Manipulation of Deformable Objects with Soft Grippers Based on Adaptive Fuzzy Control and a Self-Supervised Feature Descriptor

Yuexuan Xu, Ning Han, Jinrui Li, Tong Yang, *Member, IEEE*, Yun-Hui Liu, *Fellow, IEEE*, Ning Sun, *Senior Member, IEEE*, and David Navarro-Alarcon, *Senior Member, IEEE*

***Abstract*—Although visual servoing control is widely used in practical applications, its application to deformable object manipulation (DOM) remains underexplored. This is due to the significant diversity of material properties and geometric configurations, the high uncertainties of deformation characteristics, and the high-dimensional configuration spaces. To this end, a new data-driven visual servoing control method is proposed for three-dimensional DOM. Specifically, a self-supervised keypoint detection network is developed without manual annotation while preserving spatial geometric information, addressing limitations of traditional dimensionality reduction methods. The network employs weighted feature aggregation modules to effectively balance the fusion of diverse local information while preserving features essential for keypoint localization. Furthermore, a geodesic distance loss function ensures the semantic consistency of detected keypoints across deformation processes. Based on the detected keypoints, an adaptive fuzzy-based Jacobian estimator is developed to establish the online mapping relationship between keypoint velocities and robot joint velocities. To optimize transient performance, a prescribed performance control method is designed to ensure that keypoint errors converge within predefined spatial funnel functions. Additionally, a pneumatic soft gripper based on a rigid origami-inspired mechanism is designed to prevent damage to deformable objects and provide compliant interaction. Finally, detailed stability analysis and a series of experimental results are provided to verify the feasibility of the proposed method.**

***Index Terms*—Self-supervised, point cloud keypoint detection, prescribed performance control, pneumatic soft grippers, deformable object manipulation.**

This work was supported in part by the Research Grants Council of Hong Kong under Grant AoE/E-407/24-N, in part by the National Natural Science Foundation of China under Grant 62533014 and Grant 62303245, in part by the Tianjin Science Fund for Distinguished Young Scholars under Grant 22JCJQJC00140, in part by the Natural Science Foundation of Tianjin under Grant 24JCZXJC00220, and in part by the Beijing-Tianjin-Hebei Basic Research Cooperation Special Project under Grant F2024205028. *(Corresponding authors: David Navarro-Alarcon and Ning Sun.)*

Yuexuan Xu is with the Department of Mechanical Engineering, The Hong Kong Polytechnic University, KLN, Hong Kong, with the Institute of Robotics and Automatic Information Systems (IRAIS), College of Artificial Intelligence, and the Academy for Advanced Interdisciplinary Studies, Nankai University, Tianjin 300350, China, and with the Institute of Intelligence Technology and Robotic Systems, Shenzhen Research Institute of Nankai University, Shenzhen 518083, China (e-mail: yuexuan.xu@connect.polyu.hk).

Ning Han and David Navarro-Alarcon are with the Department of Mechanical Engineering, The Hong Kong Polytechnic University, KLN, Hong Kong (e-mail: ningg.han@connect.polyu.hk; dnavar@polyu.edu.hk).

Jinrui Li is with the Department of Mechanical Engineering, The Hong Kong Polytechnic University, KLN, Hong Kong, and with the School of Mechanical and Automotive Engineering, South China University of Technology, Guangzhou 510641, China (e-mail: 24110324d@connect.polyu.hk).

Tong Yang and Ning Sun are with the Institute of Robotics and Automatic Information Systems (IRAIS), College of Artificial Intelligence, and the Academy for Advanced Interdisciplinary Studies, Nankai University, Tianjin 300350, China, and with the Institute of Intelligence Technology and Robotic Systems, Shenzhen Research Institute of Nankai University, Shenzhen 518083, China; Ning Sun is also with the Shenzhen Loop Area Institute, Shenzhen 518048, China (e-mail: yangt@nankai.edu.cn; sunn@nankai.edu.cn).

Yun-Hui Liu is with the T Stone Robotics Institute, Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, NT, Hong Kong (e-mail: yhliu@cuhk.edu.hk).

I. Introduction

VISUAL servoing technology plays a crucial role in achieving precise object manipulation. Compared with traditional rigid object manipulation, deformable object manipulation (DOM) presents significant research value in fields, including advanced manufacturing [1], medical robotics [2], service robotics [3], and agricultural systems [4]. However, DOM remains challenging due to the infinite-dimensional configuration space, irregular deformation characteristics, and complex dynamics of such objects. Although recent research attempts to address these challenges, real-time three-dimensional (3-D) shape control in complex environments remains an open problem.

A. Related Work

In practical applications, DOM typically requires close interaction with the environment. Meanwhile, compared with non-deformable objects, deformable objects are relatively soft and fragile. Hence, DOM necessitates integrating both hardware design and theoretical methods.

From a hardware perspective, rigid grippers are widely employed in DOM tasks, but can easily damage deformable objects during shape control and create safety risks for compliant interaction. Furthermore, although traditional soft grippers exhibit excellent adaptability and flexibility, their load capacity is limited due to the lack of rigid components and internal support. When manipulating deformable objects, the insufficient stiffness leads to relative displacement between the objects and the grippers, degrading the performance of visual servoing. Once grippers are positioned obliquely, severe bending deformation can cause the grasped objects to fall off. Li *et al.* develop a new pneumatic soft gripper (PSG) that controls actuator bending through particle injection into air chambers, enhancing PSG's stiffness [5]. Qiu *et al.* propose a novel 3-D deformable PSG with mutually perpendicular pneumatic soft actuators (PSAs) that enables secure grasping of objects [6]. However, grasping tests on fragile and deformable objects are not conducted in the aforementioned work.

From a methodological perspective, DOM methods have evolved significantly over the past three decades, including model-based methods [7], [8], data-driven methods [9], [10], and learning-based methods [11], [12]. Early studies mainly aim to develop accurate models of deformable objects for

predicting their deformation. Wada *et al.* [13] propose a mass-spring model to simulate the deformation of soft objects. However, this method is restricted to small deformations and planar motion, lacking experimental validation. In [14], a forming control method is developed for rheological food dough. Despite its effectiveness in specific scenarios, this method is highly dependent on model accuracy and has limited generalization ability for different materials or shapes. To address these issues, a dynamic Broyden method is presented to establish the real-time mapping relationship between keypoint motion and robot joint motion [15].

Later, more data-driven visual servoing control methods are designed to manipulate deformable objects, which aim to learn control strategies directly from data. First, the state of deformable objects is transformed into low-dimensional features through methods such as the Fourier series [10] and principal component analysis (PCA) [16]. Subsequently, the Jacobian matrix relating control signals to feature variations is estimated using Kalman filtering [17] or adaptive learning [4]. In [10], a truncated Fourier series-based feedback representation is developed to describe object shapes with fewer coefficients accurately. Dahroug *et al.* develop a PCA-based visual servoing method to extract 3-D pose information from OCT C-scan data, achieving the positioning accuracy of 0.052 ± 0.03 mm [16]. However, traditional PCA and autoencoder methods obtain feature vectors in abstract latent spaces when reducing feature dimensions, lacking physical and spatial information. Based on this, several feasible methods are proposed to preserve the spatial and geometric information of objects. Alambeigi *et al.* propose an optimal iterative method that incorporates visual feedback data to simultaneously learn and estimate the deformation behavior of objects [9].

In recent years, several studies have integrated deep learning networks into data-driven visual servoing control methods. Deep learning networks [18] effectively achieve feature dimensionality reduction for 2-D images and 3-D point clouds of objects by detecting feature points. These feature point sets effectively represent latent spaces that encode key spatial information. To accurately detect keypoints from point clouds, Xu *et al.* develop a novel geodesic distance regression-based semantic keypoint detection method for pig point clouds [19]. Jin *et al.* propose KeypointDETR, an end-to-end transformer-based method for 3-D keypoint detection that directly predicts keypoint heatmaps and probabilities using bipartite matching loss [20]. However, the above methods belong to supervised/semi-supervised learning methods requiring manual annotation of feature points in datasets, which is extremely complex and time-consuming for 3-D deformable objects. Based on this, Hou *et al.* present an unsupervised KeyGrid network to detect the keypoints from deformable objects [21]. Zohaib and Del Bue propose a self-supervised SC3K network that estimates semantically coherent 3-D keypoints from rotated, noisy, and decimated point cloud data [22]. Then, Zohaib *et al.* present a Self-Geo network to estimate consistent 3-D keypoints on deformable shapes by enforcing geodesic distance preservation across deformation sequences [23].

However, data-driven visual servoing control methods for DOM have the following limitations that need to be addressed.

1) Traditional feature dimensionality reduction methods obtain feature vectors in abstract latent spaces, lacking physical and spatial information. Although deep learning-based keypoint detection methods can extract keypoints with crucial spatial information, current methods have the following limitations. First, supervised and semi-supervised learning methods rely on prior knowledge of keypoints in deformable objects. Second, few works consider the semantic consistency of keypoints during object deformation, focusing only on keypoint detection for the desired shape of deformable objects.
2) Data-driven visual servoing control methods suffer from learning uncertainty and initial parameter dependency [24], [25], [26], which cannot ensure that keypoints converge within predefined spatial constraints. Excessive spatial errors of keypoints degrade transient performance and may damage deformable objects or cause task failure. Recently, several barrier function-based visual servoing control methods constrain image features within the field of view [27], [28]. However, how to design spatial constraints for keypoints while ensuring closed-loop stability remains an open problem.

B. Our Contributions

To address the above-mentioned challenges, this paper proposes a deep learning-based visual servoing control method for DOM. Specifically, a new self-supervised network is constructed to detect keypoints of deformable objects. Then, an adaptive fuzzy-based Jacobian estimator is developed to obtain the real-time mapping relationship between keypoint motion and robot joint motion. To ensure that keypoints do not exceed spatial constraints, a prescribed performance control (PPC) method is proposed to guarantee that keypoint errors converge within the predefined funnel functions. Furthermore, a new PSG is developed to provide safe and compliant interaction while preventing damage to deformable objects during shape control. The core contributions are summarized as follows.

1) A *new* self-supervised network is constructed to detect keypoints from point clouds of deformable objects. Compared with [19], [20], [29], the proposed network requires *no* prior information about deformed shapes or manual annotations, while the detected keypoints effectively capture latent spaces with key spatial information. In the encoder, a Weighted Feature Aggregation (WFA) module is introduced to enhance the feature extraction layer of traditional PointNet++ [21], [23], [30]. The improved network *effectively balances* the fusion of useful information within local regions while preserving point cloud features. Compared with [30], this paper proposes a geodesic distance loss function to ensure the *semantic consistency* of keypoints during object deformation. This loss function ensures geodesic distance consistency between keypoints across all frames.
2) An adaptive fuzzy Jacobian estimator combined with PPC is proposed to estimate the deformed Jacobian matrix online and ensure that keypoint errors converge *within* predefined spatial funnel functions.
3) The proposed PSGs employ a two-layer structure, consisting of a pneumatic drive layer (PDL) and a stiffness layer (SL). The SL integrates a rigid origami-inspired mechanism (ROM) with particles encapsulated in balloons. Compared with [31], this design ensures *firm* grasping while maintaining *flexible* grasping and

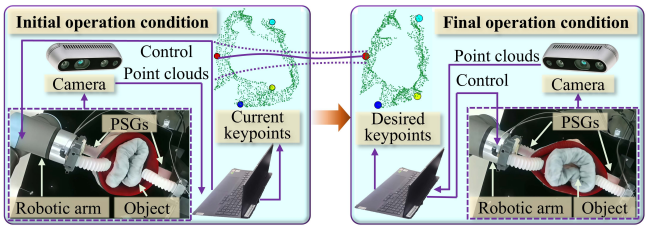


Fig. 1. Visual servoing system of the 3-D DOM.

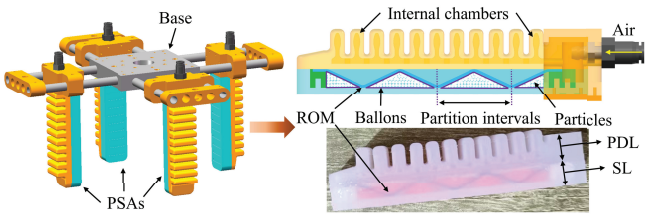


Fig. 2. Design of the PSG.

compliant interaction. Unlike [5], the proposed PSGs eliminate particle blockage in the air supply and achieve *uniform* load distribution across the SL.

4) This paper conducts detailed experiments on different deformable objects and application scenarios to verify the effectiveness of the proposed method and the structure design of PSGs. Meanwhile, the practicality and superiority of the proposed method are further verified through a series of ablation experiments.

The remainder of this paper contains five sections. The control objective of DOM is introduced in Section II. Afterwards, the design process of a new PSG is presented in Section III. Then, the keypoint detection, Jacobian estimator, and PPC are proposed in Section IV. Meanwhile, the corresponding stability analysis is provided based on Lyapunov theory in Section IV. In Section V, a series of experimental results is displayed. Finally, Section VI concludes this paper.

II. Problem Statement

The notations used throughout this paper are defined as follows: $\Re$ denotes the set of real numbers. $\Re^n$ represents n-dimensional vectors. $\Re^{m\times n}$ denotes $m \times n$ matrix[1]. $\|x\|$ is the Euclidean norm of x. Then, for $\boldsymbol{x}, \boldsymbol{y} \in \Re^n$, $\boldsymbol{x} \odot \boldsymbol{y}$ stands for the Hadamard product and $\boldsymbol{x} \cdot \boldsymbol{y} = \sum_{i=1}^{n} x_i y_i$ denotes the dot product. $\oplus$ represents element-wise concatenation for two or more vectors, matrices, or tensors.

This paper focuses on the 3-D deformable object manipulation problem using a 6-DoF robotic arm equipped with PSGs. The proposed visual servoing control method in latent space precisely manipulates deformable objects (sponges, ropes, and scarves) into desired shapes, as shown in Fig. 1. The shape control objective is formulated as follows.

A deformable object can be represented by 3-D point clouds $\boldsymbol{S}_{\text{origin}} \in \Re^{3N}$ where N is a sufficiently large positive integer. Using keypoint detection techniques, keypoints $\boldsymbol{S} = [\boldsymbol{S}_1, \boldsymbol{S}_2, \cdots, \boldsymbol{S}_u, \cdots, \boldsymbol{S}_a]^\top \in \Re^{3a}$ are detected from high-dimensional space $\boldsymbol{S}_{\text{origin}}$ where a stands for the number of keypoints ensuring $a \ll N$ and $\boldsymbol{S}_u = [x_u, y_u, z_u]^\top \in \Re^3$ denotes the coordinates of the u-th keypoint. Hence, the control objective is to control the joint angular velocities of the robotic arm[2] $\dot{\boldsymbol{q}} \in \Re^6$ making tracking errors of keypoints $\boldsymbol{e}_S = \boldsymbol{S} - \boldsymbol{S}_d \in \Re^{3a}$ converge to a neighborhood of origin under the time-varying constraints $\boldsymbol{\rho} \in \Re^{3a}$, where $\boldsymbol{S}_d \in \Re^{3a}$ denotes the desired keypoints.

[1]In this paper, regular letters denote scalars/matrices (e.g., x and $Y \in \Re^{m\times n}$) while bold letters stand for vectors (e.g., $\boldsymbol{a} \in \Re^m$).

[2]For convenience of the subsequent expressions, any information related to the state or time is omitted unless it is necessary to present.

To clarify our problem, this paper makes the following assumption.

Assumption 1: The manipulation motion of the robotic arm is sufficiently slow, so that only the quasi-static elastic deformation of the object needs to be considered [32].

Note that the quasi-static assumption can be used in many practical applications, such as the shape control of soft hoses, branches [4], fabric [33], plant and animal tissues/organs [34], deformable linear objects, sponges [30], and assembly of flexible parts [35].

III. Proposed Gripper Design

The proposed PSGs are designed to meet the following practical requirements.

1) *Grasping flexibility*: The proposed PSGs must not damage deformable objects during shape control.
2) *Grasping stability*: The designed PSGs provide firm grasping without relative displacement.

A. Structural Design of PSGs

A new type of PSGs is proposed, as depicted in Fig. 2. The gripper is composed of a base and four PSAs. Specifically, the base can be rigidly connected to the robotic arm, and the position of each PSA can be adjusted according to the practical scenarios. The proposed PSAs are divided into PDL and SL. The length, width, and height of the PSAs are 110 mm, 24 mm, and 25 mm, respectively. The heights of PDL and SL are 15 mm and 10 mm, respectively.

Inspired by human fingers, Dragon Skin 30 silicone is used as the outer layer of the PSA. The corrugated structure in the PDL enables the PSG to bend, and the ends of the PDL can be connected to external air circuits. By inflating the internal chambers of the PDL, a pressure difference is created between the PDL and the SL, causing the PSA to bend inward along the direction of the SL.

To address the insufficient stiffness of traditional PSAs, the SL is designed to provide rigid support. Specifically, the SL integrates a ROM [36] and particles encapsulated in balloons. The ROM is used to bear most of the loads and provide structural stiffness. To prevent excessive local deformation of the SL due to stress concentration, the particles are encapsulated in balloons and placed in partition intervals of the ROM. This design effectively avoids particle accumulation and blockage. During grasping operations, as the internal pressure of the PDL increases, the PSA bends to contact the object. At this time, the particles encapsulated in balloons can adapt to the surface shape of the object, while the ROM provides rigid support to achieve stable and reliable grasping.

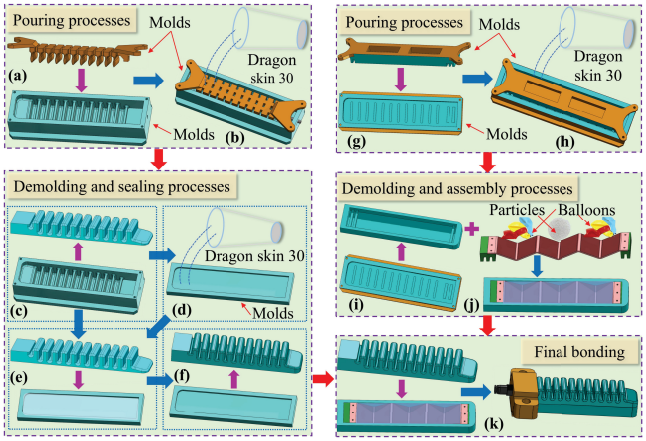


Fig. 3. Fabrication processes of the PSG. (a) Mold assembly of the PDL. (b) Pour liquid silicone rubber into the mold. (c) After curing, the preformed PDL is removed from the mold. (d) Pour liquid silicone rubber into the mold. (e) The preformed PDL is placed on the mold. (f) After curing, the PDL is removed from the mold. (g) Mold assembly of the SL. (h) Pour liquid silicone rubber into the mold. (i) After curing, the preformed SL is removed from the mold. (j) SL component assembly. (k) PDL and SL bonding.

B. Fabrication of PSAs

Fig. 3 (a)–(k) shows the complete fabrication processes of the PSA, including several key stages: pouring, demolding, sealing, assembly, and bonding.

Stage 1: Pouring and demolding. During the PSA mold assembly process, the upper mold is first inserted into the lower mold and fixed with bolts, as shown in Fig. 3(a) and (g). Due to the complex structure of the PSA, a release agent (Ease Release™ 200, Smooth-On Inc.) is uniformly applied to the mold surface before pouring to reduce demolding resistance and prevent structural damage during demolding. Subsequently, the pre-mixed liquid silicone rubber (Dragon Skin 30, Smooth-On Inc.) is slowly injected into the mold cavity to avoid turbulence and air trapping, as depicted in Fig. 3(b) and (h). Then, the mold is placed in a vacuum box and evacuated to an absolute pressure of 10 kPa (90% relative vacuum) for 3–5 minutes to completely remove air bubbles from the material. The curing process lasts for 18 hours at room temperature to ensure that the silicone rubber is fully cured. After curing, the preformed PDL and SL components are obtained through stretching and demolding processes, as shown in Fig. 3(c) and (i). At this stage, the PDL has hollow chambers, but the bottom remains unsealed.

Stage 2: PDL sealing. To achieve sealing of the PDL, a specific sealing mold is designed, as shown in Fig. 3(d). Liquid silicone rubber is injected into the sealed mold using the same pouring process, followed by vacuum degassing (10 kPa absolute pressure, 3–5 minutes). The preformed PDL is then placed on the upper surface of the mold, as depicted in Fig. 3(e). After the silicone rubber has fully cured, the sealed PDL component is obtained through stretching and demolding processes, as shown in Fig. 3(f).

Stage 3: SL component assembly. In this stage, rigid ROM and end fixing components are designed. Particles are filled into balloons, which are then uniformly distributed in the preset gaps of the ROM to ensure uniform load distribution across all sections of the SL. Finally, these components are assembled with the preformed SL to obtain the formed SL, as depicted in Fig. 3(j).

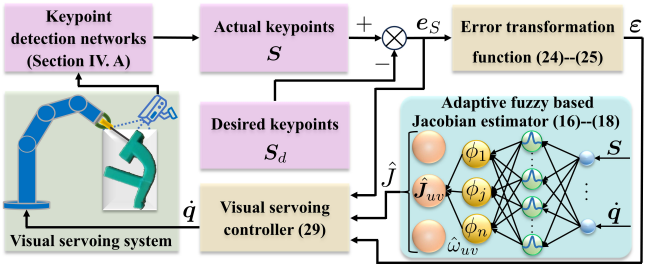


Fig. 4. Block diagram of the proposed control method.

Stage 4: Final Bonding. A bonding process is used to assemble the PDL and SL to form the complete PSA, as shown in Fig. 3(k). The entire fabrication process ensures the structural integrity and functional reliability of the PSA.

IV. Methodology

In this section, a keypoint detection method based on deep learning is proposed. Then, an adaptive fuzzy system is deployed to estimate the Jacobian matrix. Afterwards, PPC is introduced to ensure keypoints are within the spatial constraints, and the corresponding stability analysis is provided. The overall block diagram is depicted in Fig. 4.

A. Self-supervised Keypoint Detection

Without the complex and time-consuming human annotations, a self-supervised network is proposed to detect a desired number of keypoints from deformable objects, which is depicted in Fig. 5. Overall, a sequence of point clouds is fed into the proposed network, which contains two parts: encoder and decoder, as shown in Fig. 6. The encoder is composed of enhanced PointNet++ layers, with a Softmax activation function applied in the final layer. The encoder produces K probability values for each point, indicating its probability to be one of the K keypoints. The keypoint positions for each point cloud are computed and utilized to form grid heatmaps. In the decoder, the WFA modules, grid heatmaps, and multilayer perceptron (MLP) are utilized to reconstruct the input point clouds.

Given an input point cloud $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N]^\top \in \Re^{N\times 3}$ where $\boldsymbol{x}_i \in \Re^3$, $i \in N$, a weight matrix $W \in \Re^{K\times N}$ is produced by the encoder. By the matrix multiplication of W and X, the predicted K keypoints $\kappa = [\boldsymbol{\kappa}_1, \boldsymbol{\kappa}_2, \cdots, \boldsymbol{\kappa}_K]^\top \in \Re^{K\times 3}$ are derived as follows:

$$\begin{cases} \kappa = WX, \\ W = \text{Softmax}(\varphi_{en}), \end{cases} \tag{1}$$

where $\varphi_{en} \in \Re^{K\times N}$ represents the final output of the enhanced PointNet++, obtained through a hierarchical feature propagation module [37] that includes interpolation operations and unit PointNet layers. The input to the hierarchical feature propagation module is $\psi_{en}^{(H)} \in \Re^{N_H\times F_H}$, where $\psi_{en}^{(i)} \in \Re^{N_i\times F_i}$ denotes the feature of a hierarchically downsampled point cloud $X_{en}^{(i)} \in \Re^{N_i\times 3}$ at the i-th WFA layer, with $i = 1, 2, \cdots, H$. The WFA module more effectively balances the fusion of various useful information within local regions, and preserves point cloud features that are crucial for keypoint localization, as depicted in Fig. 6. Let N_1 denote the number of input points, and the set of sampled points is

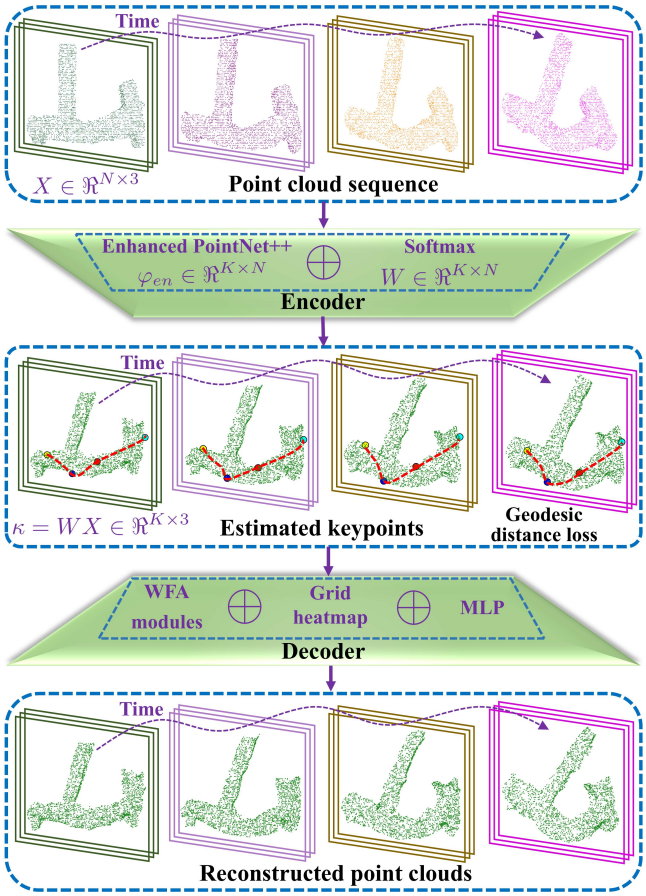


Fig. 5. Proposed self-supervised network for keypoint detection. A sequence of point clouds is fed into the keypoint estimation networks, that is, the encoder. The encoder is composed of enhanced PointNet++, with a Softmax activation function applied in the final layer. The encoder produces K probability values for each point, indicating its probability to be one of the K keypoints. The keypoint positions for each point cloud are computed by (1) and utilized to form grid heatmaps. In the decoder, the WFA modules, grid heatmaps, and MLP are used to reconstruct the input point clouds. To ensure the semantic consistency of keypoints, a geodesic distance loss function is introduced along all frames.

represented as $Y = [\boldsymbol{Y}_1, \boldsymbol{Y}_2, \cdots, \boldsymbol{Y}_i, \cdots, \boldsymbol{Y}_{N_2}]^\top \in \Re^{N_2\times 3}$. Then, the neighborhood point set for each sampled point $\boldsymbol{Y}_i$ stands for $Y^i = [\boldsymbol{Y}_{i1}, \boldsymbol{Y}_{i2}, \cdots, \boldsymbol{Y}_{ij}, \cdots, \boldsymbol{Y}_{ik}]^\top \in \Re^{k\times 3}$. Each neighborhood point $\boldsymbol{Y}_{ij}$ has a feature vector $\boldsymbol{F}_{1,ij}$ extracted from the previous layer, where the coordinate dimension is d, and the feature dimension is F_1. Hence, followed by subsampling and grouping, the input of the WFA module is $N_2 \times k \times (d + F_1)$, and the new F_2-dimensional feature $\boldsymbol{F}'_{2,ij}$ for each neighborhood point $\boldsymbol{Y}_{ij}$ is derived as [19]

$$\boldsymbol{F}'_{2,ij} = \mathrm{MLP}[(\boldsymbol{Y}_{ij} - \boldsymbol{Y}_i) \oplus \boldsymbol{F}_{1,ij}], \tag{2}$$

where $\boldsymbol{Y}_{ij} - \boldsymbol{Y}_i$ denotes the coordinate difference between the neighborhood point and the sampled point, indicating that the coordinates of points within each spherical neighborhood are normalized with respect to the center point. After extracting features $\boldsymbol{F}'_{2,ij}$, the WFA module is employed to aggregate these point features into global features for the local region. The WFA module performs weighted summation by learning weights for each feature in the sub-network. The weight vector $\boldsymbol{w}_{ij}$ is learned from both the coordinates of each point $\boldsymbol{Y}_{ij}$ and its corresponding learned features $\boldsymbol{F}'_{2,ij}$. $\boldsymbol{w}_{ij}$ has the same dimension as $\boldsymbol{F}'_{2,ij}$ and is computed as follows:

$$\boldsymbol{w}_{ij} = \mathrm{MLP}\left[(\boldsymbol{Y}_{ij} - \boldsymbol{Y}_i) \oplus \left(\boldsymbol{F}'_{2,ij} - \overline{\boldsymbol{F}'_i}\right)\right], \tag{3}$$

where $\overline{\boldsymbol{F}'_i}$ stands for the average of all features $\boldsymbol{F}'_{2,ij}$ in the spherical region. The global feature $\boldsymbol{F}_{2,i}$ for the local region is derived through a weighted summation of point features $\boldsymbol{F}'_{2,ij}$ and their corresponding weight vectors $\boldsymbol{w}_{ij}$. Specifically, $\boldsymbol{F}_{2,i}$ is formulated as follows:

$$\boldsymbol{F}_{2,i} = \sum_{j=1}^{k} \boldsymbol{w}_{ij} \odot \boldsymbol{F}'_{2,ij}. \tag{4}$$

Unlike PointNet, which employs simple pooling operations to aggregate local features into global features, the WFA module provides effective weighting of each point feature's contribution to the global feature. Hence, the WFA module preserves essential information and improves the detection accuracy of keypoints.

Then, connecting each pair of keypoints, a skeleton of the input point cloud is constructed, consisting of $C_K^2 = K(K-1)/2$ edges. Among them, the edges are expressed as $(\boldsymbol{\kappa}_i, \boldsymbol{\kappa}_j)$ to connect the keypoints $\boldsymbol{\kappa}_i \in \Re^3$ and $\boldsymbol{\kappa}_j \in \Re^3$. Meanwhile, s_{ij} is the weight of the edges $(\boldsymbol{\kappa}_i, \boldsymbol{\kappa}_j)$. Subsequently, a grid-based heatmap is employed to densely represent the 3-D shape using the information from the predicted keypoints κ. A 3-D array of grid points $G \in \Re^{4096\times 3}$ is defined in the normalized cubic 3-D space. Then, the distance $d_{ij}(\boldsymbol{g})$ between a grid point $\boldsymbol{g} \in \Re^3$ where $\boldsymbol{g} \in G$ and an edge of skeleton $(\boldsymbol{\kappa}_i, \boldsymbol{\kappa}_j)$ is defined as [21]

$$d_{ij}(\boldsymbol{g}) = \|\boldsymbol{g} - \boldsymbol{g}_{proj}\|, \tag{5}$$

where

$$\boldsymbol{g}_{proj} = \begin{cases} \boldsymbol{\kappa}_i, & \alpha \le 0, \\ (1-\alpha)\boldsymbol{\kappa}_i + \alpha\boldsymbol{\kappa}_j, & 0 < \alpha < 1, \\ \boldsymbol{\kappa}_j, & \alpha \ge 1, \end{cases} \tag{6}$$

$$\alpha = \frac{(\boldsymbol{g} - \boldsymbol{\kappa}_i)\cdot(\boldsymbol{\kappa}_j - \boldsymbol{\kappa}_i)}{\|\boldsymbol{\kappa}_i - \boldsymbol{\kappa}_j\|_2^2} \in \Re. \tag{7}$$

When $0 < \alpha < 1$, the projection point $\boldsymbol{g}_{proj} \in \Re^3$ is located within the edge $(\boldsymbol{\kappa}_i, \boldsymbol{\kappa}_j)$. When $\alpha \le 0$, the projection point is on the extension line of the edge, close to $\boldsymbol{\kappa}_i$. When $\alpha \ge 1$, the projection point is on the extension line of the edge, close to $\boldsymbol{\kappa}_j$. Afterwards, the feature of each grid point $D(\boldsymbol{g})$ is expressed as the maximum of the weighted distances from this point to the edges of the skeleton, and $D(\boldsymbol{g})$ is defined as

$$D(\boldsymbol{g}) = \max_{ij} \left\{ s_{ij} \exp\left(d_{ij}^2(\boldsymbol{g})/\nu^2\right) \right\}, \tag{8}$$

where ν represents a hyper-parameter. Meanwhile, the grid heatmap $\boldsymbol{\Gamma}$ denotes the 3-D array containing the features of all the grid points, which is defined as follows:

$$\boldsymbol{\Gamma} = [D(\boldsymbol{G}_{xyz})]_{x,y,z=1,2,\cdots,16} \in \Re^{4096\times 1}, \tag{9}$$

where $\boldsymbol{G}_{xyz} \in \Re^3$ stands for the extracted grid point coordinate from G. Consequently, the grid heatmap $\boldsymbol{\Gamma}$ contains the complete geometric information of the keypoints κ.

Based on the encoder, the decoder ψ_{de} consists of the H MLPs to reconstruct the 3-D shape of the input point clouds by incrementally refining geometric details in a hierarchical method. Formally, the feature of the $(H-i+1)$-th layer of the decoder $\psi_{de}^{(H-i+1)}$ is defined as [21]

$$\psi_{de}^{(H-i+1)} = \boldsymbol{\Gamma}(X_{en}^{(i)}) \oplus \psi_{en}^{(i)} \oplus \mathrm{Proj}(\cdot), \tag{10}$$

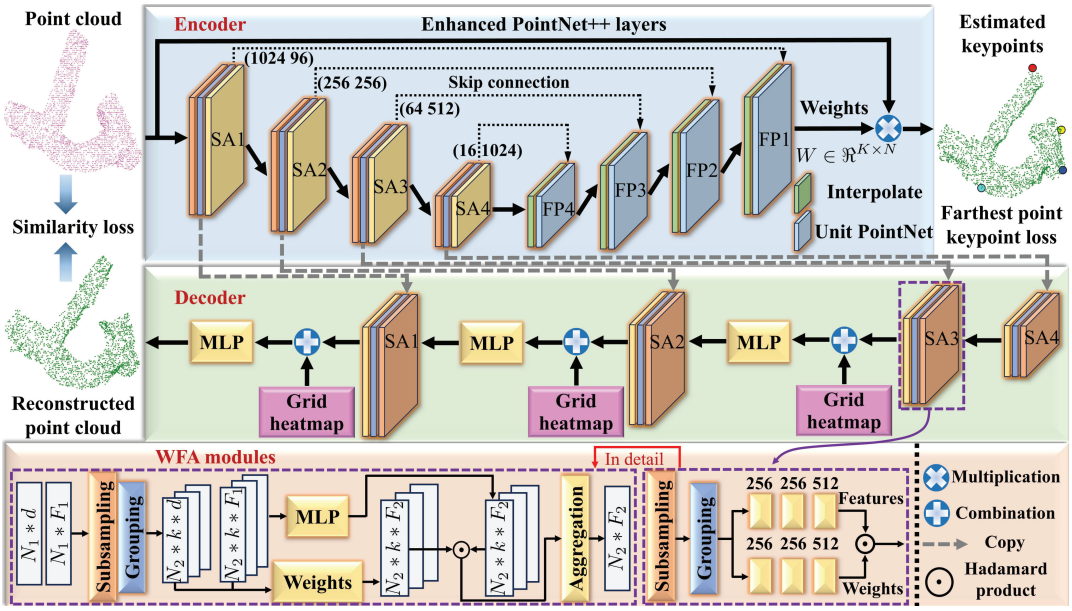


Fig. 6. Specific structure of the encoder and decoder at frame T.

where $\mathbf{\Gamma}(X_{en}^{(i)}) \in \Re^{N_i}$ is the grid heatmap of the extracted features indexed by $X_{en}^{(i)} \in \Re^{N_i \times 3}$, $\psi_{en}^{(i)} \in \Re^{N_i \times F_i}$ represents the feature of the point cloud derived from the i-th encoder, and $\mathrm{Proj}(\cdot) = \mathrm{Proj}(\psi_{de}^{(H-i)}, X_{en}^{(i-1)}, X_{en}^{(i)})$ is the feature projection operator from the previous layer of the decoder.

Furthermore, to ensure the semantic consistency of keypoints during object deformation, a geodesic distance loss function is proposed to maintain the desired geodesic distances between the estimated keypoints along all frames. During training, it is assumed that T consecutive point clouds sequences $[X(1), X(2), \cdots, X(T)]$ can be obtained, where $X(t) = [\boldsymbol{x}_1(t), \boldsymbol{x}_2(t), \cdots, \boldsymbol{x}_N(t)]^\top$ denotes the point cloud corresponding to the deformable object under relative motion at frame t and $t \in T$. The weight matrix corresponding to the j-th keypoint $\boldsymbol{\kappa}_j$ at frame t is denoted as $\boldsymbol{W}_{\boldsymbol{\kappa}_j}(\boldsymbol{x}_i(t)) \in \Re^{1\times N}$, where $i \in N$ and $j \in K$, representing the probability of each point $\boldsymbol{x}_i$ to be the j-th keypoint. Given the point cloud $X(t)$ at frame t, the desired geodesic distance between two keypoints is defined as follows [23]:

$$E_{X(t)}[d_{X(t)}(\boldsymbol{\kappa}_i, \boldsymbol{\kappa}_j)] = \sum_{m,n} \boldsymbol{W}_{\boldsymbol{\kappa}_i}(\boldsymbol{x}_m(t)) d_{X(t)}(\cdot,\cdot) \boldsymbol{W}_{\boldsymbol{\kappa}_j}^\top(\boldsymbol{x}_n(t)), \tag{11}$$

where $d_{X(t)}(\cdot,\cdot) = d_{X(t)}(\boldsymbol{x}_m(t), \boldsymbol{x}_n(t))$ denotes the precomputed geodesic distance between two points in $X(t)$, and $m, n \in N$. Then, all the keypoint probabilities are stacked row-wise in a matrix $W(t) \in \Re^{K\times N}$. By organizing the whole pairwise geodesic distances in a matrix $I(t) \in \Re^{N\times N}$ with elements $I(i,j)(t) = d_{X(t)}(\boldsymbol{x}_i(t), \boldsymbol{x}_j(t))$, the matrix of the desired distances between whole keypoint pairs can be constructed as $W(t)I(t)W^\top(t)$. Then, the geodesic distance loss is derived as follows [23]:

$$\mathcal{L}_{geo} = \sum_{\substack{t_1,t_2=1 \\ t_1\neq t_2}}^{T} \left\| W(t_1)I(t_1)W^\top(t_1) - W(t_2)I(t_2)W^\top(t_2) \right\|^2. \tag{12}$$

The geodesic distance loss is employed only during the training phase. Moreover, $I(t)$ is only required during training and can be computed offline during the dataset preprocessing. In this paper, the keypoints extracted from the last frame of the point cloud sequence during training are utilized as the desired keypoints $\boldsymbol{S}_d$[3]. The overall loss function is the weighted sum of three components, which is obtained as

$$\mathcal{L}_{all} = \lambda_{geo}\mathcal{L}_{geo} + \lambda_{sim}\mathcal{L}_{sim} + \lambda_{far}\mathcal{L}_{far}, \tag{13}$$

where λ_{geo}, λ_{sim}, and λ_{far} represent the scalar loss coefficients, $\mathcal{L}_{sim}$ is the similarity loss, and $\mathcal{L}_{far}$ stands for the farthest point keypoint loss. Precisely, the Chamfer distance between the reconstructed point cloud X_r and the input point cloud X is introduced to approximate $\mathcal{L}_{sim}$. $\mathcal{L}_{far}$ guarantees that the keypoints are well-distributed on the object surface and effectively capture the object's geometric structure. Please refer to [37] for the detailed principles and computation process of $\mathcal{L}_{sim}$ and $\mathcal{L}_{far}$.

B. Adaptive Fuzzy Based Jacobian Estimator

The Jacobian matrix $J \in \Re^{3a\times 6}$ can be denoted as

$$\dot{\boldsymbol{S}} = J\dot{\boldsymbol{q}} = \begin{bmatrix} \dot{\boldsymbol{S}}_1 \\ \vdots \\ \dot{\boldsymbol{S}}_u \\ \vdots \\ \dot{\boldsymbol{S}}_a \end{bmatrix} = \begin{bmatrix} \boldsymbol{J}_{11} & \cdots & \boldsymbol{J}_{1v} & \cdots & \boldsymbol{J}_{16} \\ \vdots & \ddots & \vdots & & \vdots \\ \boldsymbol{J}_{u1} & \cdots & \boldsymbol{J}_{uv} & \cdots & \boldsymbol{J}_{u6} \\ \vdots & & \vdots & \ddots & \vdots \\ \boldsymbol{J}_{a1} & \cdots & \boldsymbol{J}_{av} & \cdots & \boldsymbol{J}_{a6} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_v \\ \vdots \\ \dot{q}_6 \end{bmatrix}, \tag{14}$$

[3]In the training phase, K keypoints are detected from the final frame of the point cloud data. Then, a keypoints are selected within the region of interest (ROI) to serve as features for different tasks. The ROI and a can be flexibly determined manually based on the practical tasks. For ROI selection, we prioritize regions with significant deformation, and for tasks involving multiple objects, separate ROIs are defined for each object. For a selection, the selected keypoints must first be located within the ROI with a reasonable spatial distribution. Second, the value of a should balance deformation representation and control complexity. Third, the selected a keypoints should include both keypoints at significantly deformed locations and relatively static keypoints to constrain undesired deformation. Fourth, the selected a keypoints should remain reliably detectable without occlusion throughout DOM tasks. Since this process involves multiple coupled and task-dependent constraints, automatic selection remains challenging and is left for future work.

where $\dot{\boldsymbol{S}} \in \Re^{3a}$ denotes the motion velocity of the keypoints and $\boldsymbol{J}_{uv} \in \Re^3$. For online estimation of the Jacobian matrix, an adaptive fuzzy system is developed as follows:

$$\boldsymbol{J}_{uv} = \omega_{uv}^{*\top} \boldsymbol{\phi}(\boldsymbol{\delta}) + \boldsymbol{\upsilon}_{uv}, \tag{15}$$

where $\omega_{uv}^* \in \Re^{n\times 3}$ is the optimal weight matrix, $\boldsymbol{\phi}(\boldsymbol{\delta}) = [\phi_1(\boldsymbol{\delta}), \phi_2(\boldsymbol{\delta}), \cdots, \phi_j(\boldsymbol{\delta}), \cdots, \phi_n(\boldsymbol{\delta})]^\top \in \Re^n$ denotes the fuzzy basis function, $\boldsymbol{\delta} = \left[\boldsymbol{S}^\top, \boldsymbol{q}^\top\right]^\top = [\delta_1, \delta_2, \cdots, \delta_i, \cdots, \delta_{3a+6}]^\top \in \Re^{3a+6}$ represents the input vector, and $\boldsymbol{\upsilon}_{uv} \in \Re^3$ denotes the estimation error vector for the (u, v)-th element of the Jacobian matrix J, and $\Upsilon \in \Re^{3a\times 6}$ represents the overall estimation error matrix with the same structure as J, where

$$\Upsilon = \begin{bmatrix} \boldsymbol{\upsilon}_{11} & \cdots & \boldsymbol{\upsilon}_{1v} & \cdots & \boldsymbol{\upsilon}_{16} \\ \vdots & \ddots & \vdots & & \vdots \\ \boldsymbol{\upsilon}_{u1} & \cdots & \boldsymbol{\upsilon}_{uv} & \cdots & \boldsymbol{\upsilon}_{u6} \\ \vdots & & \vdots & \ddots & \vdots \\ \boldsymbol{\upsilon}_{a1} & \cdots & \boldsymbol{\upsilon}_{av} & \cdots & \boldsymbol{\upsilon}_{a6} \end{bmatrix}.$$

$\boldsymbol{\upsilon}_{uv}$ and ω_{uv}^* have upper bounds, i.e., $\|\omega_{uv}^*\| \le \bar{\omega}$ and $\|\boldsymbol{\upsilon}_{uv}\| \le \bar{\upsilon}$. The fuzzy basis function $\boldsymbol{\phi}(\boldsymbol{\delta})$ is defined as [38], [39]

$$\phi_j(\boldsymbol{\delta}) = \frac{\Pi_{i=1}^{3a+6} \mu_i^j(\delta_i)}{\sum_{j=1}^n \left[\Pi_{i=1}^{3a+6} \mu_i^j(\delta_i)\right]}, \tag{16}$$

where n represents the number of fuzzy rules and the Gaussian membership function of the fuzzy sets $\mu_i^j(\delta_i)$ is defined as

$$\mu_i^j(\delta_i) = \exp\left[-\left(\frac{\delta_i - c_{ij}}{b_j}\right)^2\right], \tag{17}$$

where c_{ij} and b_j denote the center and width of the membership function, respectively. Subsequently, the (u, v)-th block of the estimated Jacobian matrix $\hat{J} \in \Re^{3a\times 6}$ is expressed as

$$\hat{\boldsymbol{J}}_{uv} = \hat{\omega}_{uv}^\top \boldsymbol{\phi}(\boldsymbol{\delta}), \tag{18}$$

where $\hat{\boldsymbol{J}}_{uv} \in \Re^3$. The (u, v)-th block of estimation error matrix $\tilde{J} \in \Re^{3a\times 6}$ is obtained as

$$\tilde{\boldsymbol{J}}_{uv} = \boldsymbol{J}_{uv} - \hat{\boldsymbol{J}}_{uv} = \tilde{\omega}_{uv}^\top \boldsymbol{\phi}(\boldsymbol{\delta}) + \boldsymbol{\upsilon}_{uv}, \tag{19}$$

where $\tilde{\boldsymbol{J}}_{uv} \in \Re^3$ and $\tilde{\omega}_{uv} = \omega_{uv}^* - \hat{\omega}_{uv}$ is the estimation error of the weight matrix. Since we detect keypoints from the final frame of the point cloud data as the desired keypoints $\boldsymbol{S}_d$, $\boldsymbol{S}_d$ is stationary, i.e., $\dot{\boldsymbol{S}}_d = \boldsymbol{0}$. Hence, we can obtain

$$\dot{\boldsymbol{e}}_S = \dot{\boldsymbol{S}} = J\dot{\boldsymbol{q}} = \hat{J}\dot{\boldsymbol{q}} + \Delta\dot{\boldsymbol{P}}_n, \tag{20}$$

where $\Delta\dot{\boldsymbol{P}}_n = \left[\Delta\dot{\boldsymbol{P}}_{n1}^\top, \Delta\dot{\boldsymbol{P}}_{n2}^\top, \cdots, \Delta\dot{\boldsymbol{P}}_{nk}^\top, \cdots, \Delta\dot{\boldsymbol{P}}_{na}^\top\right]^\top \in \Re^{3a}$ and $\Delta\dot{\boldsymbol{P}}_n = J\dot{\boldsymbol{q}} - \hat{J}\dot{\boldsymbol{q}} = \tilde{J}\dot{\boldsymbol{q}}$. Then, we can derive

$$\Delta\dot{\boldsymbol{P}}_{nk} = \left(\tilde{\omega}_{uv}^\top \boldsymbol{\phi}(\boldsymbol{\delta}) + \boldsymbol{\upsilon}_{uv}\right) \dot{q}_v, \tag{21}$$

where $\Delta\dot{\boldsymbol{P}}_{nk} \in \Re^3$.

C. Visual Servoing Controller

To maintain the control performance of the robotic arm, PPC is implemented to bound the keypoint errors within predefined funnel constraints, ensuring the keypoints remain within the spatial constraints. Based on this, the specific constraints for the keypoint errors are formulated as follows:

$$\begin{cases} -\mu_u \rho_u(t) < e_{Su} < \rho_u(t), e_{Su} \ge 0, \\ -\rho_u(t) < e_{Su} < \mu_u \rho_u(t), e_{Su} < 0, \end{cases} \tag{22}$$

where $u = 1, 2, \cdots, 3a$, $\mu_u \in [0, 1]$ is the adjustable coefficient, and exponentially decaying funnel functions $\rho_u(t)$ are defined for each element e_{Su} in the error vector $\boldsymbol{e}_S = [\boldsymbol{e}_{S1}, \boldsymbol{e}_{S2}, \cdots, \boldsymbol{e}_{Su}, \cdots, \boldsymbol{e}_{Sa}]^\top = [e_{S1}, e_{S2}, \cdots, e_{Su}, \cdots, e_{S3a-1}, e_{S3a}]^\top \in \Re^{3a}$ as [28]

$$\rho_u(t) = (\rho_{0u} - \rho_{\infty u}) \exp(-\tau_u t) + \rho_{\infty u}, \tag{23}$$

where ρ_{0u}, $\rho_{\infty u}$, and τ_u are positive constants satisfying $\rho_{0u} > \rho_{\infty u}$. The convergence rate of $\rho_u(t)$ can be modified by adjusting the value of τ_u. Combined with $\rho_u(t)$ and e_{Su}, the error transformation function $\psi_u(\varepsilon_u)$ is introduced as [40]

$$e_{Su} = \psi_u(\varepsilon_u)\rho_u(t), \tag{24}$$

$$\psi_u(\varepsilon_u) = \frac{\overline{\delta}_u \exp(\varepsilon_u) - \underline{\delta}_u}{\exp(\varepsilon_u) + 1}, \tag{25}$$

where $-\underline{\delta}_u < \psi_u(\varepsilon_u) < \overline{\delta}_u$ with $\underline{\delta}_u$ and $\overline{\delta}_u$ are positive constants that satisfy the following conditions:

$$(-\underline{\delta}_u, \overline{\delta}_u) = \begin{cases} (-\mu_u, 1), & e_{Su}(0) \ge 0, \\ (-1, \mu_u), & e_{Su}(0) < 0. \end{cases} \tag{26}$$

According to (24) and (25), the u-th element of the transferred errors $\boldsymbol{\varepsilon} = [\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_u, \cdots, \varepsilon_{3a}]^\top$ can be derived as

$$\varepsilon_u = \ln(e_{Su} + \underline{\delta}_u \rho_u(t)) - \ln(\overline{\delta}_u \rho_u(t) - e_{Su}). \tag{27}$$

Then, by taking the derivative of ε_u, we can obtain

$$\dot{\varepsilon}_u = r_u(\rho_u(t)\dot{e}_{Su} - \dot{\rho}_u(t)e_{Su}), \tag{28}$$

where $R = \mathrm{diag}(r_1, r_2, \cdots, r_u, \cdots, r_{3a}) \in \Re^{3a\times 3a}$ and $\theta = \mathrm{diag}(\rho_1(t), \rho_2(t), \cdots, \rho_u(t), \cdots, \rho_{3a}(t)) \in \Re^{3a\times 3a}$. The specific expression for r_u is given as follows:

$$r_u = \frac{\underline{\delta}_u + \overline{\delta}_u}{(\underline{\delta}_u \rho_u(t) + e_{Su})(\overline{\delta}_u \rho_u(t) - e_{Su})}.$$

Consequently, the visual servoing controller is designed as

$$\dot{\boldsymbol{q}} = -\hat{J}^+ \left[-\frac{\dot{\theta}}{\theta}\boldsymbol{e}_S + k_1 \frac{1}{R\theta}\boldsymbol{\varepsilon} + k_2 \tanh(\boldsymbol{\varepsilon})\right], \tag{29}$$

where $k_1, k_2 \in \Re^{3a\times 3a}$ denote positive diagonal control gain matrices and $\hat{J}^+ \in \Re^{6\times 3a}$ stands for the pseudo-inverse of the estimated Jacobian matrix. Meanwhile, the adaptive update laws of $\hat{\omega}_{uv}$ are designed as follows:

$$\dot{\hat{\boldsymbol{\omega}}}_{uv,i} = \boldsymbol{\phi}(\boldsymbol{\delta})\dot{q}_v(r_u\rho_u(t)\varepsilon_u) - \eta\hat{\boldsymbol{\omega}}_{uv,i}, \tag{30}$$

where $\dot{\hat{\omega}}_{uv} = \left[\dot{\hat{\boldsymbol{\omega}}}_{uv,1}, \dot{\hat{\boldsymbol{\omega}}}_{uv,2}, \dot{\hat{\boldsymbol{\omega}}}_{uv,3}\right] \in \Re^{n\times 3}$, $i = 1, 2, 3$, and $\eta > 0$ is a positive constant.

D. Stability Analysis

Theorem 1: The proposed controller (29), adaptive fuzzy estimator (18), and update laws (30) guarantee that keypoint errors converge to small neighborhoods of the origin within the spatial constraints. Meanwhile, the prescribed performance (22) can be guaranteed when the initial errors of keypoints are within the funnel functions (23).

Proof: To prove Theorem 1, a Lyapunov function candidate is first selected as follows:

$$V_1 = \frac{1}{2}\boldsymbol{\varepsilon}^\top \boldsymbol{\varepsilon}. \tag{31}$$

By taking the time derivative of V_1 and substituting (20), (21), (28), (29), it can be derived that

$$\begin{aligned} \dot{V}_1 &= \boldsymbol{\varepsilon}^\top \dot{\boldsymbol{\varepsilon}} = \boldsymbol{\varepsilon}^\top R(\theta\dot{\boldsymbol{e}}_S - \dot{\theta}\boldsymbol{e}_S) \\ &= \boldsymbol{\varepsilon}^\top R[\theta(\hat{J}\dot{\boldsymbol{q}} + \Delta\dot{\boldsymbol{P}}_n) - \dot{\theta}\boldsymbol{e}_S] \\ &= -\boldsymbol{\varepsilon}^\top k_1 \boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^\top R\theta k_2 \tanh(\boldsymbol{\varepsilon}) + \boldsymbol{\varepsilon}^\top R\theta\tilde{J}\dot{\boldsymbol{q}}. \end{aligned} \tag{32}$$

Then, to prove the Lyapunov stability of the proposed controller, another Lyapunov function candidate is constructed as

$$V_2 = V_1 + \frac{1}{2}\sum_{v=1}^{6}\sum_{u=1}^{a}\sum_{i=1}^{3}\tilde{\boldsymbol{\omega}}_{uv,i}^{\top}\tilde{\boldsymbol{\omega}}_{uv,i}. \tag{33}$$

By taking the time derivative of V_2 and substituting (30), we can obtain that

$$\begin{aligned}\dot{V}_2 &= \dot{V}_1 - \sum_{v=1}^{6}\sum_{u=1}^{a}\sum_{i=1}^{3}\tilde{\boldsymbol{\omega}}_{uv,i}^{\top}\dot{\hat{\boldsymbol{\omega}}}_{uv,i}\\ &= \dot{V}_1 - \sum_{v=1}^{6}\sum_{u=1}^{a}\sum_{i=1}^{3}\left(\tilde{\boldsymbol{\omega}}_{uv,i}^{\top}\boldsymbol{\phi}(\boldsymbol{\delta})\dot{q}_v r_u \rho_u(t)\varepsilon_u - \eta\tilde{\boldsymbol{\omega}}_{uv,i}^{\top}\hat{\boldsymbol{\omega}}_{uv,i}\right)\\ &\le -\boldsymbol{\varepsilon}^{\top}k_1\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^{\top}R\theta k_2\,\mathbf{tanh}(\boldsymbol{\varepsilon}) + \left\|\boldsymbol{\varepsilon}^{\top}\right\|\|R\|\,\|\theta\|\,\bar{v}\,\|\dot{\boldsymbol{q}}\|\\ &\quad + \eta\sum_{v=1}^{6}\sum_{u=1}^{a}\sum_{i=1}^{3}\tilde{\boldsymbol{\omega}}_{uv,i}^{\top}\hat{\boldsymbol{\omega}}_{uv,i}.\end{aligned} \tag{34}$$

To ensure that the Jacobian matrix can effectively characterize the quasi-static elastic deformation dynamics of the object, the robotic arm operates at sufficiently low velocity, i.e., $\|\dot{\boldsymbol{q}}\|$ is a sufficiently small positive value. Based on the universal approximation theorem [41], $\bar{v}$ is also a sufficiently small positive value. Let the minimum eigenvalue of k_2 satisfy $\lambda_{\min}(k_2) \ge \bar{v}\|\dot{\boldsymbol{q}}\|$. Then, (34) can be further simplified as

$$\begin{aligned}\dot{V}_2 &\le -\boldsymbol{\varepsilon}^{\top}k_1\boldsymbol{\varepsilon} + \eta\sum_{v=1}^{6}\sum_{u=1}^{a}\sum_{i=1}^{3}\tilde{\boldsymbol{\omega}}_{uv,i}^{\top}\hat{\boldsymbol{\omega}}_{uv,i}\\ &\le -\boldsymbol{\varepsilon}^{\top}k_1\boldsymbol{\varepsilon} - \frac{\eta}{2}\sum_{v=1}^{6}\sum_{u=1}^{a}\sum_{i=1}^{3}\left(\|\tilde{\boldsymbol{\omega}}_{uv,i}\|^2 - \|\boldsymbol{\omega}_{uv,i}^{*}\|^2\right)\\ &\le -\chi V_2 + \beta,\end{aligned} \tag{35}$$

where $\chi = \min(2\gamma_{\max}(k_1), \eta)$, $\beta = \frac{\eta}{2}\sum_{v=1}^{6}\sum_{u=1}^{a}\sum_{i=1}^{3}\|\boldsymbol{\omega}_{uv,i}^{*}\|^2$, and $\gamma_{\max}(k_1)$ represents the maximum eigenvalue of k_1. By further solving inequality (35), it is not difficult to derive that

$$V_2(t) \le V_2(0)\mathrm{e}^{-\chi t} + \frac{\beta}{\chi}\left(1 - \mathrm{e}^{-\chi t}\right). \tag{36}$$

When the initial errors of keypoints $\boldsymbol{e}_S(0)$ are within the funnel functions (23), it can be deduced that

$$V_2 \in \mathcal{L}_\infty \Rightarrow \boldsymbol{e}_S, \dot{\boldsymbol{e}}_S, \boldsymbol{\varepsilon}, \dot{\boldsymbol{\varepsilon}}, \tilde{\boldsymbol{\omega}}_{uv,i} \in \mathcal{L}_\infty,\ i = 1, 2, 3. \tag{37}$$

Finally, it can be further concluded that

$$\lim_{t\to\infty}\|\boldsymbol{\varepsilon}(t)\| \le \sqrt{\frac{2\beta}{\chi}}. \tag{38}$$

Therefore, by utilizing controller (29), adaptive fuzzy estimator (18), and update laws (30), the keypoint errors converge to a small neighborhood of the origin. Furthermore, the result of $\boldsymbol{\varepsilon}, \dot{\boldsymbol{\varepsilon}} \in \mathcal{L}_\infty$ imply that (22) holds. If e_{Su} is to violate the constraints specified in (22), then by (27), ε_u would tend to infinity, contradicting the result in (37). By utilizing reduction to absurdity, $\boldsymbol{e}_S$ always remains within the prescribed constraints, thereby completing the proof of Theorem 1. ■

V. Experimental Results

A. Experimental Setup

Fig. 7 shows the experimental setup of four subsystems.

(1) **Host PC system**. The keypoint detection networks and visual servoing controller are implemented on a personal computer (PC) equipped with an Intel i7-13620H CPU, 16 GB of memory, and an NVIDIA GeForce RTX 4060 GPU. Communication among the PC, the Universal Robots 10e (UR10e), and the camera is managed using Robot Operating System 2 (ROS 2) on Ubuntu Linux. The programs for point cloud processing, keypoint detection network training, and visual servoing controller are all developed in Python, utilizing OpenCV, PyTorch framework, and URX library, respectively.

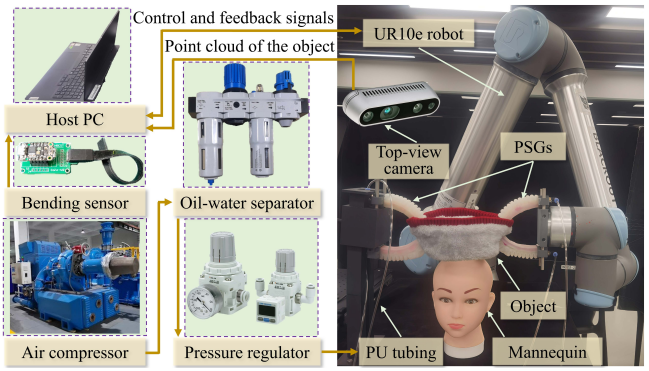


Fig. 7. Eye-to-hand robot platform with PSGs for DOM.

(2) **Sensing system**. The perception system includes bending sensors (1-Axis Flex Sensor Evaluation Kit, Bend Labs) and an Intel RealSense D435i camera. The bending sensors are attached to the inner side of the PSAs and can deform synchronously with the PSAs. The sensors transmit real-time bending angle data to the host PC via Bluetooth communication. Moreover, a camera is used to capture the point cloud of objects at a resolution of 640 × 480 in real-time.

(3) **Actuation system**. The actuation system consists of UR10e and PSGs. The UR10e is employed for the experimental verification of DOM (Section D) with an eye-to-hand configuration. To prevent object damage and ensure that the Jacobian matrix can accurately compute the dynamics of quasi-static elastic deformation of the object, the maximum joint angular velocity of the UR10e is limited to 0.04 rad/s[4].

(4) **Pneumatic supply system**. The air compressor compresses the surrounding air and stores it. The compressed air is then delivered to the oil-water separator through specific pipelines. The oil-water separator purifies the compressed air by removing oil and moisture contaminants, and the clean air is subsequently transmitted to the pressure regulator (SMC-IR20) via air tubing. The bending angle of the PSGs is controlled by adjusting the valve opening of the pressure regulator.

B. Performance Verification of PSGs

Experiment 1: Grasping performance verification.

[4]Through reasonable adjustment, the maximum joint angular velocity is selected as 0.04 rad/s. The excessively large maximum joint angular velocity causes the deformable objects to exhibit non-negligible dynamic and inertial effects, which violates the quasi-static assumption. Consequently, the Jacobian matrix can no longer accurately characterize the mapping between keypoint motion and robot joint motion, thereby increasing the estimation error $\bar{v}$. Conversely, the excessively small maximum joint angular velocity prevents the controller from providing sufficient corrective action. When the keypoint errors approach the funnel functions $\rho_u(t)$, the joint angular velocity needs to be increased rapidly to quickly force the keypoint back within the funnel functions. An overly restrictive velocity limit causes the control input saturation, potentially leading to violations of the predefined spatial constraints on the keypoint errors, which degrades the transient performance or even results in task failure.

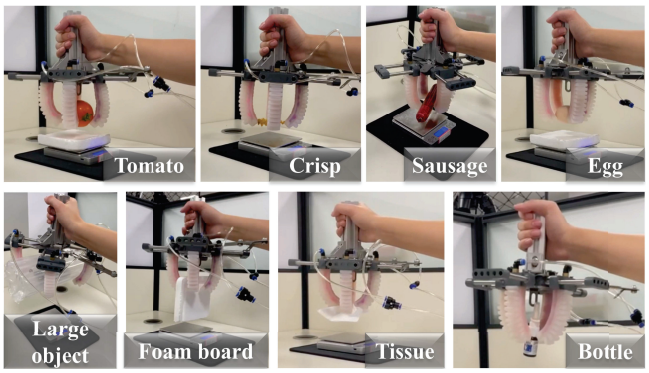


Fig. 8. Grasping performance experiments of different objects.

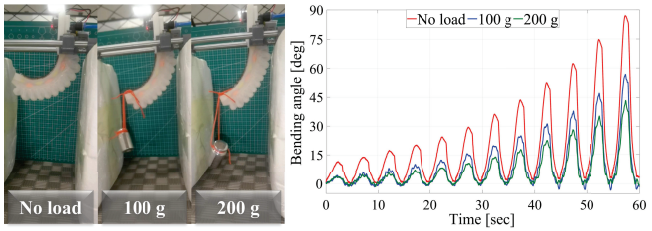


Fig. 9. Bending performance experiments of different payloads.

To verify the grasping performance of PSGs, eight different grasping experiments are conducted in Experiment 1. The experimental results demonstrate that the PSGs can firmly grasp various types of objects without relative displacement at certain heights. These objects include irregular, fragile, large, deformable, and smooth-surfaced items, as illustrated in Fig. 8. Notably, PSGs achieve damage-free grasping, particularly when handling fragile objects such as eggs, tissues, and crisps. Furthermore, when the experimenter shakes the PSGs, no relative displacement occurs between the objects and grippers. This performance is primarily attributed to the ROM in the SL, which provides necessary rigid support. The particle filling technique is then introduced to further enhance the stiffness of the PSGs, achieving uniform deformation of the grippers.

Experiment 2: Bending performance verification.

Experiment 2 validates the bending performance of the proposed gripper under different loads. A sinusoidal air pressure signal u with variable amplitude is given as

$$u = (0.65 + 0.01t)\,|\sin(0.2\pi t)|\,,\ t \in [0, 60].$$

As depicted in Fig. 9, the PSAs achieve a bending angle of 87.06 deg at 1.2 bar input pressure under no-load conditions. Under 100 g and 200 g loads, the PSAs achieve bending angles of 56.58 deg and 42.37 deg, respectively, at 1.2 bar input pressure. The experimental results demonstrate that the proposed PSGs maintain satisfactory bending performance and compliance while providing effective load-bearing capability.

C. Validation of Keypoint Detection

In this section, we compare the proposed networks with existing methods to validate the effectiveness and superiority of keypoint detection and 3-D point cloud reconstruction. The deformable object datasets are obtained from the experimental validation of DOM (Section D), including sponges, ropes, and scarves. Each dataset contains 600 sets of point clouds for training, consisting of 400 dynamic point clouds capturing slow motion from initial to desired positions and 200 static point clouds at the desired positions.

The data preprocessing consists of three main stages. First, RGB-D data of target objects are extracted from complex backgrounds using multi-stage filtering, including color filtering, depth filtering, and morphological operations. Color filtering combines HSV thresholding, RGB channel analysis, LAB processing, saturation calculation, and color ratio evaluation for robust segmentation. Depth filtering constrains detection within 0.2–0.6 m, while morphological operations generate binary masks based on area and color purity criteria. Second, point cloud processing applies statistical filtering (20–30 nearest neighbors, 3.0 standard deviation), voxel downsampling (3–5 mm), and uniform resampling to 2048 points with 0.5 mm Gaussian noise when duplication is needed. Third, the keypoint detection network is trained with Adadelta optimizer, batch size of 16, learning rate of 0.1, $\lambda_{sim} = 10$, $\lambda_{far} = 1$, and $\lambda_{geo} = 0.1$. Gradient clipping (maximum norm 1.0) and point cloud normalization ensure stable training. Then, 16 keypoints are detected from the point cloud sequence, with four keypoints selected within the ROI for different tasks. For fair comparison, four keypoints with the same indices are selected for all methods, and the keypoint detection results are visualized at the same instant.

Each dataset is trained five times, yielding average training times of 167 minutes, 117 minutes, and 135 minutes, respectively. The differences in training time are mainly due to their different geometric complexities and deformation characteristics. The sponge has the most complex surface geometry with a dispersed point cloud distribution, resulting in the highest computational cost for $I(t)$. The rope, as a linear object, has the simplest geometry and the lowest cost. The red region of the scarf, as a thin object, falls in between.

The proposed network is compared with Self-Geo [23], SC3K [22], and PointNet++ [37] to demonstrate its effectiveness and superiority, as illustrated in Fig. 10[5]. Results show that the proposed method achieves superior performance in point cloud 3-D reconstruction and keypoint detection. Specifically, the keypoints detected by the proposed method exhibit uniform spatial distribution across the point cloud surface and remain within the point cloud boundaries. Furthermore, the proposed method successfully estimates keypoints from point cloud sequences with excellent temporal and spatial consistency, which is crucial for shape control.

Remark 1: The selection of 4 keypoints from the detected 16 keypoints is based on two considerations. First, regarding the minimum dimensionality for controllability, inspired by [42], the dimension of the selected keypoints must exceed that of the control inputs to avoid rank deficiency and potential singularities of J. That is, shape control of a 6-DOF robotic arm requires at least three keypoints with the 3-D coordinates. Second, in terms of sufficient deformation representation with controlled complexity, the selected a keypoints must adequately capture the object's deformation to ensure convergence toward the desired shape. Conversely, too many keypoints not only increase the computational cost of $\hat{J}^{+}$ but also introduce redundant or noisy features, thereby reducing the control performance.

[5]Task 2 is the rope docking task. Four keypoints are selected, with one on the static rope and three on the deformable rope. For clarity of visualization, only the three keypoints on the deformable rope are presented in this paper.

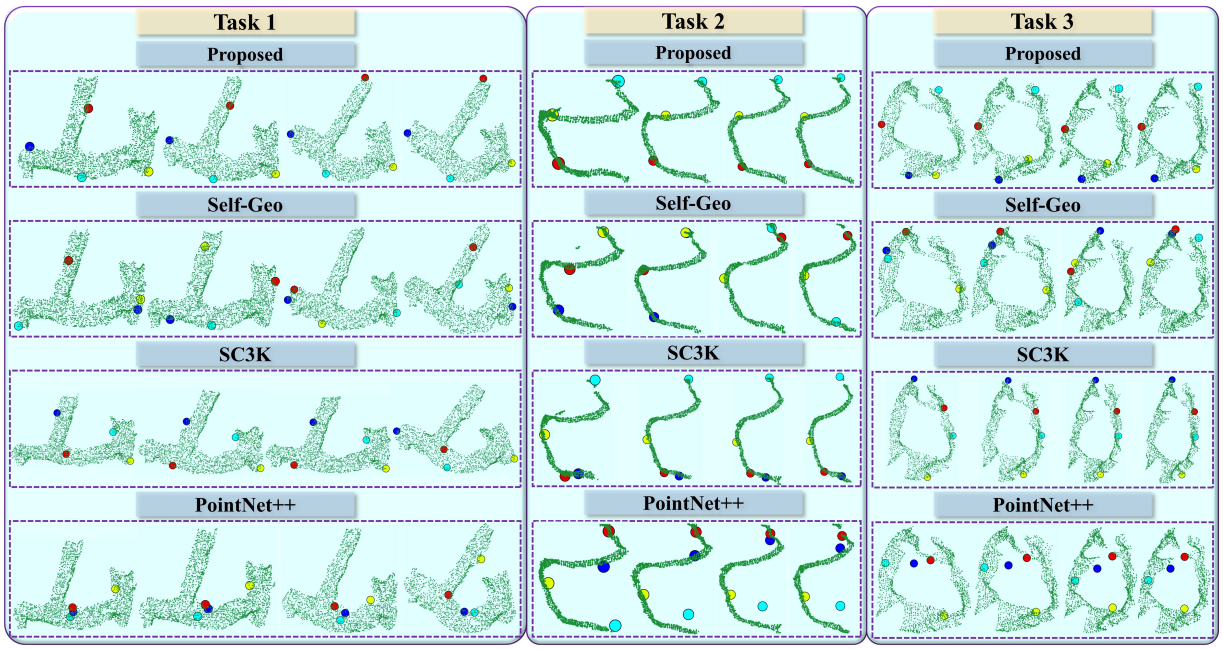


Fig. 10. Comparison of the keypoint detection.

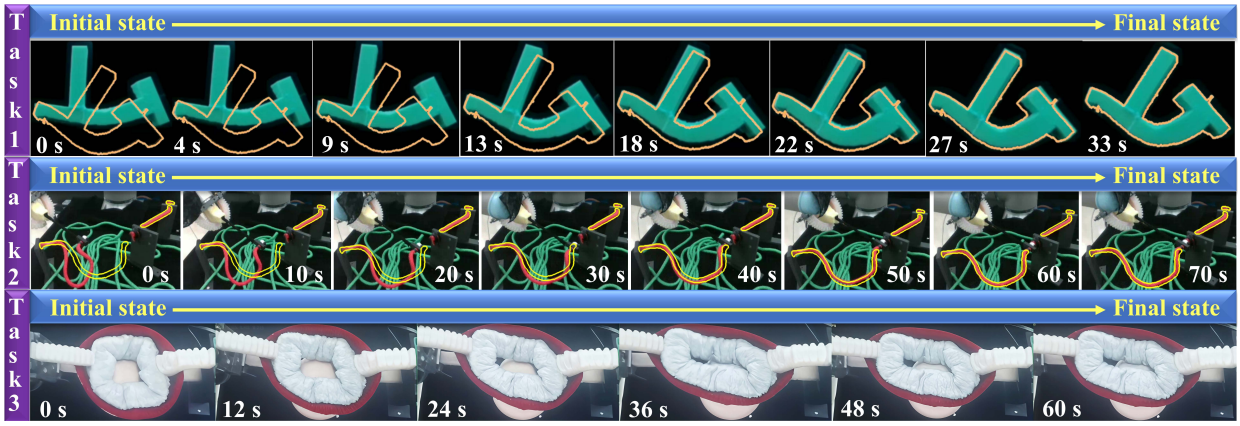


Fig. 11. Qualitative results of three tasks.

D. Experimental Verification of DOM

The parameters of the visual servoing controller are set as $k_1 = 0.12I_{12\times12}$, $k_2 = 0.01I_{12\times12}$. Funnel functions $\rho_u(t)$ are designed for the x, y, and z axes, i.e., $u = x, y, z$. In Task 1, $\rho_{0x} = 0.16$, $\rho_{0y} = \rho_{0z} = 0.08$, $\rho_{\infty x} = \rho_{\infty y} = \rho_{\infty z} = 0.01$, $\tau_x = 0.05$, $\tau_y = \tau_z = 0.07$. In Task 2, $\rho_{0x} = 0.12$, $\rho_{0y} = \rho_{0z} = 0.08$, $\rho_{\infty x} = \rho_{\infty z} = 0.012$, $\rho_{\infty y} = 0.015$, $\tau_x = \tau_y = \tau_z = 0.03$. In Task 3, $\rho_{0x} = 0.13$, $\rho_{0y} = \rho_{0z} = 0.08$, $\rho_{\infty x} = \rho_{\infty y} = \rho_{\infty z} = 0.02$, $\tau_x = \tau_y = \tau_z = 0.05$.

The robotic arm is controlled to move randomly within a region near its initial configuration, and the coordinates of keypoints and joint angles are recorded in real time as input to the fuzzy estimator. After recording 100 sets of input data, K-means clustering is applied to determine the center c of the membership functions. The width b is proportional to the average distance between adjacent cluster centers to ensure sufficient overlap between adjacent fuzzy basis functions, thereby fully covering the input. The number of fuzzy rules is set to $n = 9$ to balance the accuracy of the fuzzy estimator with real-time computational efficiency. The control frequency of the proposed method is approximately 4 Hz.

Task 1: Sponge morphing to target letters.

The aim of Task 1 is to manipulate the sponge to deform into a specified shape (letters IJ). The experimental process and control performance are shown in Figs. 11–13. The experimental results demonstrate that the total position error of keypoints steadily converges to 0.02 m at 27.3 seconds. Each keypoint moves towards the desired position without significant fluctuations, demonstrating satisfactory transient performance of the proposed method. Notably, the proposed method effectively ensures that the position errors of keypoints in the x, y, and z axes do not violate the preset constraint boundaries. When keypoints approach the constraint boundaries, the controller promptly adjusts the control inputs to ensure that the keypoints are within the predefined spatial funnel functions.

Task 2: Rope docking.

The purpose of Task 2 is to manipulate the rope to deform into a specified shape for docking with another rope. The ex-

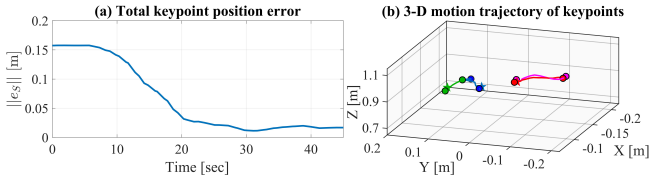


Fig. 12. Results of Task 1. In subfigure (b), ⋆ symbols are desired positions, ○ symbols are actual start/end positions of keypoints, and red, green, blue, purple solid lines are motion trajectories of keypoints 1–4, respectively.

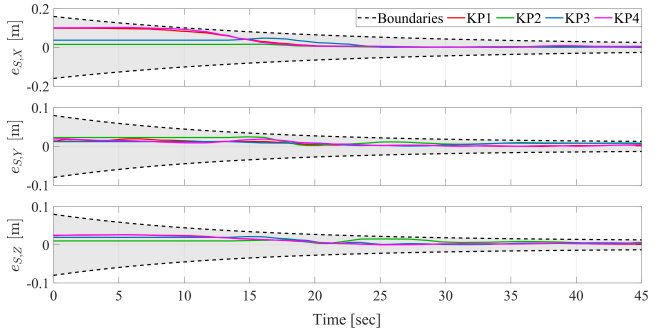


Fig. 13. Keypoint position errors in X, Y, and Z axes of Task 1.

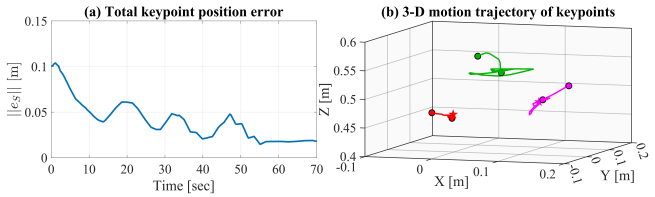


Fig. 14. Results of Task 2. In subfigure (b), ⋆ symbols are desired positions, ○ symbols are actual start/end positions of keypoints, and red, green, purple solid lines are motion trajectories of keypoints 1–3, respectively.

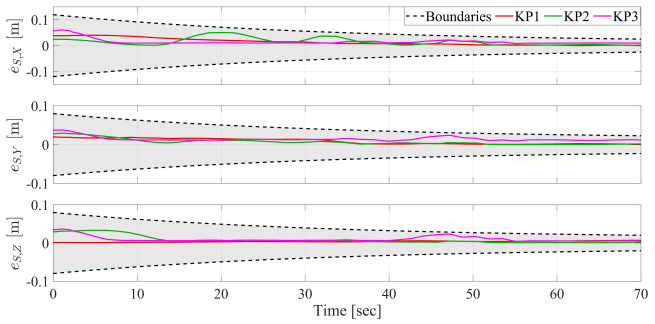


Fig. 15. Keypoint position errors in X, Y, and Z axes of Task 2.

perimental process and control performance are shown in Figs. 11, 14, and 15. Under green rope disturbances, the proposed point cloud processing method effectively extracts the point cloud of the red rope. The total position error of keypoints converges to 0.02 m at 54.3 seconds. It should be noted that the keypoint coordinates detected in real-time are sensitive along the depth direction, which tends to cause fluctuations in keypoint errors and consequently requires the robotic arm to perform multiple trial-and-error attempts during the docking process. Even so, the position errors of keypoints in the x, y, and z axes remain within the preset constraint boundaries, and the control performance meets practical requirements.

Task 3: Robot-assisted dressing.

The purpose of Task 3 is to manipulate the scarf to deform into a specified shape and then dress the mannequin with

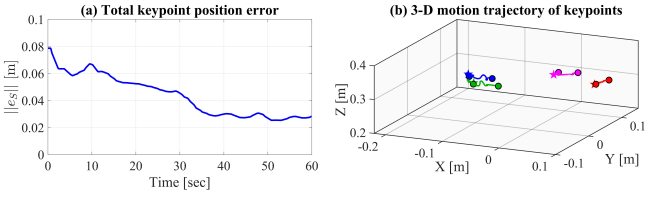


Fig. 16. Results of Task 3. In subfigure (b), ⋆ symbols are desired positions, ○ symbols are actual start/end positions of keypoints, and red, green, blue, purple solid lines are motion trajectories of keypoints 1–4, respectively.

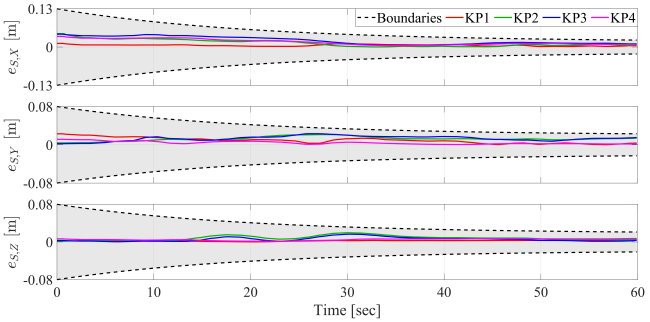


Fig. 17. Keypoint position errors in X, Y, and Z axes of Task 3.

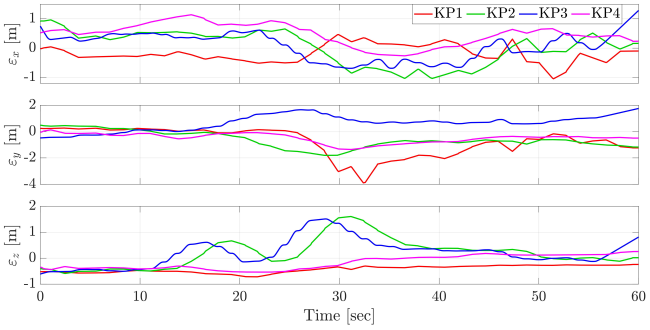


Fig. 18. Transferred errors in X, Y, and Z axes of Task 3.

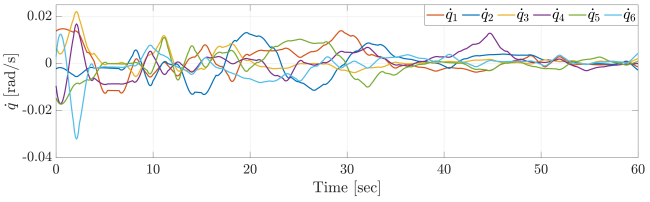


Fig. 19. Control inputs of Task 3.

the scarf, achieving robot-assisted dressing. The experimental process and control performance are shown in Figs. 11, 16–19. The total position error of keypoints converges to 0.03 m at 36.7 seconds, with each keypoint moving smoothly towards the desired position. The position errors of keypoints in the x, y, and z axes remain within the preset constraint boundaries. When the total position error remains below 0.03 m for 12 seconds, the scarf is considered to have achieved the specified shape, and the end-effector of the robotic arm descends by a predetermined height to complete the assisted dressing experiment. The proposed PSGs firmly grasp the scarf without relative displacement. More importantly, the grippers guarantee safe and compliant robot-human interaction. The fuzzy estimator exhibits satisfactory cross-task robustness. First, K-means clustering can determine the center c based on the input data distribution for each task, thus obtaining

the width b and reasonably allocating the fuzzy sets. Second, the adaptive update law optimizes the weight matrices online, which gradually reduces the estimation errors.

Remark 2: Although the proposed method is not robust to occlusion, this paper attempts to avoid occlusion from both hardware and software perspectives. From the hardware perspective, the positions of the soft grippers, the robotic arm, the objects, and the camera should be properly arranged when setting up the experimental platform. This can effectively alleviate occlusion during DOM tasks. Meanwhile, the grasping position of the soft grippers should be properly set so that it does not occlude the selected ROI and keypoints during DOM tasks. From the software perspective, the ROI and keypoints should be reasonably selected to avoid occlusion during the deformation process as much as possible.

Remark 3: In each task, to meet the conditions in (22), ρ_{0u} is set slightly larger than the initial errors $\boldsymbol{e}_s(0)$. Then, $\rho_{\infty u}$ is selected according to the control accuracy requirements of the specific task. The selection of τ_u is relatively conservative to ensure that the manipulation motion of the robotic arm is sufficiently slow under Assumption 1. The differences in parameter adjustment of funnel functions are primarily due to variations in object geometry, deformation amplitude, task requirements, and control accuracy requirements. Task 1 requires the larger ρ_{0u} due to its large initial deformation along the x-axis. In Task 2, as the keypoints are sensitive along the depth direction, we appropriately reduce τ_u, allowing the funnel functions to decrease more gradually, thereby ensuring satisfactory transient performance of the system.

E. Ablation Experiments

To validate the effectiveness of the proposed control method, ablation experiments are conducted by replacing the proposed control law with Broyden [26] and ENN [24] methods. The experimental results are presented in Figs. 20–25, and performance indicators and experimental results are summarized in Table I. In Table I, t_s denotes the settling time and $\bar{e}$ is the steady-state error of the total keypoint position error.

In Task 1, all three methods fulfill that keypoint errors converge to the neighborhood of the origin, as depicted in Fig. 20. However, the proposed method achieves a shorter settling time (27.3 s) compared with the ENN method (38.0 s), demonstrating satisfactory transient performance. Furthermore, the proposed method exhibits smaller steady-state error (0.02 m) than the Broyden (0.05 m) and ENN methods (0.04 m), while guaranteeing smooth convergence of keypoint errors. As shown in Fig. 21, the proposed method successfully manipulates the robotic arm to deform the sponge into the specified shape.

In Task 2, the proposed method achieves shorter settling time (54.3 s) compared with the Broyden (114.2 s) and ENN methods (83.8 s), demonstrating superior transient performance, as depicted in Fig. 22. Furthermore, it exhibits a smaller steady-state error (0.02 m) than the ENN method (0.04 m). Fig. 23 shows that the proposed method successfully controls the rope shape and completes the docking task.

In Task 3, the proposed method ensures keypoint errors smoothly converge to the neighborhood of the origin, as depicted in Fig. 24. Moreover, the proposed method achieves shorter settling time (36.7 s) compared with the Broyden (40.6 s) and ENN methods (41.3 s), and exhibits smaller steady-state error (0.03 m) than the above method (0.05 m). As shown in Fig. 25, the proposed method successfully manipulates the robotic arm to deform the scarf into the specified shape and complete the assisted dressing task.

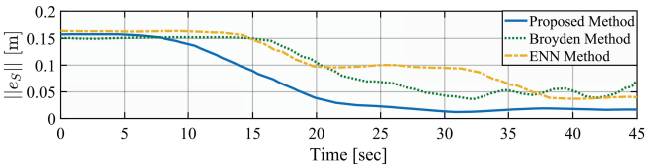


Fig. 20. Total position errors of keypoints with different methods in Task 1.

Fig. 21. Visualization results for different methods in Task 1.

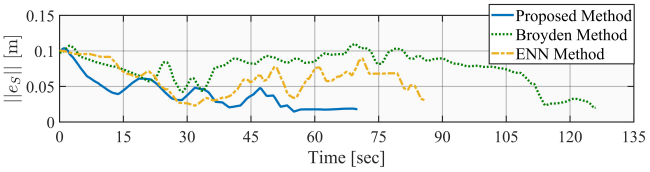


Fig. 22. Total position errors of keypoints with different methods in Task 2.

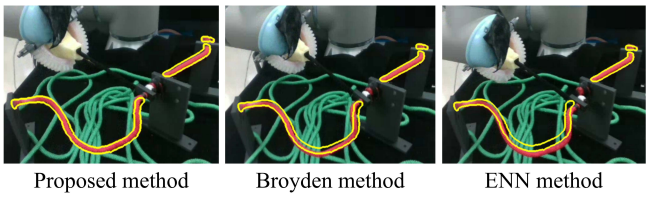


Fig. 23. Visualization results for different methods in Task 2.

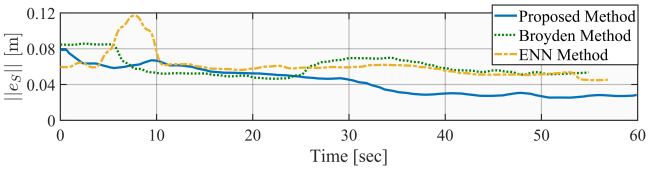


Fig. 24. Total position errors of keypoints with different methods in Task 3.

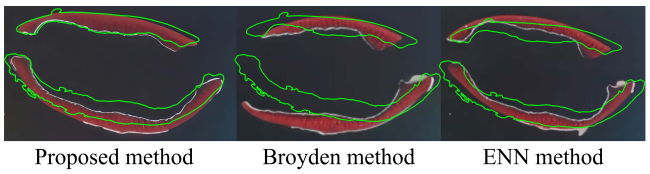


Fig. 25. Visualization results for different methods in Task 3.

VI. Conclusion

For 3-D DOM, this paper has proposed a data-driven method with spatial constraints in the latent space. Firstly, a self-supervised keypoint detection network is developed for deformable objects. The network operates without requiring ground truth labels or prior knowledge of deformed shapes while preserving spatial geometric information. The WFA-

TABLE I
t_s AND $\bar{e}$ ACROSS THREE TASKS (BEST RESULTS IN BOLD).

Task	Task 1		Task 2		Task 3	
Performance indicators	t_s [s]	$\bar{e}$ [m]	t_s [s]	$\bar{e}$ [m]	t_s [s]	$\bar{e}$ [m]
Proposed method	**27.3**	**0.02**	**54.3**	**0.02**	**36.7**	**0.03**
Broyden method	28.3	0.05	114.2	0.03	40.6	0.05
ENN method	38.0	0.04	83.8	0.04	41.3	0.05

enhanced encoder preserves critical features for keypoint localization and outputs keypoint probability scores for each point. Meanwhile, a geodesic distance loss function ensures semantic consistency of keypoints across 3-D frames regardless of shape deformation. Secondly, a Jacobian-based PPC method is proposed as a visual servoing controller. An adaptive fuzzy-based Jacobian estimator is developed to estimate the Jacobian matrix between keypoint velocities and robot joint velocities in real-time. Then, a PPC method is designed to guarantee keypoint errors converge within predefined spatial funnel functions. To prevent damage to deformable objects and provide safe interaction in DOM tasks, a PSG is designed, providing a feasible structure for soft robots. The effectiveness and practicality of the proposed method are verified in different types of deformable objects and different scenarios. Ablation experiments demonstrate that the proposed method achieves superior transient and steady-state performance.

However, the main limitations of the proposed method lie in its weak robustness to occlusion and its limited generalization ability for cross-object keypoint detection. That is, for objects with entirely new shapes or materials, a new point cloud dataset needs to be collected for offline training. In future work, we plan to combine other deep learning methods with the proposed method to address the occlusion and cross-object keypoint detection problems.

VII. ACKNOWLEDGMENT

The authors sincerely appreciate the insightful and constructive comments from the Associate Editor and reviewers, which have significantly enhanced the quality of this paper.

REFERENCES

[1] N. Lv, J. Liu, and Y. Jia, "Dynamic modeling and control of deformable linear objects for single-arm and dual-arm robot manipulations," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2341–2353, 2022.

[2] W. Li, P. W. Y. Chiu, and Z. Li, "An accelerated finite-time convergent neural network for visual servoing of a flexible surgical endoscope with physical and RCM constraints," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 12, pp. 5272–5284, 2020.

[3] D. Kruse, J. T. Wen, and R. J. Radke, "A sensor-based dual-arm tele-robotic system," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 1, pp. 4–18, 2015.

[4] O. Aghajanzadeh, M. Shetab-Bushehri, M. Aranda, J. A. Corrales Ramon, C. Cariou, R. Lenain, and Y. Mezouar, "3-D shape control of deformable linear objects for branch handling using an adaptive Lyapunov-based scheme," *Computers and Electronics in Agriculture*, vol. 232, p. 109931, 2025.

[5] Y. Li, Y. Chen, Y. Yang, and Y. Li, "Soft robotic grippers based on particle transmission," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 3, pp. 969–978, 2019.

[6] Z. Qiu, Q. Wu, Z. Liu, Y. Fang, and N. Sun, "A new three-dimensional deformation pneumatic soft actuator with mutually vertical PneuNets," *IEEE Transactions on Industrial Electronics*, vol. 71, no. 12, pp. 16003–16012, 2024.

[7] K. Tabata, H. Seki, T. Tsuji, and T. Hiramitsu, "Mass spring model for non-uniformed deformable linear object toward dexterous manipulation," *Artificial Life and Robotics*, vol. 28, no. 4, pp. 812–822, 2023.

[8] M. O. Fonkoua, F. Chaumette, and A. Krupa, "Deformation control of a 3D soft object using RGB-D visual servoing and FEM-based dynamic model," *IEEE Robotics and Automation Letters*, vol. 9, no. 8, pp. 6943–6950, 2024.

[9] F. Alambeigi, Z. Wang, R. Hegeman, Y.-H. Liu, and M. Armand, "A robust data-driven approach for online learning and manipulation of unmodeled 3-D heterogeneous compliant objects," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4140–4147, 2018.

[10] D. Navarro-Alarcon and 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*, vol. 34, no. 1, pp. 272–279, 2017.

[11] A. Caporali, K. Galassi, R. Zanella, and G. Palli, "GNN topology representation learning for deformable multi-linear objects dual-arm robotic manipulation," *IEEE Transactions on Automation Science and Engineering*, vol. 22, pp. 14738–14751, 2025.

[12] A. Caporali, K. Galassi, and G. Palli, "Dlo perceiver: Grounding large language model for deformable linear objects perception," *IEEE Robotics and Automation Letters*, vol. 9, no. 12, pp. 11385–11392, 2024.

[13] T. Wada, S. Hirai, S. Kawamura, and N. Kamiji, "Robust manipulation of deformable objects by a simple PID feedback," in *Proceedings 2001 IEEE International Conference on Robotics and Automation (ICRA)*, vol. 1, 2001, pp. 85–90.

[14] S. Tokumoto and S. Hirai, "Deformation control of rheological food dough using a forming process model," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, 2002, pp. 1457–1464.

[15] J. Armstrong Piepmeier, G. McMurray, and H. Lipkin, "A dynamic Jacobian estimation method for uncalibrated visual servoing," in *1999 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, 1999, pp. 944–949.

[16] B. Dahroug, B. Tamadazte, and N. Andreff, "Pca-based visual servoing using optical coherence tomography," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3430–3437, 2020.

[17] F. Janabi-Sharifi and M. Marey, "A kalman-filter-based method for pose estimation in visual servoing," *IEEE Transactions on Robotics*, vol. 26, no. 5, pp. 939–947, 2010.

[18] F. Cui, Q. Cui, and Y. Song, "A survey on learning-based approaches for modeling and classification of human–machine dialog systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 4, pp. 1418–1432, 2020.

[19] Z. Xu, Q. Li, W. Ma, M. Li, D. Morris, Z. Ren, and C. Zhao, "A geodesic distance regression-based semantic keypoints detection method for pig point clouds and body size measurement," *Computers and Electronics in Agriculture*, vol. 234, p. 110285, 2025.

[20] H. Jin, Y. Shen, J. Lou, K. Zhou, and Y. Zheng, "KeypointDETR: An end-to-end 3D keypoint detector," in *European Conference on Computer Vision (ECCV)*, 2024, pp. 374–390.

[21] C. Hou, Z. Xue, B. Zhou, J. Ke, L. Shao, and H. Xu, "Key-grid: Unsupervised 3D keypoints detection using grid heatmap features," in *Proceedings of the 38th International Conference on Neural Information Processing Systems (NIPS)*, 2024, pp. 49154–49179.

[22] M. Zohaib and A. Del Bue, "Sc3k: Self-supervised and coherent 3D keypoints estimation from rotated, noisy, and decimated point cloud data," in *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*, 2023, pp. 22509–22519.

[23] M. Zohaib, L. Cosmo, and A. Del Bue, "SelfGeo: Self-supervised and geodesic-consistent estimation of keypoints on deformable shapes," in *European Conference on Computer Vision (ECCV)*, 2024, pp. 71–88.

[24] X. Zhong, X. Zhong, and X. Peng, "Robust kalman filtering cooperated Elman neural network learning for vision-sensing-based robotic manipulation with global stability," *Sensors*, vol. 13, no. 10, pp. 13464–13486, 2013.

[25] N. Han, X. Ren, and D. Zheng, "Visual servoing control of robotics with a neural network estimator based on spectral adaptive law," *IEEE Transactions on Industrial Electronics*, vol. 70, no. 12, pp. 12586–12595, 2023.
[26] J. Li, X. Bian, H. Huang, T. Jiang, Y. Liu, and L. Wan, "Hybrid visual servoing control for underwater vehicle manipulator systems with multiple cameras," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 54, no. 3, pp. 1742–1754, 2024.
[27] J. Jiang, Y. Wang, Y. Jiang, Y. Feng, H. Zhong, and C. Yang, "Robust image-based adaptive fuzzy controller for guarantee field of view with uncertain dynamics," *IEEE Transactions on Fuzzy Systems*, vol. 32, no. 3, pp. 1564–1575, 2023.
[28] Q. Liu, J. Mao, L. Han, C. Zhang, and J. Yang, "Predictive observer-based dual-rate prescribed performance control for visual servoing of robot manipulators with view constraints," *IEEE Transactions on Cybernetics*, vol. 55, no. 5, pp. 2424–2436, 2025.
[29] Y. You, Y. Lou, C. Li, Z. Cheng, L. Li, L. Ma, C. Lu, and W. Wang, "Keypointnet: A large-scale 3D keypoint dataset aggregated from numerous human annotations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 13647–13656.
[30] N. Han, G. Gong, B. Zhang, Y. Xu, B. Yang, Y. Liu, and D. Navarro-Alarcon, "Prescribed performance control of deformable object manipulation in spatial latent space," *IEEE/ASME Transactions on Mechatronics*, pp. 1–11, 2026.
[31] X. Fang, Z. Liu, Y. Hao, H. Yang, J. Liu, Z. Xie, and L. Wen, "A soft actuator with tunable mechanical configurations for object grasping based on sensory feedback," in *2019 2nd IEEE International Conference on soft robotics (RoboSoft)*, 2019, pp. 25–30.
[32] D. Navarro-Alarcon, H. M. Yip, Z. Wang, Y.-H. Liu, F. Zhong, T. Zhang, and P. Li, "Automatic 3-D manipulation of soft objects by robotic arms with an adaptive deformation model," *IEEE Transactions on Robotics*, vol. 32, no. 2, pp. 429–441, 2016.
[33] P. Zhou, P. Zheng, J. Qi, C. Li, H.-Y. Lee, Y. Pan, C. Yang, D. Navarro-Alarcon, and J. Pan, "Bimanual deformable bag manipulation using a structure-of-interest based neural dynamics model," *IEEE/ASME Transactions on Mechatronics*, vol. 30, no. 5, pp. 3254–3265, 2025.
[34] B. Yang, B. Lu, W. Chen, F. Zhong, and Y.-H. Liu, "Model-free 3-D shape control of deformable objects using novel features based on modal analysis," *IEEE Transactions on Robotics*, vol. 39, no. 4, pp. 3134–3153, 2023.
[35] E. Park and J. Mills, "Static shape and vibration control of flexible payloads with applications to robotic assembly," *IEEE/ASME Transactions on Mechatronics*, vol. 10, no. 6, pp. 675–687, 2005.
[36] H. Wei, Y. Shan, Y. Zhao, L. Qi, and X. Zhao, "A soft robot with variable stiffness multidirectional grasping based on a folded plate mechanism and particle jamming," *IEEE Transactions on Robotics*, vol. 38, no. 6, pp. 3821–3831, 2022.
[37] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS)*, 2017, pp. 5105–5114.
[38] D. Liang, N. Sun, Y. Wu, G. Liu, and Y. Fang, "Fuzzy-sliding mode control for humanoid arm robots actuated by pneumatic artificial muscles with unidirectional inputs, saturations, and dead zones," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 5, pp. 3011–3021, 2022.
[39] L. Qiao, Q. Zhang, and G. Zhang, "Admissibility analysis and control synthesis for T–S fuzzy descriptor systems," *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 4, pp. 729–740, 2016.
[40] Y. Chen, L. Lan, X. Liu, G. Zeng, C. Shang, Z. Miao, H. Wang, Y. Wang, and Q. Shen, "Adaptive stiffness visual servoing for unmanned aerial manipulators with prescribed performance," *IEEE Transactions on Industrial Electronics*, vol. 71, no. 9, pp. 11028–11038, 2024.
[41] X.-J. Zeng and M. G. Singh, "Approximation theory of fuzzy systems-mimo case," *IEEE Transactions on Fuzzy Systems*, vol. 3, no. 2, pp. 219–235, 1995.
[42] F. Chaumette and S. Hutchinson, "Visual servo control. I. Basic approaches," *IEEE Robotics & Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.

Yuexuan Xu received the B.S. degree in automation and the M.S. degree in control science and engineering from Hebei University of Technology, Tianjin, China, in 2020 and 2023, respectively. He is currently working toward a dual Ph.D. degree in control science and engineering with the Institute of Robotics and Automatic Information Systems, Nankai University, Tianjin, China, and in mechanical engineering with The Hong Kong Polytechnic University, Kowloon, Hong Kong. His research interests include robot systems and control theory.

Ning Han received the B.Eng. and M.Eng. degrees in the School of Automation from Beijing Institute of Technology, Beijing, China, in 2021 and 2024, respectively. He is currently working toward the Ph.D. degree in the Department of Mechanical Engineering at The Hong Kong Polytechnic University. His primary research interests include deep learning and robot systems.

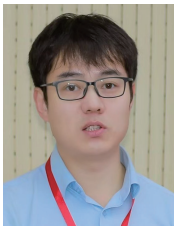
Jinrui Li is currently pursuing a dual B.Eng. degree in mechanical engineering with The Hong Kong Polytechnic University and in the School of Mechanical and Automotive Engineering with the South China University of Technology, Guangzhou, China. His research interests include additive manufacturing, robotic systems, and mechatronics.

Tong Yang (Member, IEEE) received the B.S. degree in automation and the Ph.D. degree in control science and engineering from Nankai University, Tianjin, China, in 2017 and 2022, respectively.

She is currently an Associate Professor with Nankai University and the Shenzhen Research Institute of Nankai University, Shenzhen, China. Her research interests include the nonlinear control of pneumatic artificial muscle-actuated robots and underactuated systems, including rotary cranes, offshore cranes, and tower cranes. She serves as an Associate Editor (editorial board member) for *Measurement and Control* and *Transactions of the Institute of Measurement and Control*.

Yun-Hui Liu (Fellow, IEEE) received his Ph.D. degree in Applied Mathematics and Information Physics from the University of Tokyo. After working at the Electrotechnical Laboratory of Japan as a Research Scientist, he joined The Chinese University of Hong Kong (CUHK) in 1995 and is currently Choh-Ming Li Professor of Mechanical and Automation Engineering and the Director of the T Stone Robotics Institute. He also serves as the Director/CEO of Hong Kong Centre for Logistics Robotics, sponsored by the InnoHK programme of the HKSAR government. He is an adjunct professor at the State Key Lab of Robotics Technology and System, Harbin Institute of Technology, China. He has published more than 500 papers in refereed journals and refereed conference proceedings and was listed in the Highly Cited Authors (Engineering) by Thomson Reuters in 2013. His research interests include visual servoing, logistics robotics, medical robotics, multi-fingered grasping, mobile robots, and machine intelligence.

Dr. Liu has received numerous research awards from international journals and international conferences in robotics and automation and government agencies. He was the Editor-in-Chief of Robotics and Biomimetics and served as an Associate Editor of the IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION and General Chair of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems.



Ning Sun (Senior Member, IEEE) received the B.S. degree in measurement & control technology and instruments (with honors) from Wuhan University, Wuhan, China, in 2009, and the Ph.D. degree in control theory and control engineering (with honors) from Nankai University, Tianjin, China, in 2014.

He is currently a Professor with Nankai University, Tianjin, China, the Shenzhen Research Institute of Nankai University, Shenzhen, China, and the Shenzhen Loop Area Institute, Shenzhen, China. His research interests include intelligent control for mechatronic/robotic systems with an emphasis on (industrial) applications.

Dr. Sun received the *Machines* 2021 Young Investigator Award, the prestigious Japan Society for the Promotion of Science (JSPS) Postdoctoral Fellowship for Research in Japan (Standard) in 2018, the *International Journal of Control, Automation, and Systems* Best Associate Editor Award in 2023, the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS Outstanding Associate Editor in 2024, several journal/conference best/outstanding paper awards, etc.

He serves as an Associate Editor/a Technical Editor for several journals, including IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, IEEE/ASME TRANSACTIONS ON MECHATRONICS, IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING.

David Navarro-Alarcon (Senior Member, IEEE) received the Ph.D. degree in mechanical and automation engineering from The Chinese University of Hong Kong (CUHK), in 2014. From 2014 to 2017, he worked as a Postdoctoral Fellow and then as a Research Assistant Professor at the T Stone Robotics Institute of CUHK. Since 2017, he has been with The Hong Kong Polytechnic University (PolyU), where he is currently an Associate Professor with the Department of Mechanical Engineering, and the Principal Investigator of the Robotics and Machine Intelligence Laboratory (ROMI-Lab). His current research interests include robotics and embodied intelligence. Dr. Navarro-Alarcon currently serves as an Associate Editor of the IEEE TRANSACTIONS ON ROBOTICS and a Technical Editor of the IEEE/ASME TRANSACTIONS ON MECHATRONICS.