

Geometric-Aware Attention Networks with Adaptive Constraint Adjustment for Static Modeling of the Tendon-driven Continuum Robot

Xin Xu¹, Yang Song¹, Bowen Su¹, Ke Wu², Di Wu³, Wendi Liang¹, Gang Zheng⁴, Henry K. Chu¹

Abstract—Tendon-driven continuum robots (TDCRs) with hyper-redundant DOFs are widely developed for sophisticated tasks such as minimally invasive surgery. However, the static modeling of the TDCRs is challenging due to the nonlinear characteristics of the TDCRs' structures and material properties. To tackle the above-mentioned challenges, we develop a novel end-to-end deep learning approach called geometric-aware attention network (GAANet) by combining neural network architecture and domain-specific geometric constraints. To acquire the datasets and evaluate the applicability of our developed network, we built up a two-segment TDCR with variable cross-sections, which is equipped with load cells and a motion capture system to measure the tension of tendons and the tip position of each segment simultaneously. To prove the efficacy of the developed GAANet, a multilayer perceptron (MLP) trained with the same dataset and model-based approach called pseudo rigid body model (PRBM) are both implemented for comparison. The results show that the developed GAANet outperforms the MLP and PRBM by 70.33% and 73.36% respectively in terms of the tip position prediction accuracy of each segment, demonstrating its potential for precise and real-time static modeling for our developed TDCRs with variable cross-sections.

Index Terms—Tendon-driven continuum robots, static modeling, deep learning.

I. INTRODUCTION

Tendon-driven continuum robots (TDCRs) inspired by invertebrates are widely researched in the field of robotics. The static modeling of TDCRs poses significant challenges due to the nonlinearities (e.g., non-constant cross-sections, friction, and anisotropic material properties) inside the TDCRs' structures and actuation mechanisms [1]–[3]. Conventional model-based approaches relying on certain simplified assumptions might lead to modeling inaccuracy [4]. However, data-driven approaches such as neural networks offer a promising alternative by learning complex and nonlinear

relationships between the actuation strength and TDCRs' shapes or tip positions [5]–[10].

Recently, physics-based learning models that include physical constraints and physical constitutive equations in the neural network have been widely studied in the modeling and control of robotics [11]. Also, the physics-based learning model can be more interpretable and robust than the traditional black-box models such as convolutional neural networks [12]. Moreover, compared to black-box models, physics-based learning models enhance generalization and can be trained effectively even with limited data. Therefore, this work introduces the development of a novel end-to-end approach for static modeling of TDCRs using a neural network that combines an encoder-decoder framework with a self-weighted attention mechanism. Inspired by [13] and [14], the developed architecture of the neural network maps the relationship between tendon tension and tip position of each segment of the TDCR while enhancing feature extraction through attention. Geometric constraints reflecting the TDCR's physical properties are embedded in the loss function to ensure physical feasibility and guide neural network optimization. The developed network is called geometric-aware attention network (GAANet), and its efficacy is demonstrated through comparisons with a multilayer perceptron (MLP) trained on the same dataset and a model-based approach called pseudo-rigid body model (PRBM) developed in [15].

The main contributions of this work are listed below:

- To effectively learn the mapping between the tendon tension and tip position of TDCR's each segment, an encoder-decoder architecture with a self-weighted attention mechanism is developed to map the tension-position relationship.
- Geometric constraints are embedded as penalty terms in the final loss function to guide the neural network's optimization direction, ensuring that the learned transformations fulfil the physical constraints of the TDCR system. The weights corresponding to each constraint are adjusted adaptively, enhancing the model's adaptability during different training stages.

The remaining part of the paper is organized as follows: In Section II, a generalized forward static model of the TDCR is formulated. In Section III, the design of the TDCR platform with variable cross-sections is demonstrated, and the data acquisition procedures are described. Section IV elucidates the development of the neural network embedded with physical constraints. Experimental results and discussion are shown

*This work was supported by the Research Grant Council of the Hong Kong Special Administrative Region, China, under Grant 15213621. (Xin Xu and Yang Song contributed equally to this work.) (Corresponding author: Henry K. Chu.)

¹Xin Xu, Yang Song, Bowen Su, Wendi Liang, and Henry K. Chu are with the Department of Mechanical Engineering, Hong Kong Polytechnic University, Hong Kong, SAR, China (e-mail: xin-21.xu@connect.polyu.hk; yangss.song@connect.polyu.hk; saber.su@connect.polyu.hk; wendi.liang@connect.polyu.hk; henry.chu@polyu.edu.hk).

²Ke Wu is with the Robotics Department, Mohamed bin Zayed University of Artificial Intelligence, Masdar City, Abu Dhabi 23204, UAE (e-mail: Ke.Wu@mbzuai.ac.ae).

³Di Wu is with the Maersk Mc-Kinney Moller Institute, University of Southern Denmark, 5230, Odense, Denmark (e-mail: diwu@mmmi.sdu.dk).

⁴Gang Zheng is with the Inria, CNRS, Centrale Lille.UMR 9189 CRISTAL, University of Lille, F-59000 Lille, France (e-mail: gang.zheng@inria.fr).

and offered in Section V. Section VI concludes this paper and gives the future directions.

II. FORMULATION OF THE STATIC MODELING OF TDCRS

This paper aims to develop a physics-embedded learning model for the static modeling of the TDCR. The forward static model of the TDCR can be represented by (1), which maps the relation between the tension of the actuated tendons and the state of the TDCR along the backbone.

$$u = f(T) \quad (1)$$

where $u \in \mathbb{R}^{3n}$ represents the n state of the TDCR, f is the generalized forward static model of the TDCR, and $T \in \mathbb{R}^m$ is a vector that contains the tension of m tendons. This work mainly focuses on the forward static modeling of the TDCR to prove the effectiveness of our developed physics-based learning model. In this study, we simplify the representation of u and define $u = [u_p, u_d]$, which represents only the tip position of the proximal segment and the distal segment of our developed TDCR, respectively, in the Cartesian space.

III. HARDWARE SETUP AND DATA ACQUISITION

In this section, the hardware setup and data acquisition workflow are described in detail.

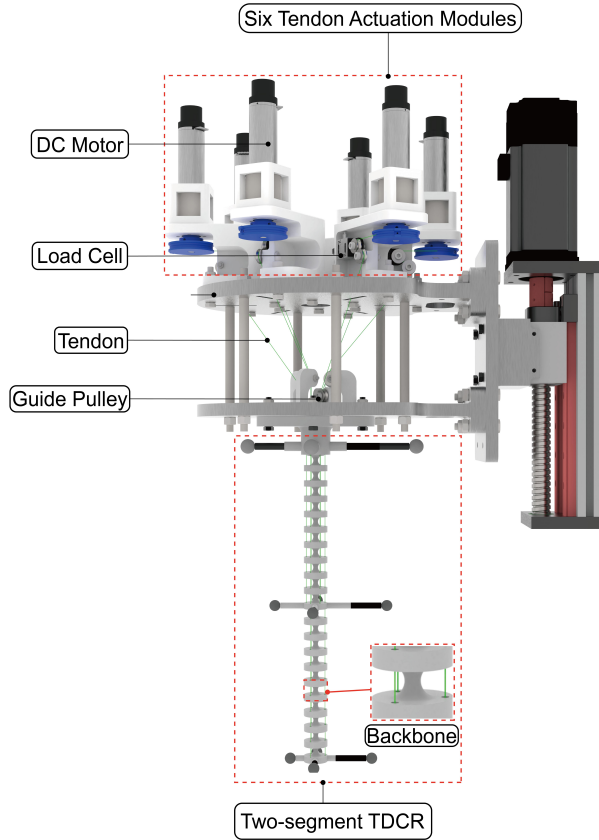


Fig. 1. Detailed View of Developed TDCR Mechatronic System.

As shown in Fig. 1 and 2a, we developed a two-segment TDCR with variable cross-sections (the diameters of the

backbone vary from 4 mm to 12 mm in each subsegment), which is challenging in terms of static modeling since it is hard to model this kind of TDCR with the traditional model-based approach (e.g., Cosserat rod theory). The proximal segment and distal segment of the TDCR were separately fabricated by a 3D printer (Bambu Lab, X1C, Shenzhen, China) using TPU (Thermoplastic polyurethane) 95A filament. The length of each segment is 135 mm and the total length of the TDCR is 270 mm. And the length of the proximal segment is denoted by L_p whereas the length of the distal segment is represented by L_d , as seen in Fig. 2a. The deformation of the two-segment TDCR is controlled by the displacements of six tendons. The tendons that we utilize in this TDCR are braid fishing lines (Spectra, PowerPro Inc., Irvine, CA, United States). As shown in Fig. 1 and 2a, each segment is driven by three tendons respectively, which are evenly spaced at 120° intervals on the disks and knotted from the end disks of each segment to the individual spools actuated by DC motors (EC-max 22, Maxon Motors Inc., Switzerland). Meanwhile, to measure the tension of the driving tendons in real-time, load cells (FSSM, Forsentek Co., Ltd, Shenzhen, China) are integrated into the tendon actuation modules.

To measure the tip position of each segment of TDCR simultaneously, this mechatronic system shown in Fig. 2b is equipped with a real-time 3D motion capture system containing four cameras (Flex13, NaturalPoint, Inc., Corvallis, United States) that enables the position tracking of the TDCR at the rate of 120 Hz with the accuracy of ± 0.2 mm. Note that the extensibility and compressibility of our fabricated TDCR are ignored due to the limited range of the actuated tendon tension (from 0 N to 4.9 N) and the relatively high infill density of the 3D-printed TDCR structure.

To construct the datasets for our proposed neural network and ensure that the datasets include diverse configurations of the TDCR, the TDCR was then deformed into different C-shapes and S-shapes in the quasi-static state by commanding six DC motors to pull or release the tendons (each shape represents one trial). Meanwhile, the tendons' tension was measured by load cells and the tip position of each segment of TDCR was collected by the 3D motion capture system. Both types of data were recorded simultaneously at a fixed frequency of 120 Hz during one trial and the number of the collected data points on each trial was equal. To validate the developed method for TDCR with different load conditions, we installed different standard weights randomly (5 g, 10 g, 20 g and 50 g respectively) as the payload at the end tip of the TDCR. To obtain the training and validation datasets, 24 trials were conducted without payloads, and 26 trials were conducted under payloads.

IV. DESIGN OF NEURAL NETWORK

In this section, we describe the details of our developed neural network architecture and the MLP. For the neural network training, the tension of the six tendons is the input, and the tip positions of each segment of the TDCR are the output. A total of 3465 samples were collected without

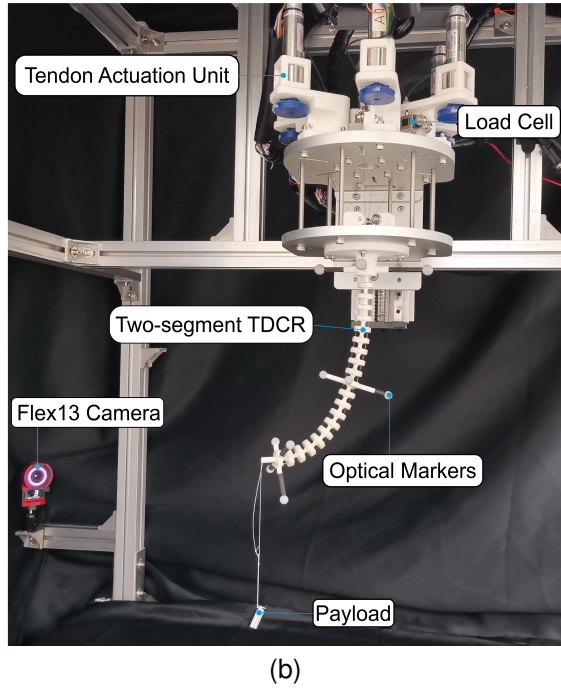
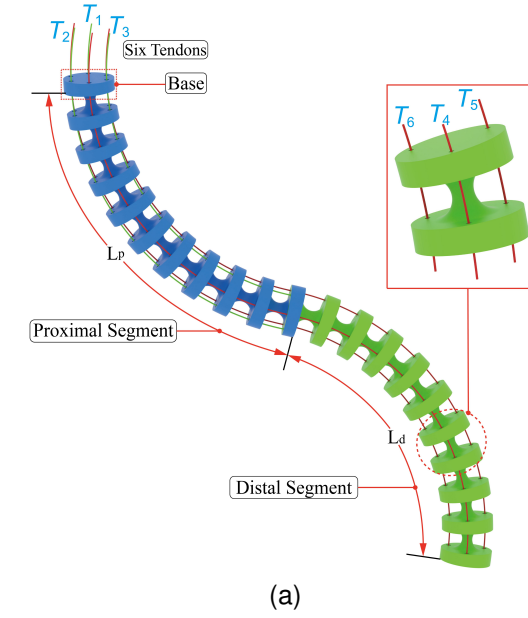


Fig. 2. Two-segment TDCR Platform. (a) TDCR with variable cross-section (The proximal segment is highlighted in blue while the distal segment is highlighted in green); (b) snapshot of the complete mechatronic system setup of our developed TDCR (including the motion capture system).

payloads, and 6052 samples were collected with payloads from the 24 and 26 trials respectively. The dataset was randomly divided into training, validation, and test sets using an 80%–20% split. Specifically, 80% of the data was used for training, and 10% of the training data was randomly set for validation. The remaining 20% was used as the test set. The training dataset includes both the samples with and without payloads. All computation was performed on a desktop computer equipped with an Intel(R) Core(TM) i7-14700F CPU with 2.10 GHz clock speed and an NVIDIA

RTX 3060 Ti GPU. The programming environment includes Python 3.8, with TensorFlow 2.10.

A. Multilayer Perceptron

Multilayer perceptron (MLP), as illustrated in Fig. 3a, is a feedforward neural network consisting of an input layer, three hidden layers, and an output layer [16]. Neurons connect via weighted connections, allowing the network to learn complex patterns using backpropagation and gradient descent as demonstrated in equation (2). And MLP is commonly used in the modeling and control of continuum robots (CRs) [4]. A pioneering work that utilized MLP as a feedforward compensator in the control of the CRs is [17], which opened a new direction for using neural networks in the modeling and control of the CRs. In this work, MLP is implemented and compared with our developed neural network.

$$a_k^{(L)} = \sigma \left(\sum_{j=1}^{n^{(L-1)}} w_{jk}^{(L)} a_j^{(L-1)} + b_k^{(L)} \right) \quad (2)$$

The notations of (2) are listed as follows:

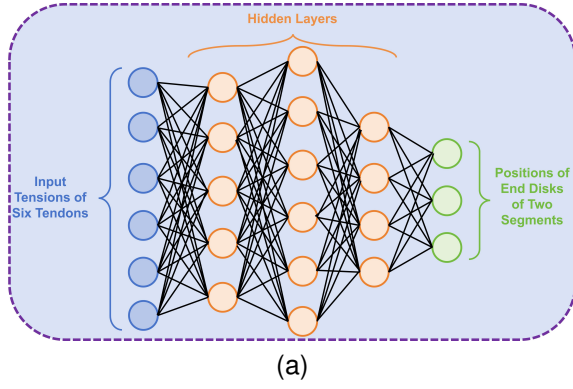
- $a_k^{(L)}$ is the output of the k -th neuron in the L -th layer after applying the ReLU function.
- $w_{jk}^{(L)}$ is the weight from the j -th neuron in the $(L-1)$ -th layer to the k -th neuron in the L -th layer.
- $a_j^{(L-1)}$ is the activation of the j -th neuron in the previous layer.
- $b_k^{(L)}$ is the bias for the k -th neuron in the L -th layer.
- σ is the activation function applied to $z_k^{(L)}$, defined as $\text{ReLU}(x) = \max(0, x)$.

B. Geometric-aware Attention Network

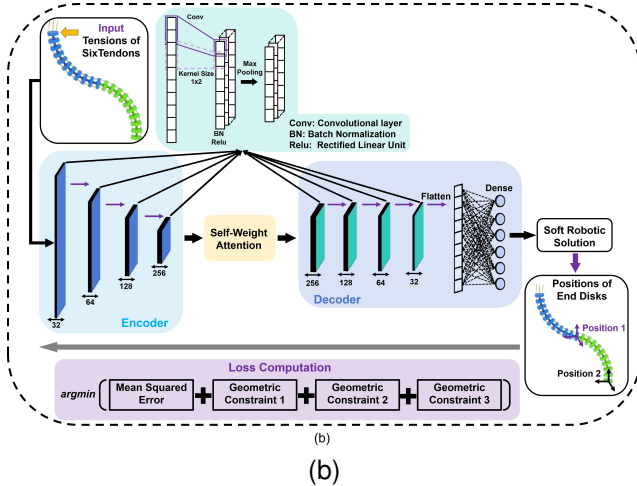
The proposed architecture, as shown in Fig. 3b, maps tendon tensions T to tip positions x , transforming high-dimensional input data into a latent space for effective feature extraction. This neural network architecture adopts an encoder-decoder structure, where both the encoder and decoder primarily consist of convolutional layers. A self-weight attention mechanism is introduced in the latent space to further optimize feature extraction.

The encoder comprises multiple convolutional layers, each incorporating convolution operations, batch normalization, ReLU activation, and dropout (with a rate of 0.1) to prevent overfitting. The convolution kernel size is 1×2 , allowing the network to capture local dependencies within the tension data. After each convolutional layer, max-pooling is applied to reduce the spatial dimensions of the data, thereby decreasing computational complexity.

Following the encoder, a self-weight attention mechanism is introduced to emphasize features that are critical for tip position prediction, especially in the context of TDCRs. In TDCRs, the relationship between tendon tensions and the tip positions is highly nonlinear and affected by complex factors such as the TDCR's geometry, non-homogeneous material properties and tendon routing patterns. The attention mechanism is designed to focus on the most relevant features of this high-dimensional input data, helping the model to



(a)



(b)

Fig. 3. Architecture of the two neural networks applied for the static modeling of the TDCR. (a) multilayer perceptron (MLP); (b) geometric-aware attention network (GAANet)

better capture the intricate dependencies between tendon tension and tip positions.

The attention mechanism works by assigning different weights to the input features based on their relevance to the tip positions prediction task. In mathematical terms, given a set of features, the attention mechanism computes a weight for each feature by comparing it to other features using a query (Q), key (K), and value (V) representation. The output of this mechanism is a weighted combination of the features, where features with higher relevance to the predicted positions receive higher attention. This mechanism helps the model to focus on key aspects of the input, such as tendon tensions that have a more direct effect on the tip positions, while less relevant information is down-weighted.

The equation governing the attention mechanism is given as:

$$\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) V \quad (3)$$

where Q , K , and V represent the query, key, and value vectors, respectively, and d_k is a scaling factor. The inclusion of this self-weight attention mechanism is crucial for improving the model's ability to learn the most relevant features, enabling it to map complex tendon tensions to accurate tip positions by highlighting the most informative parts of the

input data.

After the attention mechanism, the decoder module progressively decodes the latent features back into the output space using a series of transposed convolutional layers. These layers aim to restore the dimensionality of the data, converting it into a representation closer to the end position of the TDCR. The decoder is structured symmetrically to the encoder, ensuring that the relevant features learned through attention are effectively utilized in the final position predictions. The features are then flattened and passed through a fully connected layer, which produces the final output—the predicted tip position of each segment. The GAANet model

Algorithm 1 Custom Loss Function with Improved Dynamic Penalties

- 1: **Input:** Initial weights (w_1, w_2, w_3) , decay rate δ , increase factor α , threshold τ , true values y_{true} , predicted values y_{pred} , epoch e
- 2: **Output:** Total loss L_{total} , MSE loss L_{mse}
- 3: Initialize current weights as $\text{current_weights} \leftarrow (w_1, w_2, w_3)$
- 4:
- 5: **Function** $\text{update_weights}(e, L_{\text{mse}})$
- 6: **for** $i = 1$ **to** 3 **do**
- 7: $\text{current_weights}[i] \leftarrow \text{current_weights}[i] \cdot e^{-\delta \cdot e}$
- 8: **end for**
- 9: **if** $L_{\text{mse}} > \tau$ **then**
- 10: **for** $i = 1$ **to** 3 **do**
- 11: $\text{current_weights}[i] \leftarrow \text{current_weights}[i] \cdot \alpha$
- 12: **end for**
- 13: **end if**
- 14: **End Function**
- 15:
- 16: **Function call** $(\mathbf{u}_{\text{true}}, \mathbf{u}_{\text{pred}})$
- 17: $L_{\text{mse}} \leftarrow \frac{1}{n} \sum (\mathbf{u}_{\text{true},i} - \mathbf{u}_{\text{pred},i})^2$
- 18: $\text{pos}_1 \leftarrow \mathbf{u}_1$
- 19: $\text{pos}_2 \leftarrow \mathbf{u}_2$
- 20: $\text{distance} \leftarrow \|\text{pos}_1 - \text{pos}_2\|$
- 21: $\text{distance}_1 \leftarrow \|\text{pos}_1\|$
- 22: $\text{distance}_2 \leftarrow \|\text{pos}_2\|$
- 23: $\text{distance_penalty} \leftarrow \frac{1}{n} \sum \max(0, \text{distance} - L_d)^2$
- 24: $\text{distance_penalty}_1 \leftarrow \frac{1}{n} \sum \max(0, \text{distance}_1 - L_p)^2$
- 25: $\text{distance_penalty}_2 \leftarrow \frac{1}{n} \sum \max(0, \text{distance}_2 - (L_p + L_d))^2$
- 26: $L_{\text{total}} \leftarrow L_{\text{mse}} + \frac{\text{current_weights}[0]}{\text{distance_penalty} + \text{current_weights}[1] + \text{distance_penalty}_1 + \text{current_weights}[2] + \text{distance_penalty}_2}$
- 27: **return** $(L_{\text{total}}, L_{\text{mse}})$
- 28: **End Function**

is trained with the Adam optimizer, using a learning rate of 1×10^{-4} to balance convergence speed and stability. Training is conducted over 200 epochs, with a mini-batch size of 28 samples, to ensure proper model convergence. Based on the assumption that the compression and extension of our developed TDCR are ignored, the overall loss function L is

defined as a combination of the Mean Squared Error (MSE) and three geometric penalty terms, as shown in (4).

$$L = \text{MSE} + w_1 P_1 + w_2 P_2 + w_3 P_3 \quad (4)$$

where $\text{MSE} = \frac{1}{n} \sum_{i=1}^n (\mathbf{u}_{\text{true},i} - \mathbf{u}_{\text{pred},i})^2$. Also, P_1 represents the first geometric penalty term, which enforces a constraint on the predicted distances between the tip positions of two segments, defined in (5) below.

$$P_1 = \frac{1}{n} \sum_{i=1}^n \max \left(0, \|\mathbf{u}_1^{(i)} - \mathbf{u}_2^{(i)}\| - L_d \right)^2 \quad (5)$$

Moreover, P_2 denotes the second geometric penalty term related to the total length of the developed TDCR, defined by (6) below.

$$P_2 = \frac{1}{n} \sum_{i=1}^n \max \left(0, \|\mathbf{u}_2^{(i)}\| - (L_d + L_P) \right)^2 \quad (6)$$

Finally, P_3 denotes the third geometric penalty term related to the tip position of the first segment, defined by (7) below.

$$P_3 = \frac{1}{n} \sum_{i=1}^n \max \left(0, \|\mathbf{u}_1^{(i)}\| - (L_P) \right)^2 \quad (7)$$

To better determine the weighting factors for different constraints, the algorithm shown in Algorithm IV-B implements a custom dynamic weighted loss function for a dataset with weights. It adjusts the weights of the loss components based on the Mean Squared Error (MSE) and applies geometric penalties related to the distances of predicted values. The current weights decay over epochs and are scaled up if the MSE exceeds a predefined threshold. The total loss is calculated by combining the MSE with penalties that enforce geometric constraints on the predicted positions, ensuring better model performance by adaptively tuning the contributions of each penalty based on the learning progress.

V. RESULTS AND DISCUSSION

In this section, the results of GAANet, MLP, and PRBM are demonstrated and discussed in detail. The general evaluation criteria of different approaches are the average tip position error with respect to the total length of TDCR. To begin with, the implementation of the PRBM necessitates a calibration for the material parameters of the model, such as Young's modulus E , shear modulus G , the backbone radius r , and also the mass of each disk m . Therefore, the calibration was conducted with MATLAB 2024b using the genetic algorithm (GA) with 24 randomly selected data points collected from the 24 trials without the payload. Furthermore, the results of PRBM were also evaluated in MATLAB with the same validation datasets of MLP and GAANet.

Regarding the average tip position prediction accuracy evaluated from all the validation datasets covering 9517 samples in total, the PRBM achieved 7.12 ± 3.83 mm ($2.74 \pm 1.47\%$ of the total length of TDCR), the MLP achieved 6.40 ± 7.70 mm ($2.46 \pm 2.96\%$ of the total length of TDCR), whereas the GAANet reached 1.90 ± 4.49 mm ($0.73 \pm 1.73\%$

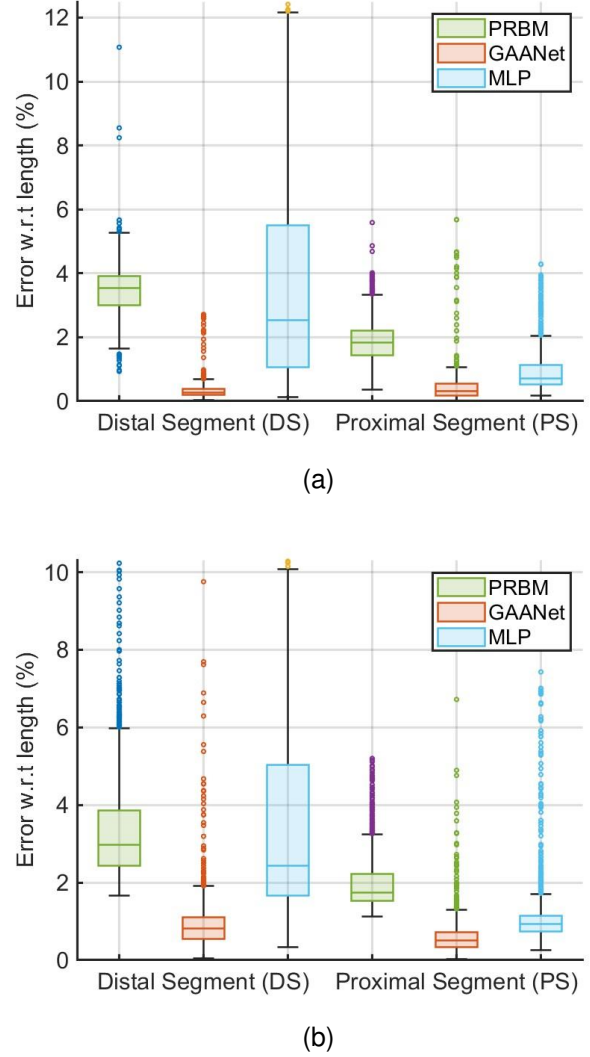


Fig. 4. Box plots comparison of tip position errors over the different validation datasets. (a) TDCR without payload (b) TDCR under different payloads.

of the total length of TDCR). In terms of the tip position prediction accuracy of each segment, it can be observed from the results that the developed GAANet outperforms the MLP and PRBM by 70.33% and 73.36% respectively. Also, the average computation time from tension input to the output of tip positions of PRBM, MLP, and GAANet is 711.20 ms, 0.072 ms, and 0.460 ms, respectively. The resulting tip position errors with or without the payloads are demonstrated in Fig. 4, and the statistical results are listed in Table I, which both indicate that the GAANet is more accurate for predicting the tip position of each segment of the TDCR with or without payloads than PRBM and MLP. The GAANet outperforms the MLP in terms of tip position errors due to its advanced feature extraction capabilities through an encoder-decoder architecture and self-weight attention mechanism. Moreover, the inclusion of geometric constraints in GAANet's loss function ensures physical feasibility in the optimization process to have more accurate and stable results.

In summary, the MLP’s more straightforward structure and lack of attention mechanism and physical constraints result in higher variance and imprecise predictions, as shown in Fig. 4a and Fig. 4b. However, the computation efficiency of the MLP is higher than the developed GAANet due to its less complex network architecture. Nonetheless, our current validation is limited to our developed TDCR with a specific geometry and certain tendon routing pattern. We will further evaluate our developed model on different TDCRs to further prove its applicability in the future.

In this study, we conducted ablation experiments on a dataset with payloads to evaluate the impact of different conditions on model performance. Specifically, we examined four scenarios: using both the attention mechanism and geometric constraints, using only the attention mechanism, using only geometric constraints, and using neither. The results, as shown in Table II and Table III, indicate that the combination of the attention mechanism and geometric constraints significantly improves the model’s mean squared error (MSE) and mean absolute error (MAE). The reported MSE and MAE values, measured in millimetres (mm), indicate that the model achieves high precision in predicting the tip positions of the TDCR. The improvement in MSE and MAE with the inclusion of the attention mechanism and geometric constraints highlights the model’s ability to deliver highly accurate predictions, which is essential for the reliable operation of TDCRs. Removing either mechanism leads to a performance drop, further emphasizing the critical role of the synergistic effect between the attention mechanism and geometric constraints in enhancing the prediction accuracy of static modeling.

In summary, our findings indicate that the proposed attention mechanism, combined with geometric constraints in the loss function, can effectively enhance the prediction accuracy of our developed GAANet.

TABLE I
ERROR STATISTICS OF VALIDATION

24 Trials without Payloads			24 Trials without Payloads		
DS ² Tip Error(%)	mean	Std Dev	PS ¹ Tip Error(%)	mean	Std Dev
PRBM ³	3.54	0.97	PRBM ³	1.89	0.71
MLP ⁴	3.92	4.44	MLP ⁴	1.01	0.75
GAANet ⁵	0.35	0.34	GAANet ⁵	0.52	1.20

26 Trials with Payloads			26 Trials with Payloads		
DS ² Tip Error(%)	mean	Std Dev	PS ¹ Tip Error(%)	mean	Std Dev
PRBM ³	3.52	1.89	PRBM ³	1.98	0.65
MLP ⁴	3.71	2.98	MLP ⁴	1.21	1.23
GAANet ⁵	1.03	2.24	GAANet ⁵	0.77	1.81

¹ PS represents the proximal segment.

² DS represents the distal segment.

³ PRBM represents the pseudo-rigid-body-model.

⁴ MLP represents the multilayer perceptron.

⁵ GAANet represents the geometric-aware attention network.

VI. CONCLUSION

Physics-based learning models have become a new trend in the modeling and control of CRs due to their higher

TABLE II
ABLATION EXPERIMENT RESULTS FOR NO PAYLOAD DATASET

Condition	MSE(mm)	MAE(mm)
With Attention & Geometric Constraints	1.033×10^{-1}	2.743
Without Attention & Geometric Constraints	1.050×10^{-1}	2.881
With Attention & Without Geometric Constraints	1.112×10^{-1}	2.625
Without Attention & Geometric Constraints	1.194×10^{-1}	2.933

TABLE III
ABLATION EXPERIMENT RESULTS FOR PAYLOAD DATASET

Condition	MSE(mm)	MAE(mm)
With Attention & Geometric Constraints	3.371×10^{-1}	5.537
Without Attention & Geometric Constraints	3.394×10^{-1}	5.813
With Attention & Without Geometric Constraints	3.706×10^{-1}	6.870
Without Attention & Geometric Constraints	4.074×10^{-1}	8.004

accuracy and robustness. In this work, we developed a novel neural network for static modeling of TDCR with complex cross-sections, which achieves higher precision compared to the model-based approach PRBM and data-driven approach MLP. The results show that the proposed method: 1) can predict the tip position of each segment merely based on tension inputs of the TDCR; 2) has a higher modeling accuracy compared to the selected analytic model PRBM and conventional data-driven approach MLP. It is important to note that the current GAANet design does not consider dynamic characteristics, such as hysteresis effects, which could influence the model’s accuracy in dynamic scenarios where past configurations influence the current state. The quality of the dataset can contribute to network errors, as the raw data used in this study does not include data in dynamic motion cases during data collection.

Furthermore, the network that we developed can be generalized to different TDCRs and we will first evaluate our developed GAANet on different TDCRs in the future. Subsequently, efforts will be directed toward combining recurrent neural networks (RNN) and our developed GAANet to cope with the time series datasets and extend the applicability of the GAANet in the dynamic modeling of the TDCRs.

REFERENCES

- [1] W. Ba, J.-C. Chang, J. Liu, X. Wang, X. Dong, and D. Axinte, “Analytical differential kinematics-based method for controlling tendon-driven continuum robots,” *Robotics and Autonomous Systems*, vol. 171, p. 104562, 2024.
- [2] D. Jakes, Z. Ge, and L. Wu, “Model-less active compliance for continuum robots using recurrent neural networks,” in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 2167–2173.
- [3] B. Y. Cho, D. S. Esser, J. Thompson, B. Thach, R. J. Webster, and A. Kuntz, “Accounting for hysteresis in the forward kinematics of

- nonlinearly-routed tendon-driven continuum robots via a learned deep decoder network,” *IEEE Robotics and Automation Letters*, 2024.
- [4] Z. Chen, F. Renda, A. Le Gall, L. Mocellin, M. Bernabei, T. Dangel, G. Ciuti, M. Cianchetti, and C. Stefanini, “Data-driven methods applied to soft robot modeling and control: A review,” *IEEE Transactions on Automation Science and Engineering*, 2024.
- [5] X. T. Ha, D. Wu, M. Ourak, G. Borghesan, J. Dankelman, A. Menciassi, and E. Vander Poorten, “Shape sensing of flexible robots based on deep learning,” *IEEE Transactions on Robotics*, vol. 39, no. 2, pp. 1580–1593, 2022.
- [6] D. Kim, S.-H. Kim, T. Kim, B. B. Kang, M. Lee, W. Park, S. Ku, D. Kim, J. Kwon, H. Lee *et al.*, “Review of machine learning methods in soft robotics,” *Plos one*, vol. 16, no. 2, p. e0246102, 2021.
- [7] D. Wu, R. Zhang, A. Pore, D. Dall’Alba, X. T. Ha, Z. Li, Y. Zhang, F. Herrera, M. Ourak, W. Kowalczyk *et al.*, “A review on machine learning in flexible surgical and interventional robots: Where we are and where we are going,” *Biomedical Signal Processing and Control*, vol. 93, p. 106179, 2024.
- [8] 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 Transactions on Robotics*, vol. 39, no. 4, pp. 2973–2988, 2023.
- [9] H. El-Hussieny, I. A. Hameed, and A. A. Nada, “Deep cnn-based static modeling of soft robots utilizing absolute nodal coordinate formulation,” *Biomimetics*, vol. 8, no. 8, p. 611, 2023.
- [10] T. George Thuruthel, E. Falotico, M. Manti, A. Pratesi, M. Cianchetti, and C. Laschi, “Learning closed loop kinematic controllers for continuum manipulators in unstructured environments,” *Soft robotics*, vol. 4, no. 3, pp. 285–296, 2017.
- [11] W. Deng, F. Ardiani, K. T. Nguyen, M. Benoussaad, and K. Medjaher, “Physics informed machine learning model for inverse dynamics in robotic manipulators,” *Applied Soft Computing*, p. 111877, 2024.
- [12] J. Liu, P. Borja, and C. Della Santina, “Physics-informed neural networks to model and control robots: A theoretical and experimental investigation,” *Advanced Intelligent Systems*, vol. 6, no. 5, p. 2300385, 2024.
- [13] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, “On the properties of neural machine translation: Encoder-decoder approaches,” *arXiv preprint arXiv:1409.1259*, 2014.
- [14] Z. Niu, G. Zhong, and H. Yu, “A review on the attention mechanism of deep learning,” *Neurocomputing*, vol. 452, pp. 48–62, 2021.
- [15] C. Troeung, S. Liu, and C. Chen, “Modelling of tendon-driven continuum robot based on constraint analysis and pseudo-rigid body model,” *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 989–996, 2023.
- [16] W. H. Delashmit, M. T. Manry *et al.*, “Recent developments in multilayer perceptron neural networks,” in *Proceedings of the seventh annual memphis area engineering and science conference, MAESC*, vol. 7, 2005, p. 33.
- [17] D. Braganza, D. M. Dawson, I. D. Walker, and N. Nath, “A neural network controller for continuum robots,” *IEEE transactions on robotics*, vol. 23, no. 6, pp. 1270–1277, 2007.