Deformation Configuration Estimation for Soft Continuum Robot Utilizing Seq2Seq Learning

Hongye Zhang, Jingyu Zhang, Pingyu Xiang, Ke Qiu, Qin Fang, Yue Wang, *Member, IEEE*, Rong Xiong, *Senior Member, IEEE*, and Haojian Lu, *Member, IEEE*

***Abstract*—Inspired by biological tentacles, soft continuum robots exhibit the potential for navigating through narrow spaces and operating in complex environments, offering extensive application possibilities. However, owing to their inherent compliance, soft continuum robots may undergo unpredictable deformations in complex environments, leading to alterations in the whole-body configurations and diminished control precision. To address the problem, this letter employs sequence-to-sequence (Seq2Seq) learning to estimate the deformation of a tendon-driven continuum robot under multi-point contact. We also introduce a streamlined approach utilizing self-organizing mapping (SOM) to obtain ground truth data for training purposes and design dynamic loss functions for two-stage training, thereby facilitating neural network optimization in terms of both speed and precision. Furthermore, a soft continuum robot with two actively controlled degrees of freedom made of silicone is fabricated to verify the performance of the proposed method. The results show a shape estimation error of 2.94 mm (1.23% of the robot length).**

***Index Terms*—Continuum robot, contact kinematics, neural network, configuration estimation.**

I. Introduction

THE research significance of soft continuum robots lies in their potential to bring revolutionary changes to industries and medicine [1]. These robots, inspired by biological structures, possess extraordinary flexibility and adaptability [2]. In the industrial context, they can navigate confined spaces and perform tasks in complex environments, thus enhancing efficiency and productivity. In medicine, they enhance minimally invasive surgeries, providing precise manipulations and reducing patient trauma [3], [4], [5]. Soft continuum robots also offer safety benefits and assist in rehabilitation [6] and human-robot collaboration [7]. Overall, advancements in continuum robot research are expected to improve human work efficiency, accuracy, and quality.

Received 26 May 2025; accepted 18 October 2025. Date of publication 6 November 2025; date of current version 13 November 2025. This article was recommended for publication by Associate Editor E. Falotico and Editor C. Laschi upon evaluation of the reviewers' comments. This work was supported in part by the National Natural Science Foundation of China under Grant 62303407, and Grant T2293724, in part by the State Key Laboratory of Industrial Control Technology under Grant ICT2025A08, in part by the State Key Laboratory of Electrical Insulation and Power Equipment under Grant EIPE24212, and in part by Xiaomi Foundation. *(Hongye Zhang and Jingyu Zhang contributed equally to this work.) (Corresponding author: Haojian Lu.)*

Hongye Zhang, Pingyu Xiang, Ke Qiu, Yue Wang, Rong Xiong, and Haojian Lu are with the State Key Laboratory of Industrial Control and Technology, and Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou 310027, China (e-mail: luhaojian@zju.edu.cn).

Jingyu Zhang and Qin Fang are with the School of Information Science and Engineering, Harbin Institute of Technology (Weihai), Weihai 264200, China.

Digital Object Identifier 10.1109/LRA.2025.3629975

However, continuum robots often experience multiple unavoidable deformations due to their flexibility and environmental intricacies. Achieving high-precision manipulation of soft continuum robots remains challenging. As the robot undergoes unpredictable deformations and forces during contact with the environment, its kinematic parameters need adaptive adjustment for precise control.

Several modeling methods are currently used to describe the interaction between soft continuum robots and the environment.

The first approach is finite element analysis (FEA) [8], which utilizes discretization techniques to divide the robot and its environment into small elements. Material properties, boundary conditions, and contact forces are incorporated into the model, allowing for accurate simulation of deformations and forces during contact. Recently, SOFA, a simulation framework in soft robotics, has received extensive attention, enabling the implementation of various control strategies for soft robots in complex environments [9]. However, the substantial number of elements generated by FEA presents challenges in balancing computational speed and accuracy, often even leading to convergence issues. These limitations make it unsuitable for high-frequency online control applications.

The second modeling method is the continuum model [10], which employs mathematical equations to describe the behavior of soft robots. Different constitutive models, such as neo-Hookean [11] or Ogden models [12], are used to represent the material properties of the robot. These models consider the nonlinearity and time-varying behavior of soft materials, enabling accurate prediction of deformations during contact. Several works [13] based on the Cosserat rod theory have successfully demonstrated the capability to simultaneously fulfill both the stringent accuracy requirements for estimation and the essential demands for computational efficiency. In addition, an open-source toolbox known as SoRoSim [14] based on Cosserat rod theory enables effective static and dynamic analyses of continuum robotics. However, these methodologies typically rely on precise prior estimations of key parameters for continuum robots, including Young's modulus, moment of inertia, and shear modulus.

The third method is data-driven modeling [15]. Techniques such as neural network model have been explored to capture the complex interactions between robots and their environment. These methods utilize training data to learn the mapping between the robot's inputs and deformations, enabling real-time prediction and control. However, due to limitations of conventional sensors (e.g., size constraints, limited flexibility), obtaining high-quality training data for practical applications remains challenging. Current data-driven solutions can be broadly categorized into three types:

- Force/torque sensor–based methods, such as [16], which train networks based on mechanical models.
- Continuous shape–sensor–based methods, primarily relying on densely arranged FBG optical fibers [17], [18], liquid metal sensors [19], tactile sensors [20], or magnetic sensors [21] for robotic perception, with curve interpolation used for global fitting [22].
- Pure vision-based approaches, where current work typically employs end-to-end networks to learn the relationship between shape images and actuators [23], avoiding explicit modeling or parameterization.

It should be noted, however, that above methods either rely on fragile and costly continuous sensors (e.g., FBGs, liquid metals, or magnetic sensors), restrictive force-based priors, or unobstructed vision. These limitations hinder their applicability in unstructured environments, and critical issues such as robustness under occlusion, contact, and deformation remain open for future investigation.Therefore, this letter explores a marker-based configuration estimation algorithm for deformations of soft continuum robots in complex environments. The main contributions include:

- A deformation model for soft continuum robots is proposed. Drawing inspiration from machine translation, a Seq2Seq neural network is employed to estimate the configuration parameters of the robot under various deformation scenarios.
- A streamlined method for acquiring supervised data is proposed, requiring only a depth camera. Special loss functions and training strategies are devised to enhance the convergence and precision of the neural network.
- A prototype of a high length-diameter ratio continuum robot, measuring 240 mm in length and 4 mm in diameter, is fabricated to validate the effectiveness of the proposed methods under various configurations.

The article is organized as follows. In Section II, deformation model and a learning-based configuration estimation algorithm are proposed for the continuum robot in a complex environment. In Section III, experiments are carried out to test the performance of the proposed algorithm. Section IV briefly concludes and discusses the main achievements of this work.

II. Methods

The soft continuum robot studied in this letter mainly comprises light silicone material with a large length-to-diameter ratio, which could be potentially applied in minimally invasive surgery and industrial exploration.

When a soft continuum robot works in such a complex environment, as shown in Fig. 1(a), it will come into contact with surrounding obstacles and is passively bent to adapt to the environment due to its inherent flexibility. To achieve accurate

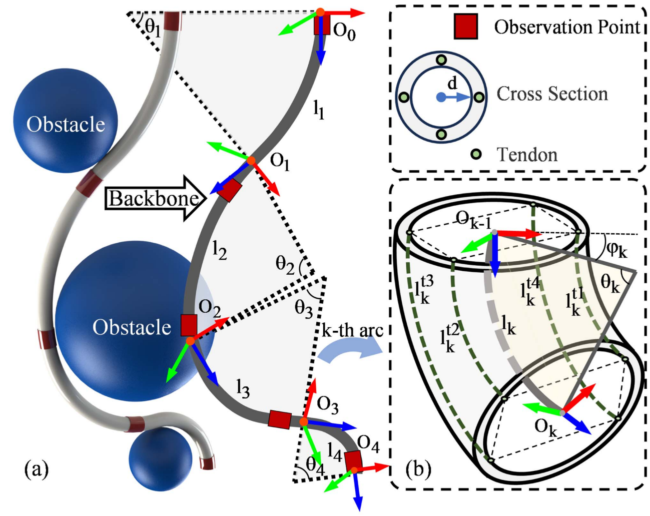


Fig. 1. Deformation model of soft continuum robot. (a) The robot makes multi-point contact with the environment and deforms according to the PCC assumption. Sparsely distributed red observation points are utilized to estimate the configuration parameters online for control. (b) the enlarged deformation illustration of the k-th arc with 4 tendons actuation.

control or motion prediction and further planning in the scenario, the configuration parameters of the robot need to be estimated and modeled. In this letter, the deformation of the robot is mainly modeled based on the PCC assumption for computational efficiency, which has been widely used in the continuum robotic community and has proven its effectiveness. As for the complex situation, the shape of every segment between two adjacent contact points could be approximately fitted well by several circular curves. Besides, in many continuum robot applications where few twisting moments are applied, the linear strain and torsional strain are negligible, while the primary attention is given to the bending deformation of the robot [24].

A. Deformation Modeling

Fig. 1 shows the schematic diagram of the continuum robot in the case of contact with the environment. The robot is driven by four tendons parallel to the centerline at a radius of d and separated by angles of $\pi/2$. Under the circumstances (the navy-blue spheres represent the complex environments, which could be human tissues or other obstacles), the robot is divided into a series of segments by the contact points, and each segment could be fitted by several circular curves. The reference frame $\{o_0 x_0 y_0 z_0\}$ is fixed on the catheter base while $\{o_{k-1} x_{k-1} y_{k-1} z_{k-1}\}$ represents the reference frame of the k-th constant curvature segment. Configuration parameters θ_k, φ_k, and l_k respectively represent the bending angle, bending plane angle and the length of the k-th constant curvature segment.

This letter intends to address the challenge of online deformation estimation of contact between continuum robots and the environment using learning-based methods, particularly in adequately representing the geometric parameters that characterize their shape when undergoing unknown deformations. Under the assumption of PCC, the deformations of continuum robots fitted

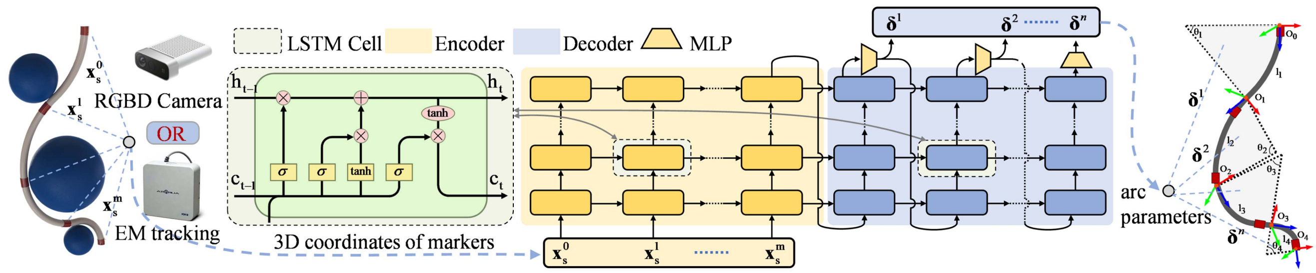


Fig. 2. The workflow of the deformation estimation and the Seq2Seq neural network to estimate the robots' deformation. The encoder takes the positions of $m+1$ sequentially arranged markers as input, while the decoder sequentially outputs parameters for n arc segments, forming a complete estimation of the robot's shape; the MLP layer reduces the dimension of the decoder's output to the desired shape.

with multiple arc segments can be analyzed as follows [25]. When a robot comes into contact with its surrounding environment, it undergoes bending deformation, segmented into n arc sections as illustrated in Fig. 1(a).

As shown in Fig. 1(b), the homogeneous coordinate transformations between adjacent coordinate systems $\{o_{k-1}x_{k-1}y_{k-1}z_{k-1}\}$ and $\{o_k x_k y_k z_k\}$ consist of a translation along the z-axis and the rotation around the x and y axes according to the respective angular velocity components. Hence, we can represent it in exponential form as follows,

$$ {}^{k}_{k-1}\mathrm{T} = e^{(\hat{\xi} l_k)}, \tag{1}$$

where $\xi = [0, 0, 1, -\kappa \sin\varphi_k, \kappa\cos\varphi_k, 0]^{\mathrm{T}}$, $\kappa = \frac{\theta_k}{l_k}$, represent the twist coordinate associated with the rotation and translation. $\hat{}$ maps $\mathbb{R}^6$ in $\mathfrak{se}(3)$, the Lie algebra of SE(3), to $\mathbb{R}^{4\times4}$ for the exponential mapping to SE(3).

Therefore, the homogeneous transformation matrix between the reference frame $\{o_0x_0y_0z_0\}$ and $\{o_nx_ny_nz_n\}$ is as follows [26],

$$ {}^{0}_{n}\mathrm{T} = \prod_{k=1}^{n} {}^{k-1}_{k}\mathrm{T}. \tag{2}$$

B. Learning-Based Configuration Estimation

Based on the PCC assumption, when a robot comes into contact with a complex environment, it can be described as spatial arcs smoothly connected in a series. In order to estimate the parameters that sufficiently define all the arcs using a finite number of distributed position sensors, such as electromagnetic (EM) tracking and camera markers, a learning-based configuration estimation framework is proposed. Visual markers can be utilized for robotic shape estimation under intraoperative X-ray imaging, while the EM tracking system boasts a wide range of application scenarios and can be utilized in various fields such as surgical procedures and pipeline inspection.

Multi-Layer Perceptron (MLP) processes inputs simultaneously and disregards the physical connections between predicted arcs, often causing training to fail. To address this, we need a network capable of handling sequential data—one that explicitly models the relationship between input markers and output arc parameters. We expect that the input consists of information from sensors arranged equidistantly along the extension direction of the robot, while the output, on the other hand, comprises arc parameters, with each subsequent arc parameter unfolding within the local coordinate system defined by the preceding arc.

Therefore, a sequence-to-sequence network [27] is selected for the arc parameter estimation task. A typical Seq2Seq network comprises an encoder and a decoder. The encoder converts the input sequence into a fixed-dimensional feature vector, which the decoder uses to generate the output sequence. Notably, during decoding, each step's input is the previous step's output, starting with the encoder's feature vector.

Fig. 2 illustrates the fundamental workflow of the proposed deformation estimation algorithm and the architecture of the employed neural network. The EM tracking system or depth camera captures the positions of markers on the robot's backbone, which are then input into the neural network to obtain the arc parameters representing the robot's shape. The orange section represents the encoder, which takes the positional information $\{x_s^0, x_s^1, \ldots, x_s^m\}$ of $m+1$ landmarks distributed along the robot body as input. The blue section represents the decoder, which receives the context vector from the encoder and recursively generates the output at each time step. Each time step estimates an arc parameter, $\boldsymbol{\delta}^i$. To transform the output dimension to the dimension required to fully define a spatial arc segment, a fully connected layer is added after each time step of the decoder.

We have employed a multi-layer LSTM as the backbone of the Seq2Seq encoder-decoder. Each individual LSTM cell, as depicted by the dashed box in Fig. 2, addresses the issues of gradient vanishing and exploding that commonly occur in long sequence problems through the selective memory of the forget gate and output gate [28]. As a result, it generally outperforms traditional recurrent neural networks. The variables h_t and c_t represent the hidden state and cell state of the LSTM cell, respectively, which are responsible for capturing information from distant to recent with a different emphasis. Consequently, during the forward propagation, h_t and c_t are passed along the direction of the arrows.

The output of the neural network consists of n vectors, each with a dimension of 4, denoted as $\delta_i = [L_i, \Theta_i, \sin\varphi_i, \cos\varphi_i]$, representing the parameters of the i-th arc. However, it is crucial to highlight that the arc length and bending plane are not directly derived from the neural network's output and require special consideration. This is due to the following reasons.

1) Arc Length l_i: A necessary condition for the estimated arc length to be reasonable is that the predicted length should be equal to the actual length of the robot, L. A normalization operation is introduced at the output to ensure that the estimated length always satisfies this condition. Therefore, the length of each segment is calculated as follows:

$$l_i = \frac{L_i^2 L}{\sum_{i=1}^{n} L_i^2} \tag{3}$$

2) Bending Angle θ: The bending angle θ of the robot ranges from 0 to 2π, corresponding to fully extended and bent states. Thus, to estimate the value of θ with a monotonically bounded output, a $Sigmoid$ activation function is applied after the corresponding output neuron. It is important to note that the derivative of the $Sigmoid$ function approaches zero when the variable is far from the origin. To prevent the variable from falling into this range and causing slow convergence of gradient descent, batch normalization is performed before the output layer. Therefore, the bending angle is calculated as follows.

$$\theta_i = \frac{2\pi}{1 + e^{-\Theta_i}} \tag{4}$$

3) Bending Plane φ: The bending plane also ranges from 0 to 2π. However, the robot can rotate continuously around the z-axis, with a smooth transition at the boundary. This is a challenging situation for artificial neural networks to handle [29]. Treating it similarly to the bending angle θ would result in a discontinuity when φ changes from 2π to 0. To address this, two separate neurons are used to output the sine and cosine values of the bending plane, circumventing the discontinuity in such circumstances. Therefore, the bending plane can be represented as follows:

$$\varphi_i = \arctan 2 \left(\frac{\sin \varphi_i}{\cos \varphi_i} \right) \tag{5}$$

In this study, we employed an RGB-D camera to capture the pose of $m + 1$ markers that are evenly distributed on the robot.

C. Data Acquisition

To ensure the effectiveness of a learning-based approach, obtaining reliable training data is essential. For neural network training, we use the positions of the markers on the continuum robot as inputs to the network, and select sequentially arranged scattered points along the robot's extension direction to represent its overall shape. Specifically, we perform binarization processing on the RGB-D images to separate the background from the foreground of the continuum robot, thereby obtaining the coordinates of all points belonging to the robot. While the marker points differ in color from other parts, their coordinates are obtained by extracting the distinct-colored regions directly from the original RGB-D camera images.

However, the point cloud captured by the RGB-D camera contains only pixel-level positional labels and does not ensure that the points are distributed in an ordered manner along the continuum robot. Theoretically, connecting these points in the correct order should yield an accurate representation of the robot's configuration without any topological errors. To accomplish this task, we decided to utilize the self-organizing mapping (SOM), as proposed in [30], with slight modifications to be more suitable for our specific requirements.

Let the set $S = \{\mathrm{P}_i = (x_i, y_i, z_i) \in \mathbb{R}^3, i = 1, \ldots, N\}$ be the 3D point cloud extracted from the robot skeleton in the space. Let the set M be the set of collected points, which are referred to as reference vectors (RVs) in the subsequent description, $M = \{Q_i = (x_i, y_i, z_i) \in \mathbb{R}^3, i = 1, \ldots, k\}$. Generally, the initial k RVs are randomly selected from set S, and their positions will be updated during later iterations. $Q_c(P_i)$ and $Q_{c'}(P_i)$ represent the nearest and second nearest reference vectors to point P_i, respectively. For each Q_i in M, calculate the Euclidean distance to all points in set S, forming the matrix $\mathbf{A}$:

$$\mathbf{A}_{mn} = \sum_{i}^{N} \begin{bmatrix} 1, if Q_m = Q_c(P_i) and Q_n = Q_{c'}(P_i) \\ 0, else \end{bmatrix}, \quad m \leq k, n \leq k \tag{6}$$

The element in the m-th row and n-th column of matrix M represents the number of robot skeleton point clouds that are nearest to Q_m and second nearest to Q_n among all reference vectors $Q_i \in M$. If $\mathbf{A}_{mn} \neq 0$, it indicates that there are several points between Q_m and Q_n, while if $\mathbf{A}_{mn} = 0$, there are no points between Q_m and Q_n, inferring that Q_m and Q_n should not be adjacent. Therefore, matrix $\mathbf{A}$ provides a sequential order for RVs. If the reference vectors are arranged in order along the growth direction of the point cloud, $\mathbf{A}$ should be a sparse matrix with a main diagonal of 0 and a bandwidth of 1. For RVs with improper arrangements, the matrix $\mathbf{A}$ is a sparse matrix with large bandwidth. To rearrange matrix $\mathbf{A}$, we use sparse reverse Cuthill-McKee ordering [31], which redistributes the non-zero elements of the matrix closer to the main diagonal, resulting in a bandwidth of 1.

After each sorting, it is necessary to update the positions of the RVs to fit the robot's skeleton. The update strategy is similar to the k-means clustering algorithm [32]. In the t-th iteration, find all point cloud sets $N_i^{(t)}$ that are nearest to the reference vector $Q_i^{(t)}$, and update the position of $Q_i^{(t)}$ in the next iteration as the centroid of set $N_i^{(t)}$:

$$Q_i^{(t+1)} = \frac{1}{\left| N_i^{(t)} \right|} \sum_{(x_j, y_j, z_j) \in N_i^{(t)}} (x_j, y_j, z_j) \tag{7}$$

Repeat this process for multiple iterations until the reference vectors are correctly sorted and accurately represent the robot's shape. This method can also be used not only to arrange the order of RVs but also to sort the markers distributed on the robot's body in just one iteration.

D. Loss Functions

Training without absolute ground truth is a challenging task when no accurate prior knowledge of the arc parameters is provided. Nevertheless, we can still define custom optimization objectives to improve the capability of our neural network with the supervision of the ordered RVs set M.

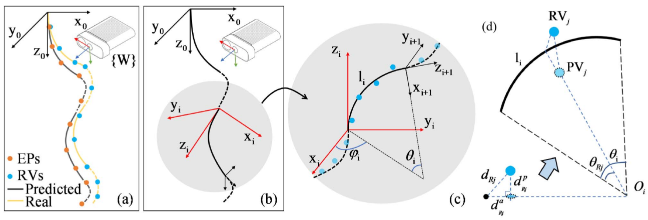


Fig. 3. Illustration of calculating two types of losses. (a) The non-uniformly spaced RVs were obtained from the SOM, and the uniformly spaced EPs were manually selected. (b) The local coordinate system of the estimated i-th arc. (c) The aligned local coordinate system, where the blue dots represent the RVs. (d) Calculate the distance between the j-th RV and the i-th arc by projecting the point RV_j onto the plane of the arc and obtain the projection PV_j and the angle θ_{Ri}. $\theta_{Ri} < \theta_i$ indicates that the point RV_j belongs to this arc.

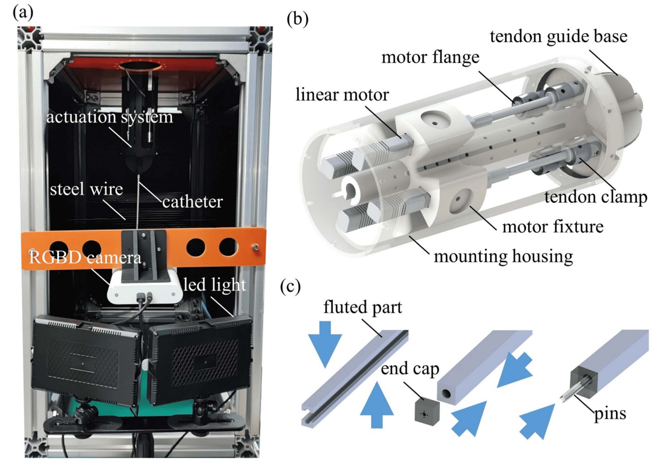


Fig. 4. Experimental Setup. (a) The composition of the experimental system. The system consists of a man-made experimental environment, an RGB-D camera to capture the pose of the catheter and a soft continuum robot. (b) The actuation system of the soft continuum robot. The actuation system consists of four linear motors that pull four tendons for the robot control. (c) The fabrication of the catheter. The catheter is made of PDMS (Polydimethylsiloxane) material by the silicone molding process.

Inspired by the dynamic loss functions of neural networks, which set up appropriate learning objects for different situations of the networks [33], we adopt a staged training approach with dynamic loss functions. In the first stage, we perform coarse training to ensure that the predicted curve aligns with the actual robot curve to a basic extent. In the second stage, we fine-tune the network's fitting capability.

1) Coarse Training Stage: Select equidistant points (EPs) along the predicted robot curve to form a set $E = \{S_i = (x_i, y_i, z_i) \in \mathbb{R}^3, i = 1\ldots k\}$. The mean squared error (MSE) is calculated between every element S_i in set E and sorted Q_i in set S.

In addition, to prevent the problem from falling into a local optimum, penalty terms should be added to the loss function. To ensure all arcs serve their purpose requires avoiding overly short arcs, keeping length variations moderate, and preventing high curvature in segments. The overall optimization objective is as follows:

$$f(w) = \sum_{i=1}^{k} \|S_i - Q_i\| + \lambda_1 \sqrt{\frac{\sum_{j=1}^{n} (l_j - \bar{l})}{n-1}} + \lambda_2 \sum_{j=1}^{n} \frac{\theta_j}{l_j} \tag{8}$$

where w are the parameters to be optimized in neural networks, and λ_1, λ_2 are the weights of penalty term regarding the variance of arc length and curvature, respectively.

2) Fine-Tune Stage: The RVs obtained through SOM are not uniformly distributed or equally spaced, resulting in a misalignment between S_i and Q_i during the coarse training stage, as illustrated in Fig. 3(a). Although the training process described earlier has resulted in an initial fitting of the robot, additional fine-tuning is necessary to rectify any remaining inaccuracies or shortcomings.

As depicted in Fig. 3(b) and (c), the RVs are projected onto the local coordinate system of the corresponding arc. By comparing the angles of θ_i and θ_{Rj} in Fig. 3(d), we can determine which arc each RV belongs to. Fig. 3(c) illustrates that there are 4 points falling within the i-th arc.

The distance between the j-th RV and its corresponding arc i, denoted as d_{Rj}, is calculated as $d_{Rj} = \sqrt{{d^a_{Rj}}^2 + {d^p_{Rj}}^2}$. This distance is defined as the distance between the RVs and the estimated robot, as illustrated in Fig. 3(d). Here, d^p_{Rj} represents the distance from the j-th RV to the bending plane, and $d^a_{Rj} = |\frac{l_i}{\theta_i} - \mathrm{PV}_j\mathrm{O}_i|$ is the shortest distance from the projection point PV_j to the corresponding arc.

Therefore, the overall optimization objective is formulated as follows:

$$f(w) = \sum_{j=1}^{k} d_{Rj} \tag{9}$$

III. Results

A. Experimental Setup

To verify the performance of the proposed configuration estimation algorithm, an experimental system was established, as shown in Fig. 4(a). This experimental system employs an aluminum profile framework, featuring multiple sets of wire ropes affixed to the backplanes on both sides to simulate varied environments. It is equipped with an RGB-D camera to capture the pose of red markers attached to the continuum catheter and dual LED lights for illuminating the experimental setup.

A soft continuum robot is designed and fabricated for the experimental verification, including a silicone catheter and its actuation system (Fig. 4 b). The actuation system consists of four linear motors installed on the motor fixture, which are used to pull four tendons routed by the tendon guide base. One end of the tendon is connected to the catheter tip, and the other end is connected to the linear motor through the tendon clamp. The motor fixture is installed on the mounting housing that is mounted on the aluminum profile frame. During experimentation, we actuate the linear motor to pull the tendon, causing the continuum robot to bend and advance through steel cables at various positions and quantities. In addition, the positions of the steel cables can be modified by adjusting their fixed mounting

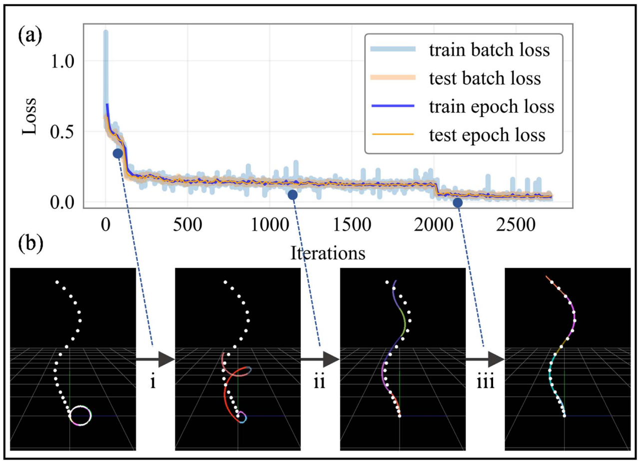


Fig. 5. Visualization of neural network training process. (a) Error variation during training. (b) The estimated configuration of different stages in the training process. As the number of iterations increases, the estimated configuration gradually approximates that of a real robot.

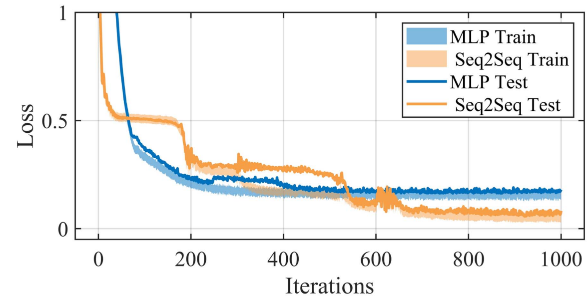
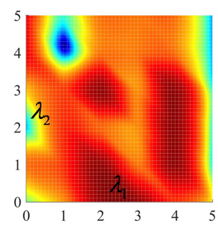


Fig. 6. Training loss and validation loss of the MLP and Seq2Seq networks under the same reduced training set. The MLP network failed to meet the threshold requirements during the coarse training phase, whereas the Seq2Seq architecture demonstrated superior convergence characteristics.



(a)

		Error (%)	S. D. (%)
s1&s2	**CR**	**1.22**	0.13
	DR	4.16	1.23
s1 only	CR	3.75	1.73
	DR	7.42	1.75
s2 only	CR	4.31	**0.09**
	DR	2.83	7.50

(b)

Fig. 7. Comparative results of the training process. (a) the grid search of hyperparameters λ_1 and λ_2. (b) The performance of the network under different strategies and variable representations.

points on the supporting frame. Utilizing the friction between the robot and the cables, we secured the robot in place, enabling its body to assume diverse configurations.

The catheter is made of PDMS (Polydimethylsiloxane) using a silicone molding process. The aluminum mold for the catheter (Fig. 4(c)) is made with computer numerical control (CNC) technology. The mold is formed by joining two fluted parts to create a cylindrical groove, followed by closing an end cover with holes for inserting fine pins. These pins create tendon channels and an intermediate channel for holding a micro camera or other tools.

The coordinate values of the input points are normalized and fed into the Seq2Seq neural network which generates a set of n spatial arc parameters. In this experiment, $m = 4$ and $n = 6$. The network parameters were subsequently optimized by different losses based on distinct training stages, as mentioned in Section II(D).

B. Training Process of Neural Network

Before training, the number of segments for the continuum robot was predetermined through comprehensive environmental assessment, resulting in 6 segments. To improve the accuracy of the configuration estimation algorithm, a variety of circumstances are constructed to collect the data according to the method described in Section II(C). The collected data is randomly divided into training and testing sets in a 7:3 ratio. The neural networks were trained in PyTorch using the Adam optimizer with a learning rate of 0.0005. Training was performed on a high-performance workstation with an Intel Core i7-13600KF CPU and an NVIDIA GeForce RTX 4070 GPU. A dataset of 200 samples was randomly selected, and the network was trained for about 3,000 epochs until convergence. Fig. 5(b) illustrates the roles of loss functions in training and network performance. Initially, random weight initialization caused significant errors, resulting in n arcs that were curved at the base. The first loss function optimized these curves (Process i). After a plateau, losses dropped rapidly, and the output curves approached the actual data points (Process ii). Once the mean of several losses fell below a threshold, the loss function was switched to the fine-tune stage, minimizing distances between RVs and the output curve (Process iii), enhancing shape estimation. Training was stopped at convergence, with no significant testing error increase, indicating no overfitting, as shown in Fig. 5(a).

On another reduced training set, we trained both the MLP network and our Seq2Seq network using identical parameters. As shown in Fig. 6, the training loss and the validation loss both demonstrate that the Seq2Seq network achieves a faster decline in loss, while the MLP network exhibits a slower decline and converges to a local minimum.

C. Comparative Analyses

Several comparative analyses are conducted to prove the effectiveness of various adjustments outlined in Section II.

Firstly, we determined two hyperparameters, λ_1 and λ_2, of the loss function ((8)). Through grid search within a certain range during the training process on the same dataset, the optimal hyperparameters are identified as $\lambda_1 = 1, \lambda_2 = 4.2$. As illustrated in Fig. 7(a), regions with cooler tones represent smaller losses, while warmer regions represent larger losses. This indirectly corroborates that incorporating appropriate penalty terms can enhance performance for this specific learning task. Secondly, we compared the neural network's performance under different training strategies: step 1 only, step 2 only, and using both with

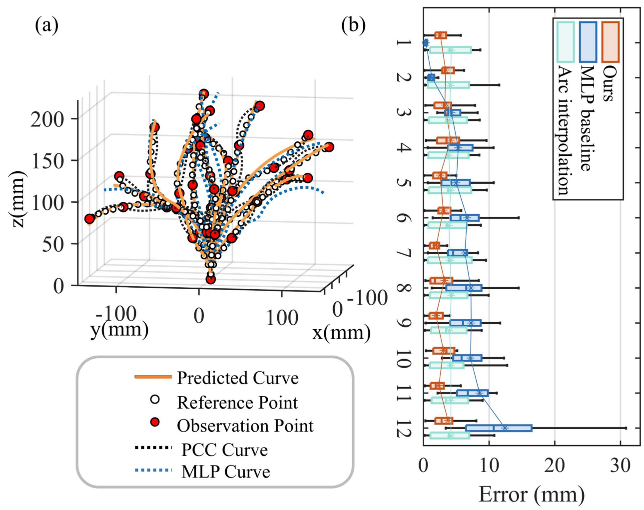


Fig. 8. Verification and comparable results of the configuration estimation method. (a) The fitting plots of the proposed method, MLP baseline and arc interpolation for the reference points. (b) Error bar plot of proposed method, MLP baseline and arc interpolation.

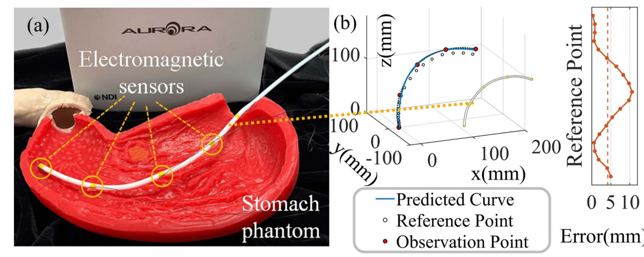


Fig. 9. Setup and results of application with EM Tracker. (a) Experimental setup, including a stomach model, a continuum robot, and an EM tracking system. (b) Comparison between the output of the network and the ground truth obtained from EM trackers.

dynamic loss adjustment, as shown in Fig. 7(b). The former improvement in the neural network training process is evident. When trained with both steps, the network converges faster to the least error, 1.22% of the robot length. The latter improvement in network performance is reflected in the standard deviation (S. D.) of errors during training and testing. DR poses issues with larger S. D. at the boundary between 0 and 2π as CR does. Instead, they can only adjust neuron outputs in the opposite direction towards another limit, which can lead to output instability.

D. Verification Results of Configuration Estimation Algorithm

In order to test the proposed configuration estimation algorithm, several different circumstances are constructed to verify the accuracy, as shown in Fig. 8. These scenarios are generated through random interactions, following the same generation methodology as the dataset. The actual points of the catheter (hollow circle as the RVs and solid red circle as the input) and predicted configuration curve (orange solid curve) are also plotted to show the accuracy. It can be seen that the dots are basically distributed on the orange curve, indicating the high precision of the configuration estimation.

To further evaluate the accuracy, the error metric is computed as the mean positional deviation between RVs and the reconstructed arcs based on the network's output parameters. This measurement approach aligns with the loss function implementation used in the fine-tuning phase. As shown and calculated in Fig. 8(b), it can be seen that the errors of these circumstances are mostly distributed within a range of about 5 mm. The mean error of our method is 2.94 mm, about 1.23% of the length of the robot (240 mm). The error distribution is relatively uniform, with a variance of 2.16 mm, indicating that our method produces stable errors, which facilitates further investigation into error characteristics.

To evaluate the effectiveness of our method, we compared it with a PCC-based arc interpolation and an MLP baseline. As shown in Fig. 8, PCC fits five input points with four arcs constructed from adjacent points. Attempting to compute additional arcs from the same reference points would introduce multiple possible solutions that are difficult to resolve and may not guarantee optimality. While reasonably accurate for simple deformations, PCC performs poorly on complex configurations, with a mean error of 4.10 mm and a standard deviation of 3.17 mm, both higher than our method's 2.94 mm and 2.16 mm, respectively. PCC curves must pass through reference points, resulting in small errors at these points but large errors elsewhere, leading to inaccurate overall shape estimation. The MLP baseline trained to estimate PCC parameters achieved a mean error of 6.00 mm with a standard deviation of 5.16 mm. This likely reflects the limited representational capacity of MLPs, resulting in larger errors and even underperforming simple PCC arc interpolation.

E. Application With EM Tracker

To validate the effectiveness of our method in other scenarios, we constructed an EM tracker system to conduct experiments within a gastric model.

As shown in Fig. 9(a), the setup includes a human stomach model and a continuum robot equipped with several EM trackers. Fig. 9(b) presents the experimental results. It can be observed that the ground truth points of the continuum robot are distributed around the blue curve, with the error range consistent with the results in Section III.D. The results demonstrate that our method possesses generalization capabilities for transfer to other scenarios such as surgical scenarios under non-line-of-sight conditions using EM sensors.

IV. Conclusion

This letter utilizes Seq2Seq learning to obtain a deformation estimation for tendon-driven continuum robots deformed in complex environments. Additionally, a streamlined method based on SOM is proposed to acquire supervised data, and a training approach with dynamic loss functions is employed to address the training issues of the neural network. Moreover, an experimental system was built to test the performance of the proposed configuration estimation algorithm. The results show a shape estimation error of 2.94 mm (1.23% of the robot length). And the parameterized shape prediction obtained by our method can be readily integrated into state-of-the-art control frameworks, such as model predictive control.

Our approach only requires the capture of markers along the continuum robot's backbone, significantly expanding its potential application scenarios. The proposed method demonstrates theoretical applicability to systems with a continuous backbone structure, including but not limited to flexible endoscopes, guidewires, and catheters. Nevertheless, the integration of additional sensors to enhance precision remains a crucial research focus for future investigations.

It should be emphasized that the proposed algorithm is fundamentally based on the PCC assumption, which inherently carries certain limitations [34]. Under significant environmental interactions and contact forces, the PCC assumption rapidly deteriorates. For instance, in recent tests with a soft manipulator performing complex end-effector tasks, one module exhibited severe deviation from the constant curvature assumption [35]. While our method demonstrates improved performance compared to direct PCC computation, these limitations cannot be entirely eliminated. The selection of markers significantly impacts algorithm precision. Generally, in practical implementations, the selection of markers' quantity and position requires careful trade-offs to balance environmental adaptability with training complexity. Additionally, our current approach necessitates preliminary environmental assessments to determine the number of PCC segments.

While the design of our network allows for variable arc segmentation through dynamic prediction length methods, this capability requires larger datasets and remains a key focus for future research. Furthermore, more research should consider a wider range of complex environments, such as industrial and medical scenarios, which require precise navigation and manipulation.

REFERENCES

[1] X. Gu and H. Ren, "A survey of transoral robotic mechanisms: Distal dexterity, variable stiffness, and triangulation," *Cyborg Bionic Syst.*, vol. 4, 2023, Art. no. 0007.

[2] L. Liu et al., "AI search, physician removal: Bronchoscopy robot bridges collaboration in foreign body aspiration," *Sci. Robot.*, vol. 10, no. 104, 2025, Art. no. eadt5338.

[3] J. Duan, K. Zhang, K. Qian, J. Hao, Z. Zhang, and C. Shi, "An operating stiffness controller for the medical continuum robot based on impedance control," *Cyborg Bionic Syst.*, vol. 5, 2024, Art. no. 0110.

[4] X. Duan et al., "A novel robotic bronchoscope system for navigation and biopsy of pulmonary lesions," *Cyborg Bionic Syst.*, vol. 4, 2023, Art. no. 0013.

[5] B. Wang et al., "Magnetically driven biohybrid blood hydrogel fibres for personalized intracranial tumour therapy under fluoroscopic tracking," *Nature Biomed. Eng.*, vol. 9, pp. 1471–1485, 2025.

[6] X. Du et al., "Active exploration and reconstruction of vascular networks using microrobot swarms," *Nature Mach. Intell.*, vol. 7, pp. 553–564, 2025.

[7] J. Zhang et al., "AI co-pilot bronchoscope robot," *Nature Commun.*, vol. 15, no. 1, 2024, Art. no. 241.

[8] S. Grazioso, G. Di Gironimo, and B. Siciliano, "A geometrically exact model for soft continuum robots: The finite element deformation space formulation," *Soft Robot.*, vol. 6, no. 6, pp. 790–811, 2019.

[9] R. Dreyfus, Q. Boehler, and B. J. Nelson, "A simulation framework for magnetic continuum robots," *IEEE Robot. Automat. Lett.*, vol. 7, no. 3, pp. 8370–8376, Jul. 2022.

[10] J. Till, V. Aloi, and C. Rucker, "Real-time dynamics of soft and continuum robots based on cosserat rod models," *Int. J. Robot. Res.*, vol. 38, no. 6, pp. 723–746, 2019.

[11] Y. Kim, G. A. Parada, S. Liu, and X. Zhao, "Ferromagnetic soft continuum robots," *Sci. Robot.*, vol. 4, no. 33, 2019, Art. no. eaax7329.

[12] Q. Zhao, J. Lai, K. Huang, X. Hu, and H. K. Chu, "Shape estimation and control of a soft continuum robot under external payloads," *IEEE/ASME Trans. Mechatron.*, vol. 27, no. 5, pp. 2511–2522, Oct. 2022.

[13] S. Lilge, T. D. Barfoot, and J. Burgner-Kahrs, "State estimation for continuum multi-robot systems on SE(3)," *IEEE Trans. Robot.*, vol. 41, pp. 905–925, 2025.

[14] A. T. Mathew, I. B. Hmida, C. Armanini, F. Boyer, and F. Renda, "SoRoSim: A MATLAB toolbox for hybrid rigid–soft robots based on the geometric variable-strain approach," *IEEE Robot. Automat. Mag.*, vol. 30, no. 3, pp. 106–122, Sep. 2023.

[15] P. Xiang et al., "Learning the inverse kinematics of magnetic continuum robot for teleoperated navigation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2024, pp. 13070–13075.

[16] S. Shekari and S. A. A. Moosavian, "Shape estimation of continuum arms based on inverse kinematics data-driven modeling," in *Proc. 12th RSI Int. Conf. Robot. Mechatron.*, 2024, pp. 356–361.

[17] J. Gan, A. Yang, Q. Guo, and Z. Yang, "Flexible optical fiber sensing: Materials, methodologies, and applications," *Adv. Devices Instrum.*, vol. 5, 2024, Art. no. 0046.

[18] S. Sefati, C. Gao, I. Iordachita, R. H. Taylor, and M. Armand, "Data-driven shape sensing of a surgical continuum manipulator using an uncalibrated fiber bragg grating sensor," *IEEE Sensors J.*, vol. 21, no. 3, pp. 3066–3076, Feb. 2021.

[19] Z. Xie et al., "Octopus-inspired sensorized soft arm for environmental interaction," *Sci. Robot.*, vol. 8, no. 84, 2023, Art. no. eadh7852.

[20] L. Scimeca, J. Hughes, P. Maiolino, and F. Iida, "Model-free soft-structure reconstruction for proprioception using tactile arrays," *IEEE Robot. Automat. Lett.*, vol. 4, no. 3, pp. 2479–2484, Jul. 2019.

[21] H. Guo et al., "Continuum robot shape estimation using permanent magnets and magnetic sensors," *Sensors Actuators A: Phys.*, vol. 285, pp. 519–530, 2019.

[22] L. Hou, S. Tam, X. Zhao, and B. Tao, "Shape reconstruction method for continuum robot using FBG sensors," in *Proc. Intell. Robot. Applications: 15th Int. Conf.*,, Harbin, China, Springer, 2022, pp. 388–395.

[23] E. Almanzor, F. Ye, J. Shi, T. G. Thuruthel, H. A. Wurdemann, and F. Iida, "Static shape control of soft continuum robots using deep visual inverse kinematic models," *IEEE Trans. Robot.*, vol. 39, no. 4, pp. 2973–2988, Aug. 2023.

[24] J. Yang, H. Peng, W. Zhou, J. Zhang, and Z. Wu, "A modular approach for dynamic modeling of multisegment continuum robots," *Mechanism Mach. Theory*, vol. 165, 2021, Art. no. 104429.

[25] J. Burgner-Kahrs, D. C. Rucker, and H. Choset, "Continuum robots for medical applications: A survey," *IEEE Trans. Robot.*, vol. 31, no. 6, pp. 1261–1280, Dec. 2015.

[26] R. J. Webster III and B. A. Jones, "Design and kinematic modeling of constant curvature continuum robots: A review," *Int. J. Robot. Res.*, vol. 29, no. 13, pp. 1661–1683, 2010.

[27] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 3104–3112.

[28] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: LSM cells and network architectures," *Neural Computation*, vol. 31, no. 7, pp. 1235–1270, 2019.

[29] X. T. Ha et al., "Shape sensing of flexible robots based on deep learning," *IEEE Trans. Robot.*, vol. 39, no. 2, pp. 1580–1593, Apr. 2023.

[30] G. S. Kumar, P. K. Kalra, and S. G. Dhande, "Curve and surface reconstruction from points: An approach based on self-organizing maps," *Appl. Soft Comput.*, vol. 5, no. 1, pp. 55–66, 2004.

[31] E. Cuthill and J. McKee, "Reducing the bandwidth of sparse symmetric matrices," in *Proc. 24th Nat. Conf.*, 1969, pp. 157–172.

[32] M. Ahmed, R. Seraj, and S. M. S. Islam, "The K-means algorithm: A comprehensive survey and performance evaluation," *Electronics*, vol. 9, no. 8, 2020, Art. no. 1295.

[33] L. Wu et al., "Learning to teach with dynamic loss functions," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 6466–6477.

[34] C. Armanini, F. Boyer, A. T. Mathew, C. Duriez, and F. Renda, "Soft robots modeling: A structured overview," *IEEE Trans. Robot.*, vol. 39, no. 3, pp. 1728–1748, Jun. 2023.

[35] C. Alessi, D. Bianchi, G. Stano, M. Cianchetti, and E. Falotico, "Pushing with soft robotic arms via deep reinforcement learning," *Adv. Intell. Syst.*, vol. 6, no. 8, 2024, Art. no. 2300899.