Human-Robot Collaboration for Reactive Deformable Linear Object Manipulation Using Topological Latent Control Model

Peng Zhou\textsuperscript{a,c,\ast}, Pai Zheng\textsuperscript{b}, Jiaming Qi\textsuperscript{b}, Chengxi Li\textsuperscript{b}, Hoi-Yin Lee\textsuperscript{c}, Anqing Duan\textsuperscript{a}, Liang Lu\textsuperscript{c}, Zhongxuan Li\textsuperscript{c}, Luyin Hu\textsuperscript{c}, David Navarro-Alarcon\textsuperscript{a}

\textsuperscript{a}Department of Mechanical Engineering, The Hong Kong Polytechnic University, KLN, Hong Kong.
\textsuperscript{b}Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, KLN, Hong Kong.
\textsuperscript{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. This proposed topological control model provides a framework for controlling the dynamic shape of deformable objects, with its latent representation providing an efficient characterization of the object’s shape. 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) research and development, with the aim of achieving more efficient and effective collaboration between humans and robots in various domains, such as manufacturing \textsuperscript{15}, construction \textsuperscript{16} \textsuperscript{17}, healthcare \textsuperscript{18}, and service industries \textsuperscript{19}. One area of HRC that has received a lot of attention is the manipulation of objects \textsuperscript{10} \textsuperscript{11}. 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 \textsuperscript{12}, wires \textsuperscript{13}, cables \textsuperscript{14}, and soft tissues \textsuperscript{15}. Manipulating deformable objects is more challenging than manipulating rigid objects because they have complex and nonlinear behaviors \textsuperscript{16} \textsuperscript{17} \textsuperscript{18} \textsuperscript{19}. Despite these challenges, there are many potential applications \textsuperscript{20} \textsuperscript{21} 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. \textsuperscript{22}

The manipulation of deformable objects in the context of HRC has not been sufficiently studied; Most of the existing HRC research has focused on rigid object manipulation. Furthermore, existing approaches \textsuperscript{23} \textsuperscript{24} for the manipulation of deformable objects typically rely on analytical or numerical models \textsuperscript{25} \textsuperscript{26} 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 \textsuperscript{27}.

Therefore, this gap is a significant challenge that needs to be addressed to fully realize the potential of DOM in various HRC applications \textsuperscript{27}.

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 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 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. The topological latent representation allows for efficient representation of the object’s shape and topology based on feedback data, which is critical for real-time control. The application of 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.

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](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 unmodelled human and robot 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 presents 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) 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 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, Kruse et al 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 have focused on learning-based approaches for DOM, such as DeformableRavens, 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.

In this paper, we propose a novel method for reactive deformable linear object manipulation under human-robot collaboration using a topological latent control model. Our method can handle complex deformable linear objects and can adapt to the human partner’s actions in real time. Our method also leverages a topological latent space to represent the deformation state of the object and to generate appropriate control actions for the robot. We evaluate our method on several collaborative DOM tasks and demonstrate its advantages over existing methods in terms of deformation error, task response time, and robustness to human interventions. To the best of our knowledge, this is the first work that combines topological modeling and latent control for collaborative DOM.

3. Preliminaries

In computational topology, the method used for analyzing topological features of data across multiple scales is called persistent homology. 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
Fig. 2: (a)-(c) show at varying scales $\epsilon_0$, $\epsilon_1$, and $\epsilon_2$, the Vietoris-Rips complex $V_{\epsilon}(q)$ of a point cloud $q$ changes its connectivity as the distance threshold $\epsilon$ is increased. (d) presents the $d$-th persistence diagram $G_d$ captures the emergence and disappearance of $d$-dimensional topological features.

Fig. 3: Conceptual representation of human-robot collaboration for reactive deformable linear object manipulation. The proposed framework is composed of three components, namely, a Gaussian Mixture Model (GMM)-based state estimator for deformable linear object (top left), a latent shape space built upon topological loss using the topological auto-encoder (bottom left), and a fixed-time sliding model-based controller for reactive controls on the manipulated linear object (bottom right). Given an initial deformable linear shape $s_0$ and current shape $s_i$, the target is to deform the desired shape $s_d$ into a similar shape with $s_0$ in a reactive manner. Beginning with the state estimator, the original shapes perceived by point clouds $\{q_i\}$ are able to be represented by a set of centerline points $\{c_i\}$. Subsequently, by combining reconstruction and topological loss, corresponding latent shapes $\{z_i\}$ are generated. Finally, a fixed-time sliding model is designed as the controller in latent shape space to command the robot agent to complete the reactive shape servoing during human-robot collaborations.
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 $\mathcal{Q} = \{q_1, \ldots, q_n\} \subseteq \mathbb{R}^3$ and a distance metric: $\mathcal{Q} \times \mathcal{Q} \rightarrow \mathbb{R}$ (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 \cite{44}. Let $\mathcal{V}(\mathcal{Q})$ be the Vietoris-Rips complex of $\mathcal{Q}$ with a scale $\epsilon$, and it has all simplices of the point cloud $\mathcal{Q}$ whose elements satisfy a distance criterion, namely $\text{dist}(q_i, q_j) \leq \epsilon$ for all $i, j$. As the Vietoris-Rips complex provides a nesting structure, $\mathcal{V}_\epsilon(\mathcal{Q}) \subseteq \mathcal{V}_{\epsilon'}(\mathcal{Q})$ when $\epsilon_i \leq \epsilon_j$, it becomes possible to trace alterations in the homology groups when $\epsilon$ increases \cite{35}.

Let $\mathcal{PH}(\mathcal{V}_\epsilon(\mathcal{Q}))$ represent the persistent homology of the point cloud $\mathcal{Q}$’s Vietoris-Rips complex, and it results in a tuple $((G_1, G_2, \ldots, G_i), \phi)$, and $\phi$ denote the persistence diagrams and persistence pairings, respectively. In $d$-dimensional persistence, we define a tuple of $(a, b)$, where $a$ denotes a scale $\epsilon$ at which a $d$-dimensional topological feature emerges, and $b$ represents another scale $\epsilon'$ at which it disappears. The $d$-dimensional persistence consists of pairs of indices denoted as $i, j$ that correspond to simplices $s_i, s_j \in \mathcal{V}(\mathcal{Q})$ 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: $d_b(\mathcal{G}, \mathcal{G}') := \inf_{\eta : \mathcal{G} \rightarrow \mathcal{G}'} \sup_{x \in \mathcal{G}} \|x - \eta(x)\|_\infty$, where $\eta : \mathcal{G} \rightarrow \mathcal{G}'$ is defined as a bijection between the diagram $\mathcal{G}$ and $\mathcal{G}'$, and $\|\cdot\|_\infty$ refers to the $L_\infty$ norm. Finally, we define $\mathcal{G}^\epsilon$ as the set of persistence diagrams for the point cloud $\mathcal{Q}$, which can be obtained from the computation of $\mathcal{PH}(\mathcal{V}(\mathcal{Q}))$.

4. Problem Formulation

In this article, we propose a novel human-robot collaboration system consisting of a robot manipulator, a human hand, and a deformable linear object, as depicted in Fig. 4. 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 between the object and the human hand, respectively. The human hand applies force on the one side of the deformable linear object and leads 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 “follower” (see Fig. 4 for details).

As shown in Fig. 3 a classic deformable object shape servants 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 is leading the deformation first, then the robot controller follows the human action and manipulate the object into the initial shape. Therefore, we defined the human partner as the leader role, the robot controller as follower role in this human-robot collaboration.

Assumptions 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 $q \in \mathbb{R}^6$, and the end-effector pose (3-DOF position and 3-DOF orientation) as $x \in \mathbb{R}^6$, respectively. According to the classical kinematic equation of the manipulator, the differential relationship between $q$ and $x$ is given as follows:

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

where the matrix $\frac{\partial x}{\partial q}(q) \in \mathbb{R}^{6 \times 6}$ 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 paper, the speed control signal of the end-effector is designed, and by using \cite{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.

Assumptions 2. The deformation Jacobian matrix (DJM) is able to describe the kinematic relationship between the robot and the manipulated deformable linear object.
In this paper, a depth camera within eye-to-hand configuration to observe the shape of the elastic cable. For simplicity, we use the commonly used centerline to represent the object’s shape, with the following definitions:

\[ s = [c_1, \ldots, c_N]^T \in \mathbb{R}^{3N} \]  

(2)

where \( N \) denotes the number of total centerline points constituting the centerline 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.

The pose \( x \) of the end-effector would take an effect on the shape \( s \) of the elastic cable, which can be represented by the unknown nonlinear function, i.e., \( s = f_s(x) \). However, the large dimension of the original shape \( s \) is \( 3N \), 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 \( z \in \mathbb{R}^d (k \ll 3N) \) to represent \( s \), which characterizes the original shape \( s \) but with significantly fewer-dimensional feedback vector. Theoretically, the feature \( z \) has a one-to-one mapping relationship with \( s \), i.e., \( z = f_z(s) \). Thus, the latent shape feature \( z \) can be obtained as below:

\[ z = f_z(s) = f_z(f_s(x)) \]  

(3)

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

\[ \dot{z} = \frac{\partial f_z}{\partial x} \dot{x} = J_z(x) \dot{x} \]  

(4)

where \( J_z(x) \) 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, thus 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.

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

**Assumptions 4.** A bound exists for the approximation error \( \tilde{J} \), \( \|\tilde{J}\|^2 \leq \eta \), for \( \eta \) 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. 3, 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 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.

Given an initial deformable linear shape \( s_0 \), let the current shape be \( s_t \) after a series of human-robot interactions and 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 \( s_d \), which is regarded as a similar shape with \( s_0 \). Meanwhile, the entire process is performed in a reactive manner. Beginning with the state estimator, the original shapes perceived by point clouds \( \{q_i\} \) are able to be represented by a set of centerline points \( \{c_i\} \). Subsequently, by combining the reconstruction loss \( \mathcal{L}_{rec} \) and topological loss \( \mathcal{L}_{topo} \), a topological-aware auto-encoder \( f_e: C \rightarrow Z \) is trained to encode the centerline-based shapes \( \{c_i\} \) from originally high-dimensional shape space \( C \) into a low-dimensional latent shape space \( Z \). With this built latent shape space, a deep neural network-based latent shape predictor is used to output the desired latent shape \( z_d \), together with \( z_l \), that are regarded as the inputs for the designed fixed-time sliding model to command the robot agent to complete the reactive shape servoing task during human-robot collaborations in a high-efficient and effective manner.

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 \( s(t) \) at time step \( t \) as a dense, noisy, and occluded point cloud \( q(t) = q_1, q_2, \ldots, q_M \in \mathbb{R}^{M \times D} \), 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 \( C = \{c_1', c_2', \ldots, c_N'\} \in \mathbb{R}^{N \times 3} \) at time step \( t \), where \( c_i' \in \mathbb{R}^{1 \times 3} \) represents the 3D coordinate of the \( i \)-th key point at time step \( t \). Structure preservation registration (SPR) [35] consider that the perceived point clouds \( q_i \) are sampled \( c_i \) from a Gaussian Mixture Model (GMM), and centroids of the point cloud represent the key points of the deformable linear object shape \( s_i \). Based on Bayes’ theorem, the probability of a point \( q_m \) sampled from the mixture model can be defined as below:

\[ p(q_m) = \sum_{n=1}^{N+1} p(n)p(q_m | n) \]  

(5)

where \( p(n) \) denote the weight of the \( n \)-th mixture component, and \( p(q_m | n) \) denote the probability of sampling \( q_m \) 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:

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

(6)

\[ p(q_m \mid n) = \begin{cases} N \left( q_m, c_n^0, \sigma^2 I \right), & n = 1, \ldots, N \\ \frac{1}{M}, & n = N + 1 \end{cases} \]  

(7)

The main objective is to maximize the log likelihood \( L \) sampled from the point cloud \( q_c \), which can be formulated as a problem of Maximum Likelihood Estimation (MLE). The optimization of mixture centroids for maximizing the log likelihood function \( L \) is non-convex due to the summation inside \( \log(\cdot) \), making direct optimization infeasible. Thus, we construct another log-likelihood function \( O \) having a lower bound of \( L \). The maximization of \( O \) through the EM algorithm \[37\] involves the E-step (expectation step) and M-step (maximization step), that iteratively estimate \( (c_n, \sigma^2) \) by maximizing \( O \). The formula for \( O \) is given as:

\[ O(c_n, \sigma^2) = \sum_{n=1}^{N} \sum_{m=1}^{M} p(n \mid q_m) \log \left( p(n)p(q_m \mid n) \right) \]  

(8)

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 \( O \) is aimed to increase the value of \( L \) except at local optima, and The use of Jensen’s inequality \[38\] can demonstrate that function \( O \) serves as a lower bound for function \( L \). As a result, elevating the value of \( O \) will inevitably lead to an increase in the value of \( L \) unless it has already reached a local optimum. By comparing the structures of \( O \) and \( L \), we see that the summation inside the logarithm in \( L \) has been moved to the front in \( O \), providing computational convenience. The EM algorithm \[37\] can be used with the definition of the complete log-likelihood function to iteratively estimate \( (c_n, \sigma^2) \) by maximizing \( O \) through the E-step and M-step. By utilizing the definition of the complete log-likelihood function, the EM algorithm \[37\] can be employed to iteratively estimate \( (c_n, \sigma^2) \) by maximizing \( O \) through E-step and M-step.

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. Additionally, the centerline-based shape representation is not able to directly feed into a controller since considering its high dimensionality it will make the reactive robotic system run inefficiently and even lose the control of the robot. 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, we propose a generic approach which applies topological auto-encoders \[39\] for constraining the encoding network to preserve topological structures represented by persistent homology in the generated latent shape space. Fig. 5 (also see the second component in Fig. 3) depicts an overview of our method, and we divide this learning process into three individual steps in the following.

5.2.1. Vietoris-Rips Complex Calculation

To begin, we employ the distance matrix \( D^S \) to compute the persistent homology of the Vietoris-Rips complex of a metric space \( S \), which can be represented by centerline points. In this work, we choose to use the Euclidean distance for the calculation of \( D^S \), but other distances can be used as well. We then determine \( \epsilon := \max D^S \) and construct the corresponding Vietoris-Rips complex, denoted by \( \text{Ve}(D^S) \). For a dimension \( d \in \mathbb{N} > 0 \), a set of persistence diagrams \( G^S \) and a set of persistence pairings \( \phi^S \) can be obtained. The persistence pairing \( \phi^S \) 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.

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 \( \phi^0 \). 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 \( S \) as \( (G^S, \phi^S) \).

5.2.3. Topological Autoencoder

We start by considering a mini-batch \( \epsilon \) consisting of \( l \) points from the shape data space \( C \) (i.e., a point cloud). We then construct an autoencoder using a composite function \( f_h \circ f_g \), where \( f_g : C \to Z \) is the encoder function and \( f_h : Z \to C \) is the decoder function. Here, \( z \) denotes the latent codes obtained by ap-
plying the encoder function to the mini-batch \( c \), i.e., \( z = f_b(c) \). In a forward pass, we compute the persistent homology in both the original shape space and the generated latent space, as below:

\[
(G^*, \phi^*) := \mathcal{H}(V_c(c)) \\
(G^{'*}, \phi^{'*}) := \mathcal{H}(V_c(z))
\]  

(9)

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 from the pairings, denoted as \( G^* = D^*[\phi^*] \). We treat \( D^*[\phi^*] \) as a vector in \( \mathbb{R}^{ |\phi^*| } \). By comparing the persistence diagrams obtained from the data space and latent space, we can construct a topological regularization term \( L_{topo} \), which is added to the reconstruction loss of an autoencoder. The overall loss function is then given by:

\[
L = L_{rec}(c, f_g(f_b(c))) + \lambda L_{topo}
\]  

(10)

where \( L_{rec} \) is the reconstruction loss, \( f_b \) and \( f_g \) are the encoder and decoder functions respectively, and \( \lambda \) is a regularization parameter that controls the strength of the regularization.

Let us consider how to express \( L_{topo} \). We select edge indices from \( \pi' \) and \( \pi \) to calculate the \( V \) 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 \( e \) and \( z \). 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, \( L_{topo} = L_{C \rightarrow Z} + L_{Z \rightarrow C} \), where

\[
L_{C \rightarrow Z} := \frac{1}{2} \left\| D^*[\phi^*] - D^*[\phi^{'*}] \right\|^2
\]

and

\[
L_{Z \rightarrow C} := \frac{1}{2} \left\| D^*[\phi^{'*}] - D^*[\phi^*] \right\|^2.
\]

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

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

\[
\frac{\partial}{\partial \omega} L_{C \rightarrow Z} = \frac{\partial}{\partial \omega} \left( \frac{1}{2} \left\| D^*[\phi^*] - D^*[\phi^{'*}] \right\|^2 \right)
\]

\[
= -\delta^T \left( \frac{\partial D^*[\phi^*]}{\partial \omega} \right)
\]

\[
= -\delta^T \left( \sum_{i=1}^{\left|\phi^*\right|} \frac{\partial D^*[\phi^*]}{\partial \omega} \right)
\]

In the above equation, the size of a persistence pairing is denoted by \( |\phi^*| \), while \( D^*[\phi^*] \) indicates the \( i \)th component of the vector of paired distances. An analogous derivation applies to \( L_{Z \rightarrow C} \), where \( \phi^{'*} \) is substituted with \( \phi^* \). Furthermore, since the distances between input samples are independent of encoding by definition, the derivative of \( D^* \) with respect to \( \eta \) must be zero. Although a diagram is robust to infinitesimal changes of its entries, according to Cohen-Steiner et al. (2007), our topological loss is differentiable for each update step during training.

![Fig. 6: 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 \( z_d \).](image)

With the built latent shape space, we train a neural network as shown in Fig. 6 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 bimananipulation algorithms in [42].

### 5.3. Controller Design

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

**Lemma 1.** In [40], the function \( \text{sgn}(x) = |x| \) if \( x \neq 0 \), and \( \text{sgn}(x) \) is defined, where \( x \in \mathbb{R} \), \( k > 0 \), and \( \text{sgn} \) denotes the standard sign function.

**Lemma 2.** [40] \( \left( \sum_{i=1}^{n} |x_i| \right)^p \leq \sum_{i=1}^{n} |x_i|^p \) holds for any \( x_i \in \mathbb{R}, i = 1, \ldots, n \), where \( p \) is a real number satisfying \( 0 < p < 1 \).

**Lemma 3.** [40] \( n^{1-p} \left( \sum_{i=1}^{n} |x_i| \right)^p \leq \sum_{i=1}^{n} |x_i|^p \) holds for any \( x_i \in \mathbb{R}, i = 1, \ldots, n \), and \( p > 1 \).

**Lemma 4.** [41] For any \( x \in \mathbb{R} \) and \( \delta > 0 \), we have the inequality satisfies:

\[
0 \leq |x| - x \tan h(x/\delta) - \delta \kappa \quad \text{where} \quad \kappa = 0.2785 \quad \text{with satisfying} \quad \kappa = e^{-(e^\delta)}.
\]

**Lemma 5.** [42] For \( h > 0 \) and \( x \geq 0, y > 0 \), the following inequality holds:

\[
x^h(y - x) \leq (y^h - x^h)/(1 + h).
\]

**Lemma 6.** [42] For \( h > 1, x > 0, y \leq x \) and \( y \in \mathbb{R} \), it holds that:

\[
(x - y)^h \geq y^h - x^h.
\]
**Definition 1.** In \(\mathbb{R}^n\), we can find the following vectorial power definitions for any arbitrary vector \(x \in \mathbb{R}^n\):

\[
\text{sign}^k(x) = \begin{bmatrix} \text{sign}^k(x_1), \ldots, \text{sign}^k(x_n) \end{bmatrix}^T \in \mathbb{R}^n
\]

\[
|x|^k = \text{diag}\left[|x_1|^k, \ldots, |x_n|^k\right] \in \mathbb{R}^{n \times n}
\]

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:

\[
s = \mathbf{J}\mathbf{u} = \mathbf{J}\mathbf{u} + \dot{\mathbf{J}}\mathbf{u}
\]

(11)

Two error variables are defined by:

\[
e_1 = s - s_d, \quad e_2 = s - \dot{\mathbf{J}}\mathbf{u}
\]

(12)

and its derivative with respect to time is:

\[
\dot{e}_1 = s - s_d, \quad \dot{e}_2 = s - \dot{\mathbf{J}}\mathbf{u} - \dot{\mathbf{J}}\mathbf{u}
\]

(13)

Combining with (11), it yields

\[
\dot{e}_1 = \dot{\mathbf{J}}\mathbf{u} + \dot{\mathbf{J}}\mathbf{u} - \dot{s}_d
\]

(14)

The velocity control input can be defined as below:

\[
\mathbf{u} = \dot{\mathbf{J}}^+ \left(\dot{s}_d - a_{11} \text{sign}^{2b_{11}}(e_1) - a_{12} \text{sign}^{2b_{12}}(e_1)\right)
\]

(15)

where \(\dot{\mathbf{J}}^+\) is the pseudo-inverse of the Jacobian matrix \(\dot{\mathbf{J}}\), \(a_{11} > 0, a_{12} > 0, 0 < b_{11} < 1, b_{12} > 1\) are design parameters specifying the convergence speed of the controller (15) and the system stability indirectly. In order to measure the error of shape tracking, a quadratic function is introduced:

\[
V_1(e_1) = \frac{1}{2} e_1^T \dot{e}_1
\]

(16)

Time differentiation of (16) yields

\[
V_1(e_1) = e_1^T \dot{e}_1 = e_1^T \left(\dot{\mathbf{J}}\mathbf{u} - \dot{s}_d\right) + e_1^T \dot{\mathbf{J}}\mathbf{u}
\]

(17)

Substituting the controller (15) into (17), one can get:

\[
V_1 = e_1^T \left(-a_{11} \text{sign}^{2b_{11}}(e_1) - a_{12} \text{sign}^{2b_{12}}(e_1)\right) + e_1^T \dot{\mathbf{J}}\mathbf{u}
\]

(18)

Considering Lemma 2 and Lemma 3, it yields

\[
V_1 \leq -a_{11} \|e_1\|^2 / 4 - a_{12} p^{Lb_{12}} \|e_1\|^2 + e_1^T \dot{\mathbf{J}}\mathbf{u}
\]

(19)

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

\[
e_1^T \dot{\mathbf{J}}\mathbf{u} \leq \|e_1\|^2 / 4 + \eta \|\mathbf{u}\|^2
\]

(20)

Substituting (19) into (18) obtains:

\[
V_1 \leq -a_{11} \|e_1\|^2 / 4 - a_{12} p^{Lb_{12}} \|e_1\|^2 + e_1^T \dot{\mathbf{J}}\mathbf{u}
\]

(21)

The adaptation rule of the DJM is can be defined as:

\[
\dot{\mathbf{J}} = (a_{21} \text{sign}^{2b_{11}}(e_2) + a_{22} \text{sign}^{2b_{12}}(e_2) + s - \dot{\mathbf{J}}\mathbf{u} + \sigma)\mathbf{u}^T
\]

\[
\sigma = e_2^T (\text{tanh}(\|\mathbf{u}\|^2 / \delta)) \|\mathbf{u}\|^2 + \frac{1}{4} \|\mathbf{e}_1\|^2
\]

(22)

where \(a_{21}, a_{22} > 0, 0 < b_{11} < 1, b_{12} > 1\) are design parameters determining the convergence speed of the approximation of the Jacobian matrix. The adaptive rule of \(\dot{\eta}\) is designed as follows:

\[
\dot{\eta} = \text{tanh}(\|\mathbf{u}\|^2 / \delta) \|\mathbf{u}\|^2 - a_{31} \eta^{2b_{11}} + a_{32} \eta^{2b_{12} + 1}
\]

(23)

where \(\delta\) is a positive constant. \(a_{31} > 0, a_{32} > 0, 0 < \frac{1}{2} < b_{12} < 1\) are design parameters. Define the quadratic function \(V_2(e_2) = \frac{1}{2} e_2^T e_2\), and differentiating \(V_2\) with respect to time and using (13) gains

\[
\dot{V}_2 = e_2^T \dot{e}_2 = e_2^T (s - \dot{\mathbf{J}}\mathbf{u} - \dot{\mathbf{J}}\mathbf{u})
\]

(24)

Substituting (20) into (22), and considering Lemma 2 and Lemma 3 it yields

\[
\dot{V}_2 = -a_{21} e_2^T \text{sign}^{2b_{11}}(e_2) - a_{22} e_2^T \text{sign}^{2b_{12}}(e_2) - e_2^T \sigma
\]

\[\leq -a_{21} \|e_2\|^2 b_{11} - a_{22} p^{Lb_{12}} \|e_2\|^2 b_{12} - e_2^T \sigma
\]

(25)

Consider the energy-like function:

\[
V = V_1 + V_2 + \frac{1}{2} \hat{\eta}^2
\]

(26)

where \(\hat{\eta} = \eta - \tilde{\eta}\) is the estimation error, with \(\tilde{\eta}\) being the estimation of \(\eta\). With \(V_1\) and \(V_2\), time differentiation of (26) yields

\[
\dot{V} = -a_{11} \|e_1\|^2 b_{11} - a_{12} p^{Lb_{12}} \|e_1\|^2 b_{12} - e_1^T \sigma - \eta \hat{\eta}
\]

\[\leq -a_{21} \|e_2\|^2 b_{11} - a_{22} p^{Lb_{12}} \|e_2\|^2 b_{12} - e_2^T \sigma
\]

(27)

With the adaptive update rule (21) and Lemma 4, we can get the following inequality:

\[
\frac{1}{4} \|e_1\|^2 + \eta \|\mathbf{u}\|^2 - e_1^T \sigma - \eta \hat{\eta}
\]

\[= \eta \|\mathbf{u}\|^2 - \hat{\eta} \|\mathbf{u}\|^2 - \tilde{\eta} \hat{\eta}
\]

\[\leq \eta \|\mathbf{u}\|^2 + \eta \|\mathbf{u}\|^2 \text{tanh}(\frac{\|\mathbf{u}\|^2}{\delta}) - \hat{\eta}
\]

\[\leq \eta \delta + \eta \|\mathbf{u}\|^2 \text{tanh}(\frac{\|\mathbf{u}\|^2}{\delta}) - \hat{\eta}
\]

\[\leq \eta \delta + a_{31} \hat{\eta}^{2b_{11}} + a_{32} \hat{\eta}^{2b_{12} + 1}
\]

And, considering Lemma 3 and Lemma 4, we have:

\[
\hat{\eta}^{2b_{11}} - \frac{2 \|\mathbf{u}\|^2 - \hat{\eta}^{2b_{11}}}{\delta} \leq 2 \|\mathbf{u}\|^2 + \hat{\eta}^{2b_{12} + 1}
\]

(28)
Substituting (26) and (27) into (25), it yields
\[
\dot{V} = -(a_{11}||e_i||^2b_1 + a_{21}||e_i||^2b_2) \\
- (a_{12}p^{1-b_1}||e_i||^2b_1 + a_{22}p^{1-b_2}||e_i||^2b_2) \\
- \frac{a_{31}}{2b_3} \eta k \delta \eta k + \frac{a_{32}}{2b_3} + 2 \eta k^{b_2+1} + \eta k^{b_2+2} \\
+ (\eta k \delta + a_{31} \eta k^{b_2} + a_{32} \eta k^{b_2+1} + \eta k^{b_2+2}) \\
\leq -a_1 V^b - a_2 V^{b_1} + \Omega
\]
where the coefficients are:
\[
a_1 = \min(2a_{11}, 2a_{21}, \frac{a_{31}}{b_3}) \\
a_2 = \min(2a_{12}, 2a_{22}, \frac{a_{32}}{b_3 + 1}) \\
b_1 = \min(b_{11}, b_{21}, b_2) \\
b_2 = \min(b_{12}, b_{22}, b_3 + 1) \\
\Omega = \eta k \delta + \frac{a_{31}}{b_3} \eta k^{b_2} + \frac{a_{32}}{b_3 + 1} \eta k^{b_2+2}
\]

According to the fixed-time Lyapunov stability [x], it can be concluded that the system is practical fixed-time stable and the tracking error convergences to the neighborhood near the equilibrium in the fixed-time.

where 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 Naive-C controller and its main process is illustrated in the upper part of the Fig. 7. The leftmost k centerline points of the leader side is marked with red color (here k = 3 for illustration). After the motion of the leader side of the deformable linear object, the leader pose will become \( p_{i+1}^{\text{leader}} \) from \( p_{i}^{\text{leader}} \), then Naive-C estimates a transformation \( T_{i+1}^{\text{leader}} \) to finally generate the desired pose for the follower side using the Equ. (30) 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 expansive optimization solving process will always reduce the system response time and perform poorly during human robot interaction process. The second baseline approach is to directly estimate the \( p_{i}^{\text{follower}} \) 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 \( \{p_0, p_1, p_2\} \). 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.

5.4. Baseline Model

In this section, 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. The is based on the regularity that the relative transformations of the leader side and follower side are identical, which can be denote as below:

\[
p_{i}^{\text{follower}} = p_{i-1}^{\text{follower}} + T_{i}^{\text{leader}} T_{i}^{\text{follower}} (j \leq i)
\]

where \( T_{i}^{\text{leader}} \) is the transformation of the leader side of the deformable linear object between time step 0 and time step i. The experimental setting for validating the performance of the proposed human-robot collaboration approach for reactive deformable object tasks is given in this section, followed by the implementation details of the GMM-based deformable linear object state estimator, latent shape space construction us-

Fig. 7: Conceptual representation of our two baseline models for solving reactive shape servoing tasks of the deformable linear object under human-robot collaboration. The upper Naive-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.

Fig. 8: 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 human hand and robot hand, a depth camera D455 (eye-to-hand configuration) to measure the object’s state, a step-motor to simulate the human’s interaction, a single-arm robot (URS) to manipulate the cable to maintain its origin shape configuration.
ing topological auto-encoder, and reactive shape controller in
the latent space, respectively. Finally, our human-robot col-
aboration approach is compared to other advanced solutions
for reactive deformable object manipulation tasks. The
advantage of our proposed latent shape controller is validated by
quantitatively measuring a series of reactive deformable ob-
ject manipulation tasks using motor-robot interactions imitating
the human-robot interactions. Furthermore, we also conducted
a set of experiments and compare our proposed framework to
other advanced approaches on reactive deformable object ma-
nipulation tasks with real-time human-robot collaborations.

6.1. Experiment Setup

Fig. 8 shows the experiment setup for our approach, where a
RealSense RGB-D camera (D455) is used to observe the de-
formable object manipulation process from a top-down per-
pective, namely, main view. Besides, we also consider prov-
ing a side view with a commonly used Logi RGB camera
(C270) to have a better overview of the entire manipulation pro-
cess from a third person perspective. During the process, an
elastic sponge bar (viz. a deformable linear object) is manipu-
lated with a UR-5 robot when 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 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.
8(a)). Furthermore, a real-time human robot collaboration is
conducted to examine the overall performance of our proposed
entire 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 the Fig. 8 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., \( |u| \leq 0.04 \text{m/s} \). The proposed algorithm is implemented
on ROS/URX running within a servo-control loop of around

20Hz. A video of the conducted experiments can be obtained
from https://sites.google.com/view/hrc-dom

<table>
<thead>
<tr>
<th>Category</th>
<th>( \ell )-Trust</th>
<th>( \ell )-Cont</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PCA</td>
<td>TSNE</td>
<td>AE</td>
</tr>
<tr>
<td>Line-shaped</td>
<td>0.907</td>
<td>0.953</td>
<td>0.947</td>
</tr>
<tr>
<td>Pos. Arch-shaped</td>
<td>0.868</td>
<td>0.910</td>
<td>0.884</td>
</tr>
<tr>
<td>Neg. Arch-shaped</td>
<td>0.859</td>
<td>0.938</td>
<td>0.909</td>
</tr>
<tr>
<td>Pos. S-shaped</td>
<td>0.788</td>
<td>0.888</td>
<td>0.837</td>
</tr>
<tr>
<td>Neg. S-shaped</td>
<td>0.838</td>
<td>0.865</td>
<td>0.857</td>
</tr>
</tbody>
</table>

Fig. 9: 3D shape processing pipeline for the reactive deformable object manip-
ulation tasks. Fig. 9(a) is an aligned RGBD image frame of the working space.
Fig. 9(b) is to extract the deformable object region by using appropriate HSV
color filter. Fig. 9(c) presents the point cloud of the deformable linear object
after computing with Open3D library. Fig. 9(d) 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. 9 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. 9(a). Then,
we mask the deformable object region by designing an appro-
priate HSV color filter to compute out the point cloud of the
manipulated object based on camera parameters with Open3D
library (see Fig. 9(a)). After that, GMM-based estimator is per-
formed 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 cen-
terline 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.

Fig. 10: Five different shape categories to measure the latent representation
performance for the deformable linear object, namely, Line, Arch., Arch., S,
and S. categories.
6.3. Validation of Latent Shape Representation

To validate the performance of the topological auto-encoder on shape representation, we compare our TAE with three commonly used representation learning approaches including PCA, TSNE, 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 the proceeding section. After examining the collected shapes, we classify them into five different shape categories, namely, Line, Pos. Arch, Neg. Arch, Pos. S and Neg. S class as shown in Fig. 10. We evaluate the reconstruction errors between the input shape $c_i$ and reconstructed shape $\tau_i$ with root mean square error (RMSE).

\[ \text{RMSE}_{\text{rec}} = \|c_i - \tau_i\|^2 \]  

(31)

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 $\ell$ in the abbreviations). Specifically, the first is called trustworthiness ($\ell$-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 ($\ell$-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 $\ell$-Trust and lowest RMSE over all shape categories and shows a large improvement over other approaches. With respect to $\ell$-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. Validation of Sensorimotor Approximation

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, two 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 [44] (referred as VS in the following), and the other is a latent shape control (LSC) model with naive auto-encoder [45] (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 human action. The shape accuracy is defined as the RMSE between the current shape $c_i$ and its desired target shape $c^*_i$ as below:

\[ \text{RMSE}_{\text{dom}} = \|c_i - c^*_i\|^2 \]  

(32)

where the desired target shape is computed based on the transformation $T^p_p$ between the beginning leader pose $p_0$ and current leader pose $p_i$. By calibrating the transformation between the stepping motor device and the robot base, the current leader pose $p_i$ is easy to obtain in the global coordinate system. Finally, the desired target shape is obtained as below:

\[ c^*_i = c_0 \ast T^p_p \]  

(33)

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

Four different experimental cases are set to deform four different shapes, namely, Vertical Track for shape Arch $^1$, Vertical Tilt Track for Arch $^3$, Tilt Right Track for S $^1$, and Left Track for Line. We ran 10 experiments for different experimental settings, and Table 2 and Fig. 11 show the quantitative and qualitative performance results of different sensorimotor models on different tasks for motor-robot experiments, respectively. We measure and compare different approaches from two perspectives: the shape accuracy and the response time, which are reported in centimeters (cm) and in milliseconds (ms), respectively. From the table, it can be seen that TopoLSC has the best performance across all cases, followed by AutoLSC, and then VS. The differences in performance between the three methods are quite significant, especially in Cases 1 and 2. TopoLSC achieved a Shape Accuracy of less than 0.5 cm in all cases, while VS had a Shape Accuracy of around 3.5-3.9 cm. AutoLSC’s performance was in between the two, with a Shape Accuracy of around 1.8-2.8 cm. In terms of Response time, TopoLSC and AutoLSC show a very similarly competitive performance, achieving the response time around 43-51 ms and showing a noticeable lifting over VS’s Response Time (around 113-197 ms) In summary, the results of the comparison show that the TopoLSC method is the most accurate and efficient among the three methods, followed by the AutoLSC method, while the VS method performs the worst.

6.5. Evaluation of Reactive Manipulation

To further measure the overall performance of proposed approach, a series of reactive shape manipulation tasks are also conducted in the context of real human-robot collaborations. For different tasks, the human hands are executing roughly similar motions for different approaches. Table 2 presents a comparison of the performance of different frameworks (Naive-C, PoseCNN-C, and TopoLSC) on different human-robot collaboration tasks, with the same metrics in the previous section being Shape Accuracy (cm) and Response Time (ms). In terms of shape accuracy, TopoLSC (Ours) outperforms both Naive-C and PoseCNN-C in all four tasks. The difference is particularly significant in case 2, where TopoLSC achieves a shape accuracy of 1.23 cm compared to 5.52 cm and 3.78 cm achieved
(a) Vertical Track, $\text{Arch}^1$
(b) Vertical Tilt Track, $\text{Arch}^2$
(c) Tilt Right Track, $S$
(d) Left Track, $\text{Line}$

Fig. 11: Qualitative and quantitative results in motor-robot experiments. (a)-(d) show qualitative results of four motor-robot experiments ({$\text{Vertical Track, Arch}^1$}, {$\text{Vertical Tilt Track, Arch}^2$}, {$\text{Tilt Right Track, S}$}, and {$\text{Left Track, Line}$}) by using three different sensorimotor approaches (VS [44], AutoLSC [45], and TopoLSC (Ours)), where red arrows represents 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 motor and robot, respectively. (e)-(h) shows quantitative results of deformation error and velocity curve along x, y, and z-axis under the corresponding motor-robot experiments.

Table 2: Performance of Different Sensorimotor Models on Different Tasks for Motor-robot Experiments

<table>
<thead>
<tr>
<th>Method</th>
<th>Shape Accuracy (cm)</th>
<th>Response Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Case (a)</td>
<td>Case (b)</td>
</tr>
<tr>
<td>VS [44]</td>
<td>3.65 ± 0.07</td>
<td>3.15 ± 0.09</td>
</tr>
<tr>
<td>AutoLSC [45]</td>
<td>1.81 ± 0.73</td>
<td>2.21 ± 0.65</td>
</tr>
<tr>
<td>TopoLSC (Ours)</td>
<td>0.44 ± 0.09</td>
<td>0.35 ± 0.06</td>
</tr>
</tbody>
</table>
Table 3: Performance of Different Frameworks on Different Human-robot Collaboration Tasks.

<table>
<thead>
<tr>
<th>Method</th>
<th>Task 1</th>
<th>Task 2</th>
<th>Task 3</th>
<th>Task 4</th>
<th>Task 1</th>
<th>Task 2</th>
<th>Task 3</th>
<th>Task 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naive-C</td>
<td>4.26 ± 2.41</td>
<td>5.52 ± 2.99</td>
<td>4.21 ± 2.69</td>
<td>4.40 ± 2.43</td>
<td>237 ± 83</td>
<td>259 ± 108</td>
<td>227 ± 87</td>
<td>246 ± 93</td>
</tr>
<tr>
<td>PoseCNN-C</td>
<td>3.37 ± 1.21</td>
<td>3.78 ± 1.94</td>
<td>3.68 ± 2.02</td>
<td>3.52 ± 2.38</td>
<td>32 ± 11</td>
<td>33 ± 9</td>
<td>18 ± 13</td>
<td>17 ± 14</td>
</tr>
<tr>
<td>TopoLSC (Ours)</td>
<td>0.72 ± 0.10</td>
<td>1.23 ± 0.17</td>
<td>0.90 ± 0.12</td>
<td>0.95 ± 0.19</td>
<td>44 ± 10</td>
<td>49 ± 13</td>
<td>51 ± 12</td>
<td>47 ± 12</td>
</tr>
</tbody>
</table>

Fig. 12: Qualitative and quantitative results in human-robot experiments. (a)-(d) show qualitative results of four human-robot experiments (Case 1, S\(^{-}\), Case 2, Arch\(^{\dagger}\), Case 3, Arch\(^{\ddagger}\), and Case 4, Arch\(^{\|}\)) by using three different overall frameworks (Naive-C, PoseCNN-C, and TopoLSC (Ours)), 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 motor and robot, respectively. (e)-(h) shows quantitative results of deformation error and velocity curve along x, y, and z-axis under the corresponding human-robot experiments.
by Naive-C and PoseCNN-C, respectively. In terms of response time, PoseCNN-C achieves the lowest response time in all four cases, while Naive-C has the highest response time. TopoLSC’s response time is also lower than Naive-C but higher than PoseCNN-C in all cases. Overall, the results suggest that TopoLSC outperforms the other frameworks in terms of shape accuracy while maintaining a reasonable response time. Therefore, TopoLSC could be a promising framework for human–robot collaboration tasks that require high accuracy. However, for tasks that require fast response times, PoseCNN-C could be a better option.

6.6. Limitations

While the proposed approach of using topological latent control models and human–robot collaboration for reactive deformable linear object manipulation 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 limited to the manipulation of deformable linear objects, which may not be suitable for all deformable object manipulation applications. For example, in surgical robotics, the manipulation of soft tissue may require a different perception approach. Finally, the use of human–robot collaboration may introduce additional challenges, such as the need for effective communication and coordination between the human operator and the robot. In addition, the system may be sensitive to differences in human expertise, which may affect the quality of the manipulation performance. Overall, while the proposed approach has shown promise 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 needed to address the limitations and explore its applicability to other types of deformable objects and real-world scenarios.

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 the 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 can focus on extending this approach to other types of deformable objects and exploring its applicability in real-world scenarios.

References


