, u_y, u_z (m) Case 6: u_x, u_y, u_z

Fig. 13: Qualitative and quantitative results in human-robot experiments. (a) shows qualitative results of six human-robot experiments (S_-, $Arch^{\dagger}_{-}$, $Arch^{\ddagger}_{-}$, $Arch_+$, $Helix_+$, and $Helix_+$) by using different overall frameworks, where red arrows represent the motion direction of motor (first rows), red, yellow and green points represent the current, initial, and ground truth centerline points of the deformable linear object co-manipulated by the human hand and robot, respectively. (b)-(g) and (h)-(m) show quantitative results of deformation error e_d and velocity curve along the x, y, and z-axis within the world frame in corresponding human-robot experiments, respectively.

Table 4: Performance of Different Frameworks on Different Human-robot Experiments.

Method	Shape Accuracy (cm)						Response Time (ms)					
	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6
Naïve-C	4.26 ± 2.41	5.52 ± 2.99	4.21 ± 2.69	4.40 ± 2.43	4.68 ± 2.27	4.39 ± 2.56	237 ± 83	259 ± 108	227 ± 87	246 ± 93	237 ± 97	241 ± 102
PoseCNN-C	3.37 ± 1.21	3.78 ± 1.94	3.68 ± 2.02	3.52 ± 2.38	3.94 ± 2.83	3.81 ± 2.49	32 ± 11	33 ± 9	18 ± 13	17 ± 14	28 ± 14	31 ± 12
AutoLSC$_2$	1.33 ± 0.17	1.54 ± 0.22	1.49 ± 0.17	1.65 ± 0.18	2.37 ± 0.58	2.50 ± 0.49	45 ± 11	51 ± 14	51 ± 9	52 ± 12	59 ± 14	58 ± 11
TopoLSC (Ours)	0.72 ± 0.10	1.23 ± 0.17	0.90 ± 0.12	0.95 ± 0.19	1.09 ± 0.24	0.98 ± 0.16	44 ± 10	49 ± 13	51 ± 12	47 ± 12	53 ± 10	58 ± 17

and unpredictable, whereas motor-robot experiments tend to offer more stable interactions. Similarly, in terms of Response Time, the PoseCNN-C method has the fastest response times across all tasks, with average times ranging from 17 ms to 33 ms. The **TopoLSC** and **AutoLSC**$_2$ methods have comparable response times, both significantly faster than the Naïve-C method, which has the longest response times, ranging from 237 ms to 259 ms.

In conclusion, the **TopoLSC** method offers the best balance between shape accuracy and response time, making it the most efficient method for these human-robot interaction tasks. Although the **PoseCNN-C** method has faster response times, its shape accuracy is not as good as the **TopoLSC** or **AutoLSC**$_2$ methods. Conversely, while the **Naïve-C** method is less efficient in both metrics, it might still be useful in scenarios where computational resources or time are not limiting factors. As for the ablation study, from the above two tables, it can be observed that **AutoLSC**$_2$ has a performance similar to our proposed approach in terms of Response Time, as both utilize the same fixed-time sliding mode controller for DOM tasks in HRC. However, in terms of Shape Accuracy, our method consistently outperforms **AutoLSC**$_2$. Although the advantage is minor in the motor-robot experiments, the Shape Accuracy improvement is significant in human-robot experiments. This can be attributed to the more dynamic and unpredictable nature of human-robot interactions, where our **TopoLSC** better handles the unpredictability due to its use of a topological representation. This representation captures the connectivity and structure of the objects' shapes more effectively. Specifically, in more complex shape servoing tasks of helix shape categories in human-robot experiments (Tasks 5 and 6), our method maintains a stable and high shape accuracy, indicating superior generalization ability and performance when dealing with complex deformable shapes.

6.6. Limitations

While the proposed human-robot collaboration approach for reactive deformable linear object manipulation tasks using topological latent control models shows promise, there are several limitations that need to be considered. Firstly, the effectiveness of the proposed approach is highly dependent on the accuracy of the perception system. Any inaccuracies or delays in the perception system can lead to incorrect control signals being generated, which can result in suboptimal manipulation performance. Secondly, the proposed approach is specifically tailored for the manipulation of deformable linear objects and does not inherently consider the morphing of shapes from one class to another. Notably, our current system and study are primarily focused on a visual shape servoing task. Our present robotic system may not be equipped to seamlessly handle such a transition, thus limiting the scope of its application. Thirdly, while our approach is effective for certain types of objects, it may not be suitable for all deformable object manipulation applications. For instance, in surgical robotics, the manipulation of soft tissue may necessitate a different perception approach. Another layer of complexity is added through the use of human-robot collaboration. This introduces additional challenges, such as the need for effective communication and coordination between the human operator and the robot. Moreover, the system may be sensitive to differences in human expertise, which may affect the quality of the manipulation performance. In summary, although our approach has demonstrated potential for achieving real-time reactive manipulation of deformable linear objects through the use of topological latent control models and human-robot collaboration, further research is required. This should aim to address the identified limitations and explore the applicability of our method to other types of deformable objects and a broader range of real-world scenarios.

It's important to note that in the human-robot collaboration literature, the focus is predominantly on the manipulation of rigid objects. This allows for easier modeling of the relationship between human movements and their effects on manipulated objects. However, when dealing with deformable objects like in our case, the task becomes significantly more challenging due to the infinite degrees of freedom and complex dynamics associated with these objects. As a result, it becomes difficult to formulate a precise model of such behavior. In our work, the robotic system is designed to focus more directly on the perception and manipulation of the deformable object. This focus, while necessary for the success of our tasks, may have inadvertently caused the human aspect to appear less prominent in our study. While our current study was limited in this respect, we plan to include a diverse range of human subjects in our future work.

7. Conclusion

In conclusion, this article presents an innovative approach to address the challenge of deformable object manipulation in human-robot collaboration scenarios. The proposed Topological Latent Control Model (TopoLSC) enables the robot to learn a low-dimensional representation of the deformable object, allowing the controller to reactively adapt its manipulation strategy in real time based on the human partner's behavior. The experimental results demonstrate the effectiveness of the proposed approach in achieving accurate and efficient manipulation of

deformable linear objects while maintaining high shape accuracy and low response time between the human and the robot. We also provide a comprehensive analysis of the system's performance and robustness under different scenarios and conditions. Overall, this paper provides a significant contribution to the field of human-robot collaboration, especially in the domain of deformable object manipulation. The proposed approach has the potential to enable more complex and versatile collaborative tasks between humans and robots, where the robots can learn to reactively manipulate an object based on the human partner's actions and adapt their behavior accordingly. Future research could concentrate on broadening the application of this approach to encompass a wider variety of deformable objects and human movements. Additionally, the exploration of its potential in applying deformable object manipulation techniques in more human-robot collaboration tasks could prove fruitful. Importantly, a key focus should be on assessing the viability and impact of this approach within real-world scenarios.

References

[1] S. Li, P. Zheng, S. Liu, Z. Wang, X. V. Wang, L. Zheng, L. Wang, Proactive human–robot collaboration: Mutual-cognitive, predictable, and self-organising perspectives, Robotics and Computer-Integrated Manufacturing 81 (2023) 102510.

[2] P. Zheng, S. Li, L. Xia, L. Wang, A. Nassehi, A visual reasoning-based approach for mutual-cognitive human-robot collaboration, CIRP annals 71 (1) (2022) 377–380.

[3] S. Li, R. Wang, P. Zheng, L. Wang, Towards proactive human–robot collaboration: A foreseeable cognitive manufacturing paradigm, Journal of Manufacturing Systems 60 (2021) 547–552.

[4] L. Wang, X. V. Wang, J. Váncza, Z. Kemény, Advanced Human-Robot Collaboration in Manufacturing, Springer, 2021.

[5] F. Cini, T. Banfi, G. Ciuti, L. Craighero, M. Controzzi, The relevance of signal timing in human-robot collaborative manipulation, Science Robotics 6 (58) (2021) eabg1308.

[6] P. Zhou, P. Zheng, J. Qi, C. Li, A. Duan, M. Xu, V. Wu, D. Navarro-Alarcon, Neural reactive path planning with riemannian motion policies for robotic silicone sealing, Robotics and Computer-Integrated Manufacturing 81 (2023) 102518.

[7] P. Zhou, R. Peng, M. Xu, V. Wu, D. Navarro-Alarcon, Path planning with automatic seam extraction over point cloud models for robotic arc welding, IEEE Robotics and Automation Letters 6 (3) (2021) 5002–5009.

[8] H.-Y. Lee, P. Zhou, A. Duan, J. Wang, V. Wu, D. Navarro-Alarcon, A multisensor interface to improve the learning experience in arc welding training tasks, IEEE Transactions on Human-Machine Systems (2023).

[9] H. Su, C. Yang, G. Ferrigno, E. De Momi, Improved human–robot collaborative control of redundant robot for teleoperated minimally invasive surgery, IEEE Robotics and Automation Letters 4 (2) (2019) 1447–1453.

[10] V. V. Unhelkar, S. Li, J. A. Shah, Semi-supervised learning of decision-making models for human-robot collaboration, in: Conference on Robot Learning, PMLR, 2020, pp. 192–203.

[11] J. Mainprice, D. Berenson, Human-robot collaborative manipulation planning using early prediction of human motion, in: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2013, pp. 299–306.

[12] H. Ha, S. Song, Flingbot: The unreasonable effectiveness of dynamic manipulation for cloth unfolding, in: Conference on Robot Learning, PMLR, 2022, pp. 24–33.

[13] J. Zhu, B. Navarro, R. Passama, P. Fraisse, A. Crosnier, A. Cherubini, Robotic manipulation planning for shaping deformable linear objects withenvironmental contacts, IEEE Robotics and Automation Letters 5 (1) (2019) 16–23.

[14] J. Qi, G. Ma, J. Zhu, P. Zhou, Y. Lyu, H. Zhang, D. Navarro-Alarcon, Contour moments based manipulation of composite rigid-deformable objects with finite time model estimation and shape/position control, IEEE/ASME Transactions on Mechatronics (2021).

[15] D. Navarro-Alarcon, Y.-H. Liu, Fourier-based shape servoing: A new feedback method to actively deform soft objects into desired 2-d image contours, IEEE Transactions on Robotics 34 (1) (2017) 272–279.

[16] H. Yin, A. Varava, D. Kragic, Modeling, learning, perception, and control methods for deformable object manipulation, Science Robotics 6 (54) (2021) eabd8803.

[17] J. Matas, S. James, A. J. Davison, Sim-to-real reinforcement learning for deformable object manipulation, in: Conference on Robot Learning, PMLR, 2018, pp. 734–743.

[18] A. M. Howard, G. A. Bekey, Intelligent learning for deformable object manipulation, Autonomous Robots 9 (2000) 51–58.

[19] X. Lin, Y. Wang, J. Olkin, D. Held, Softgym: Benchmarking deep reinforcement learning for deformable object manipulation, in: Conference on Robot Learning, PMLR, 2021, pp. 432–448.

[20] D. Kruse, R. J. Radke, J. T. Wen, Collaborative human-robot manipulation of highly deformable materials, in: 2015 IEEE international conference on robotics and automation (ICRA), IEEE, 2015, pp. 3782–3787.

[21] P. Zhou, J. Zhu, S. Huo, D. Navarro-Alarcon, Lasesom: A latent and semantic representation framework for soft object manipulation, IEEE Robotics and Automation Letters 6 (3) (2021) 5381–5388.

[22] M. Aranda, J. A. C. Ramon, Y. Mezouar, A. Bartoli, E. Özgür, Monocular visual shape tracking and servoing for isometrically deforming objects, in: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2020, pp. 7542–7549.

[23] M. H. D. Zakaria, M. Aranda, L. Lequièvre, S. Lengagne, J. A. C. Ramón, Y. Mezouar, Robotic control of the deformation of soft linear objects using deep reinforcement learning, in: 2022 IEEE 18th International Conference on Automation Science and Engineering (CASE), IEEE, 2022, pp. 1516–1522.

[24] D. Sirintuna, A. Giammarino, A. Ajoudani, Human-robot collaborative carrying of objects with unknown deformation characteristics, in: 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2022, pp. 10681–10687.

[25] F. Iori, G. Perovic, F. Cini, A. Mazzeo, E. Falotico, M. Controzzi, Dmp-based reactive robot-to-human handover in perturbed scenarios, International Journal of Social Robotics 15 (2) (2023) 233–248.

[26] X. Ma, D. Hsu, W. S. Lee, Learning latent graph dynamics for deformable object manipulation, arXiv preprint arXiv:2104.12149 (2021).

[27] A. Khalifa, G. Palli, New model-based manipulation technique for reshaping deformable linear objects, The International Journal of Advanced Manufacturing Technology (2021) 1–9.

[28] D. Berenson, Manipulation of deformable objects without modeling and simulating deformation, in: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2013, pp. 4525–4532.

[29] D. McConachie, M. Ruan, D. Berenson, Interleaving planning and control for deformable object manipulation, in: Robotics Research: The 18th International Symposium ISRR, Springer, 2019, pp. 1019–1036.

[30] P. Jiménez, Survey on model-based manipulation planning of deformable objects, Robotics and computer-integrated manufacturing 28 (2) (2012) 154–163.

[31] A. Caporali, K. Galassi, R. Zanella, G. Palli, Fastdlo: Fast deformable linear objects instance segmentation, IEEE Robotics and Automation Letters 7 (4) (2022) 9075–9082.

[32] A. Caporali, M. Pantano, L. Janisch, D. Regulin, G. Palli, D. Lee, A weakly supervised semi-automatic image labeling approach for deformable linear objects, IEEE Robotics and Automation Letters 8 (2) (2023) 1013–1020.

[33] D. McConachie, D. Berenson, Estimating model utility for deformable object manipulation using multiarmed bandit methods, IEEE Transactions on Automation Science and Engineering 15 (3) (2018) 967–979.

[34] E. Matheson, R. Minto, E. G. Zampieri, M. Faccio, G. Rosati, Human–robot collaboration in manufacturing applications: A review, Robotics 8 (4) (2019) 100.

[35] G. Hoffman, Evaluating fluency in human–robot collaboration, IEEE Transactions on Human-Machine Systems 49 (3) (2019) 209–218.

[36] D. Seita, P. Florence, J. Tompson, E. Coumans, V. Sindhwani, K. Goldberg, A. Zeng, Learning to rearrange deformable cables, fabrics, and bags with goal-conditioned transporter networks, in: 2021 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2021, pp. 4568–

4575.
[37] D. Andronas, E. Kampourakis, K. Bakopoulou, C. Gkournelos, P. Angelakis, S. Makris, Model-based robot control for human-robot flexible material co-manipulation, in: 2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), IEEE, 2021, pp. 1–8.
[38] S. Makris, E. Kampourakis, D. Andronas, On deformable object handling: Model-based motion planning for human-robot co-manipulation, CIRP Annals 71 (1) (2022) 29–32.
[39] Z. Wang, A. H. Qureshi, Deri-bot: Learning to collaboratively manipulate rigid objects via deformable objects, arXiv preprint arXiv:2305.13183 (2023).
[40] H. Edelsbrunner, J. Harer, et al., Persistent homology-a survey, Contemporary mathematics 453 (26) (2008) 257–282.
[41] N. Otter, M. A. Porter, U. Tillmann, P. Grindrod, H. A. Harrington, A roadmap for the computation of persistent homology, EPJ Data Science 6 (2017) 1–38.
[42] A. Zomorodian, Fast construction of the vietoris-rips complex, Computers & Graphics 34 (3) (2010) 263–271.
[43] H. Edelsbrunner, D. Letscher, A. Zomorodian, Topological persistence and simplification, in: Proceedings 41st annual symposium on foundations of computer science, IEEE, 2000, pp. 454–463.
[44] T. Bretl, Z. McCarthy, Quasi-static manipulation of a kirchhoff elastic rod based on a geometric analysis of equilibrium configurations, The International Journal of Robotics Research 33 (1) (2014) 48–68.
[45] T. Tang, M. Tomizuka, Track deformable objects from point clouds with structure preserved registration, The International Journal of Robotics Research 41 (6) (2022) 599–614.
[46] A. P. Dempster, N. M. Laird, D. B. Rubin, Maximum likelihood from incomplete data via the em algorithm, Journal of the Royal Statistical Society: Series B (Methodological) 39 (1) (1977) 1–22.
[47] M. Kuczma, An introduction to the theory of functional equations and inequalities, PWN, Warszawa, Kraków, Katowice (1985).
[48] X. Da, J. Grizzle, Combining trajectory optimization, supervised machine learning, and model structure for mitigating the curse of dimensionality in the control of bipedal robots, The International Journal of Robotics Research 38 (9) (2019) 1063–1097.
[49] M. Moor, M. Horn, B. Rieck, K. Borgwardt, Topological autoencoders, in: International conference on machine learning, PMLR, 2020, pp. 7045–7054.
[50] D. Cohen-Steiner, H. Edelsbrunner, J. Harer, Stability of persistence diagrams, in: Proceedings of the twenty-first annual symposium on Computational geometry, 2005, pp. 263–271.
[51] M. Ou, H. Sun, Z. Zhang, S. Gu, Fixed-time trajectory tracking control for nonholonomic mobile robot based on visual servoing, Nonlinear Dynamics (2022) 1–13.
[52] H. Yang, D. Ye, Adaptive fixed-time bipartite tracking consensus control for unknown nonlinear multi-agent systems: An information classification mechanism, Information Sciences 459 (2018) 238–254.
[53] J. Ma, H. Wang, J. Qiao, Adaptive neural fixed-time tracking control for high-order nonlinear systems, IEEE Transactions on Neural Networks and Learning Systems (2022).
[54] M. Van, M. Mavrovouniotis, S. S. Ge, An adaptive backstepping nonsingular fast terminal sliding mode control for robust fault tolerant control of robot manipulators, IEEE Transactions on Systems, Man, and Cybernetics: Systems 49 (7) (2018) 1448–1458.
[55] J. Stoer, R. Bulirsch, Introduction to numerical analysis, Vol. 12, Springer Science & Business Media, 2013.
[56] A. Levant, Higher-order sliding modes, differentiation and output-feedback control, International journal of Control 76 (9-10) (2003) 924–941.
[57] R. Lagneau, A. Krupa, M. Marchal, Automatic shape control of deformable wires based on model-free visual servoing, IEEE Robotics and Automation Letters 5 (4) (2020) 5252–5259.
[58] J. Qi, G. Ma, P. Zhou, H. Zhang, Y. Lyu, D. Navarro-Alarcon, Towards latent space based manipulation of elastic rods using autoencoder models and robust centerline extractions, Advanced Robotics 36 (3) (2022) 101–115.