Shape sensing of optical fiber Bragg gratings based on deep learning

IOP Publishing Mach. Learn.: Sci. Technol. 4 (2023) 025037 https://doi.org/10.1088/2632-2153/acda10

MACHINE LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
13 February 2023

REVISED
17 May 2023

ACCEPTED FOR PUBLICATION
30 May 2023

PUBLISHED
3 July 2023

PAPER

Shape sensing of optical fiber Bragg gratings based on deep learning

Samaneh Manavi Roodsari[1,*], Antal Huck-Horvath[1], Sara Freund[1], Azhar Zam[2,3], Georg Rauter[4], Wolfgang Schade[5] and Philippe C Cattin[1]

[1] Center for Medical Image Analysis and Navigation (CIAN), Department of Biomedical Engineering, University of Basel, Hegenheimermattweg 167C, Allschwil 4123, Switzerland
[2] Division of Engineering, New York University Abu Dhabi, Abu Dhabi 129188, United Arab Emirates
[3] Tandon School of Engineering, New York University, Brooklyn, NY 11201, United States of America
[4] Bio-Inspired RObots for MEDicine-Laboratory (BIROMED-Lab), Department of Biomedical Engineering, University of Basel, Allschwil 4123, Switzerland
[5] Department of Fiber Optical Sensor Systems, Fraunhofer Institute for Telecommunications, Heinrich Hertz Institute, HHI, Am Stollen 19H, Goslar 38640, Germany
[*] Author to whom any correspondence should be addressed.

E-mail: samaneh.manavi@unibas.ch

Keywords: supervised deep learning, shape sensing, bending birefringence, bending loss, eccentric FBG, fiber sensor, curvature sensing

Supplementary material for this article is available online

Abstract

Continuum robots in robot-assisted minimally invasive surgeries provide adequate access to target anatomies that are not directly reachable through small incisions. Achieving precise and reliable shape estimation of such snake-like manipulators necessitates an accurate navigation system, that requires no line-of-sight and is immune to electromagnetic noise. Fiber Bragg grating (FBG) shape sensing, particularly eccentric FBG (eFBG), is a promising and cost-effective solution for this task. However, in eFBG sensors, the spectral intensity of the Bragg wavelengths that carries the strain information can be affected by undesired bending-induced phenomena, making standard characterization techniques less suitable for these sensors. We showed in our previous work that a deep learning model has the potential to extract the strain information from the eFBG sensor's spectrum and accurately predict its shape. In this paper, we conducted a more thorough investigation to find a suitable architectural design of the deep learning model to further increase shape prediction accuracy. We used the Hyperband algorithm to search for optimal hyperparameters in two steps. First, we limited the search space to layer settings of the network, from which, the best-performing configuration was selected. Then, we modified the search space for tuning the training and loss calculation hyperparameters. We also analyzed various data transformations on the network's input and output variables, as data rescaling can directly influence the model's performance. Additionally, we performed discriminative training using the Siamese network architecture that employs two convolutional neural networks (CNN) with identical parameters to learn similarity metrics between the spectra of similar target values. The best-performing network architecture among all evaluated configurations can predict the shape of a 30 cm long sensor with a median tip error of 3.11 mm in a curvature range of 1.4 m^{-1} to 35.3 m^{-1}.

1. Introduction

Minimally invasive surgeries (MISs) are delicate operations performed through small incisions or natural orifices on anatomical structures of the human body. Such interventions are beneficial compared to conventional open surgeries, as they reduce patient trauma, shorten recovery time [1], and are cost-effective [2]. In traditional MIS procedures, surgeons often have to adapt themselves to counterintuitive ergonomic principles, as haptics, dexterity, and visual-motor coordination of the human hand are replaced by rigid instruments. The integration of robotic technologies to assist surgeons during such interventions enables

enhanced dexterity, manipulability, stability, and motion accuracy [2]. In addition, small, flexible, and strong manipulators, also known as continuum robots, are particularly of interest, especially for MIS procedures in which adequate and direct access through small incisions to target anatomies is not available [1, 3–8]. However, the inherent snake-like and deformable designs of continuum robots make it difficult to accurately estimate their shape using model-based approaches [9, 10]. Therefore, a precise and accurate navigation system is desirable for enabling closed-loop control to better follow the planned pathways and alleviate safety concerns.

The most common and commercially available medical tracking systems include optical trackers [11, 12], electromagnetic (EM) sensors [11], intraoperative imaging technologies [13], and fiber Bragg grating (FBG)-based sensors [14]. Optical trackers are state-of-the-art technology for tracking medical tools and patients inside the operating room (OR). Wireless tracking, accurate measurement, and stable performance are the key advantages of this technology. However, they require a line-of-sight and are best suited to use with large rigid tools. EM tracking systems do not require a line-of-sight and can be embedded or placed at the tip of flexible tools. However, EM tracking systems are less accurate than optical trackers and have a smaller working volume. Adding multiple sensors along the endoscope is often impossible, as the sensors must be wired. Furthermore, they are sensitive to environmental EM interferences (e.g. the EM field of the robot) and the presence of conductive or ferromagnetic metals. Intraoperative imaging modalities, including fluoroscopy, cone-beam CT, and ultrasound, can be alternatives to EM sensors for intracorporeal tracking. Some imaging modalities like biplane fluoroscopy can achieve even higher accuracy compared to EM sensors [13], but are challenging to handle in crowded OR settings. In addition, imaging modalities have limitations such as high doses of radiation (e.g. X-ray-based imaging), high computational cost (e.g. cone-beam CT), and low resolution (e.g. ultrasound).

FBG-based shape sensors, on the other hand, are easily integrable into medical devices, can carry an array of FBGs to extract shape information along the length of the fiber, and are immune to EM interferences. The coating layer of the optical fiber can be a bio-compatible material, which makes it suitable for tracking catheters as well [15]. Although different configurations for FBG-based shape sensors have been studied in recent years [16–21], the only fiber shape sensors that have been commercialized work based on multicore fibers (e.g. [22, 23]). Multicore fiber shape sensors are able to track themselves with a millimeter range accuracy (e.g. an average error of 1.13 mm for a 38 cm long sensor [19]). However, the cost of such systems is quite high, as a multichannel interrogator and a fan-out device would be needed for reading the signal from each core [24, 25].

Recently, a new configuration for fiber shape sensors, based on eccentric FBGs (eFBGs) [26–29], has been proposed by Waltermann *et al* [30]. In this configuration, several highly-localized FBGs are inscribed off-axis inside the core of a single-mode fiber at different angular positions. Unlike standard FBG shape sensors, which rely on changes in the Bragg wavelength, eFBG sensors are based on changes in the spectral intensity at Bragg wavelengths [26, 28–30]. Currently, shape deformations in eFBG shape sensors are calculated based on the mode-field displacement method (MFD) [27, 30]. In this approach, the shift of the mode-field's center at each sensing plane is calculated by the ratio between the spectral intensity of the eFBGs. To simplify the analysis, the MFD approach assumes that bending-induced birefringence [31] and cladding mode coupling [28, 32, 33] do not significantly affect the spectral intensity of the eFBGs. Therefore, the shape prediction accuracy can be relatively high [34] when low-cost readout units with grating-based spectrometers are used for eFBG sensor.

The authors showed in [35] that despite the complicated impact of bending-induced phenomena on the signal of eFBG sensors, it is feasible to use a deep learning model to predict the shape of a 30 cm long sensor based on the full spectra of eFBGs and were able to achieve a median tip error of 6 mm. In this paper, we investigated the design of such deep learning models in more detail. First, we identified a good set of tuning parameters, known as hyperparameters, for our deep learning algorithm to extract relevant features from the eFBG sensor. We performed this hyperparameter tuning when the model's input (the sensor's spectra) and output data (the sensor's spatial shape) were preprocessed using different rescaling methods. Ultimately, we employed the most suitable data rescaling approach and the optimized feature-extracting network to perform discriminative training using the Siamese network [36].

2. Methodology

The importance of choosing a good set of hyperparameters for a deep learning algorithm is well-known. The Hyperband, as one of the most common hyperparameter optimizers, considers several possible resource allocations (e.g. the total number of epochs used during evaluation) and invokes Successive Halving [37] on randomly sampled hyperparameter configurations [38]. Compared to black-box approaches like Bayesian optimization, the Hyperband is 5× to 30× faster and evaluates an order-of-magnitude more configurations

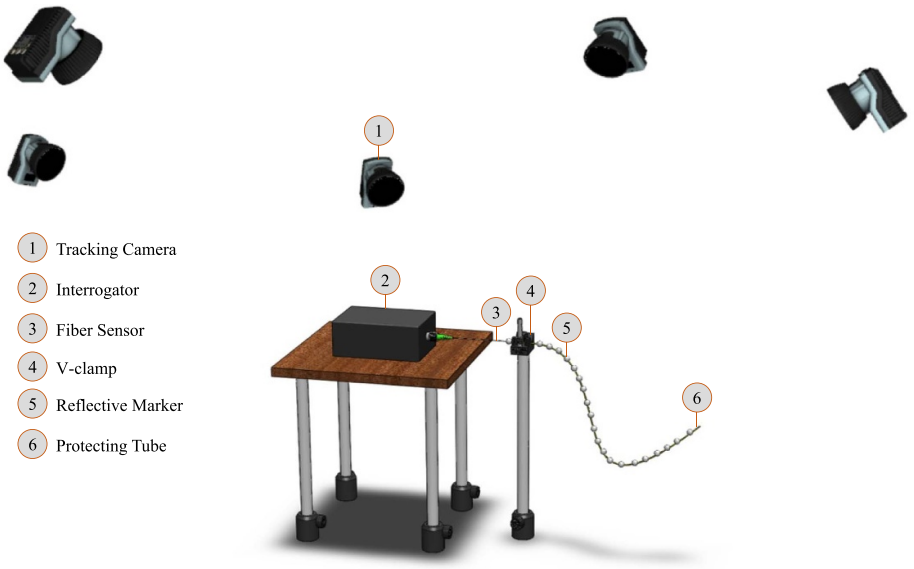

Figure 1. The data acquisition experimental setup of the eFBG sensor. Five tracking cameras (Oqus 7+, Qualisys AB, Sweden) were used to monitor the sensor's shape. The sensor was inserted into a Hytrel furcation tubing to protect the fiber during shape manipulation.

[38]. In addition to the hyperparameter configuration, rescaling the input and the output variables before being presented to the deep neural networks greatly affects the model's performance [39]. When input variables have large values, the model learns large weights, which may cause numerical instability, poor performance during training, and generalization error in the network. Large-scaled target values result in significant update rates in weight values, making the learning process unstable. In common practices, preprocessing transformations are applied to input variables prior to training the networks, and postprocessing steps are introduced to the model's predictions for calculating the desired target values [39]. Therefore, we investigated different rescaling methods on the input and output variables to identify the most suited data preprocessing steps.

2.1. Training setup

The dataset used in this work is from [35] and [40] with almost 53 000 samples. Each sample consists of three consecutively measured eFBG spectra, the intensity values of 125 wavelength components from 812 nm to 871 nm, as the input and the spatial coordinates of 21 discrete points along the fiber's length as corresponding target values. These discrete points are the positions of reflective markers that are attached to the sensor and monitored using a motion capture system. Figure 1 shows a schematic of the experimental setup. The sensor's spectrum contains the reflected signal of 15 eFBGs from five sensing planes. Each sensing plane has three eFBGs with $\sim 90^{\circ}$ angular separation inside a single-mode fiber's core (SM800p from FIBERCORE company, UK).

We used the Hyperband algorithm, built in the Keras tuner [41], to perform hyperparameter optimization in two steps. First, we defined a bigger search space to optimize the settings of 1D convolutional layers (conv1D) and pooling layers (search criteria can be found in table 3 in appendix). The number of conv1D layers was set to seven for this hyperparameter tuning step, as we observed that the network with seven conv1D layers is deep enough for feature extraction. Then, we fixed these layer settings and modified the search space to tune the loss function, the optimizer, and the dropout rate (more detail on the search criteria is available in table 4 in appendix). The model's input and output data were preprocessed using various data rescaling methods in this hyperparameter tuning step. The objective of the Hyperband was set to be the root-mean-square error (RMSE) on the validation set so that the scale of the scores, assigned to suggested configurations, would not be affected when different loss functions were selected. As the Hyperband is based on a random search, we repeated each hyperparameter search three times for statistical robustness before selecting the final settings. To evaluate the predictive performance of the proposed model in an unbiased way, the dataset was split into mutually disjoint Train-Validation-Test subsets: 80% for training, 10% for validating, and 10% for testing.

We investigated various configurations in the first hyperparameter tuning step. Figure 2 shows the best-performing architecture among the suggested configurations, in which, each conv1D layer is followed by

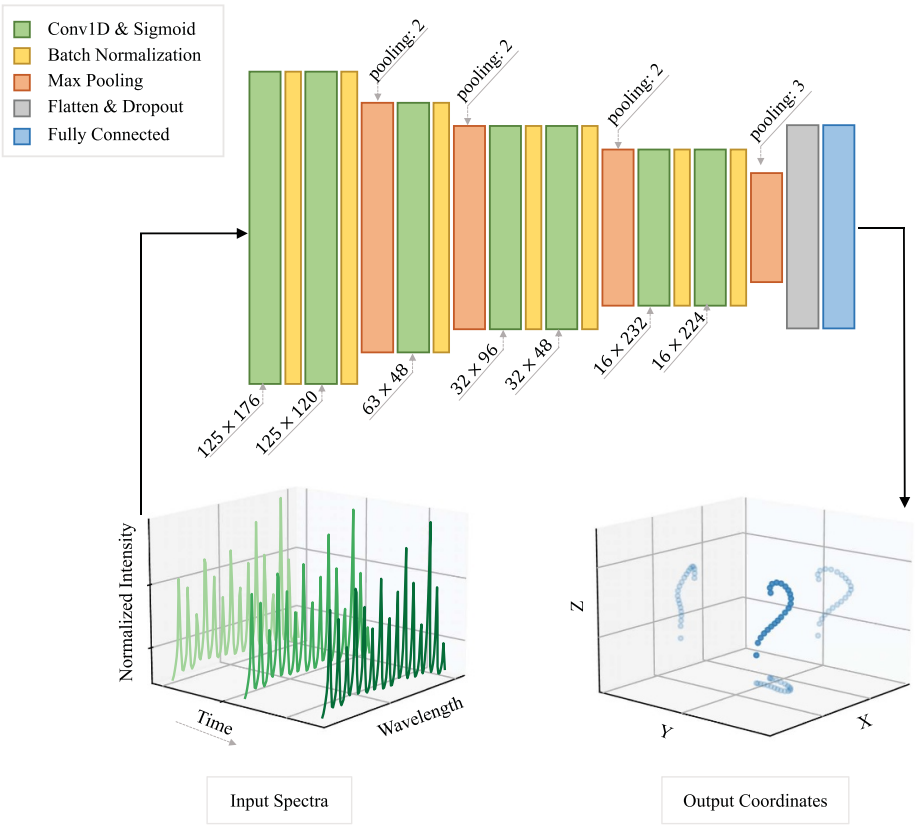

Figure 2. The architecture of the best-performing configuration after the first step of hyperparameter search. For this network, the input data is three consecutive eFBG spectra, normalized using one-dimensional z-scaling (see section 2.2 for more detail), and the output data is the absolute coordinates of 21 discrete points along the sensor's length. The channel sizes of the seven 1D convolution layers (conv1D) are 176, 120, 48, 96, 48, 232, and 224, respectively. The pooling sizes of the four max pooling layers are 2, 2, 2, and 3, respectively.

a Sigmoid activation function and batch normalization. The kernel size for the conv1D layers is 10. Four max pooling layers are placed after the conv1D layers number two, three, five, and seven for downsampling the features. The final layer is a fully connected layer with a linear activation function to map the extracted features into desired target values. These hyperparameters were fixed for the remainder of this paper.

2.2. Input data preprocessing

As mentioned earlier, the ratio of spectral intensities between eFBGs at each sensing plane carries the strain information. Therefore, the input variables should not be normalized/standardized independently. We investigated two preprocessing transformations on the input variables, one-dimensional and multi-dimensional z-scaling [42]. In the first method, we applied the standard scaling technique by considering the input data as a one-dimensional vector. The data distribution after rescaling has a zero mean value, and its standard deviation is one. Figures 3(a) and (b) show the data distribution before and after rescaling. In the second method, we applied multi-dimensional standard scaling [42] by subtracting each wavelength component from its mean value and dividing them over the square root of the covariance matrix

$$Z = UD^{-1/2}U^{t}(X - \mu), \tag{1}$$

where U and D are the Eigenvectors and the Eigenvalues of the covariance matrix, X is the input data (sensor's spectra), and μ is the mean spectral intensity value at each wavelength component over the training dataset. With this approach, we achieved an approximately Gaussian density distribution (see figure 3(c)).

2.3. Output data preprocessing

This section examines the measured coordinates of each marker as a point cloud and evaluates the model's performance under various preprocessing transformations applied to these point clouds (for more details, refer to appendix). In the first method (M1), we translated each point cloud to the origin of the global frame by subtracting the mean coordinate values from the original coordinates. We calculated the radial distance of the points from the cloud's center in all markers and computed its mean value. We then divided the

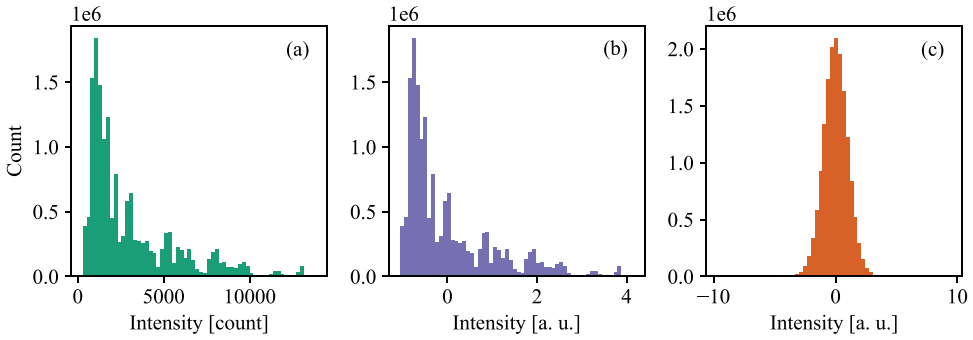

Figure 3. The histogram of the flat original input data (a), normalized input data using one-dimensional (b), and multi-dimensional z-scaling (c). The density distribution when using multi-dimensional z-scaling is approximately Gaussian, with skewness of $\sim$0.003 and kurtosis of ~ -1.2.

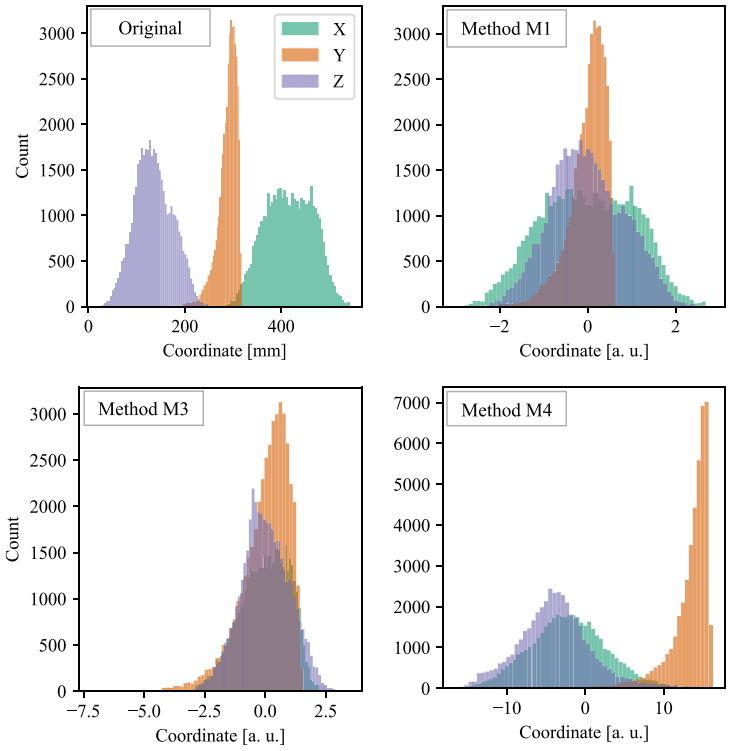

Figure 4. The histograms of the original *x*-, *y*-, and *z*-coordinates, their normalized values using M1 and M3, and the relative values using M4 for the point cloud number 10 are shown. The histogram of processed values using M2 is not depicted, as the distributions of *x*-, *y*-, and *z*-coordinates are similar to M1.

translated coordinates by this calculated mean radial distance. In this method, the spatial coordinate elements for each marker have different scaling from the original data, but the ratio between the markers' coordinates remains unchanged (see figure 4). In the second approach (M2), each already translated point cloud was rescaled based on the mean of the radial distance of its own points, and therefore, the scaling factor is different for each point cloud. For the third method (M3), we applied a three-dimensional standard scaling to each point cloud. This way, the transformed data is uncorrelated, and the density profiles are different from the original ones. Lastly, as the fourth method (M4), we used the relative coordinates between the markers instead of absolute coordinate values. Figure 4 shows how the distribution of the coordinate elements changes when applying different preprocessing transformations to the target values.

2.4. Siamese network

A further improvement of the network's performance is possible by guiding the feature-extracting part of the network in selecting relevant features for a given spectral sample [43, 44]. The Siamese network [36] is an

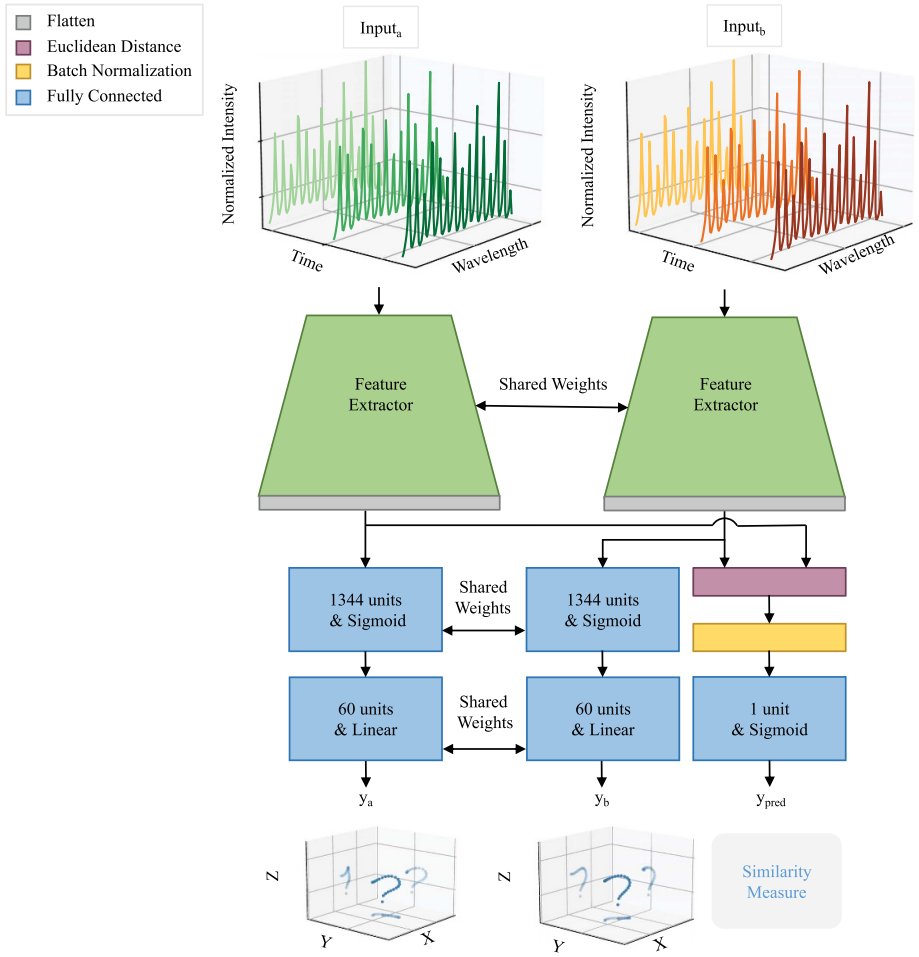

Figure 5. The architecture of the Siamese model. The feature extractor subnetworks have a similar architecture to what is shown in figure 2, but without the dropout and the fully connected layers.

architecture, designed for learning similarity metrics, which is well known from face recognition [45, 46] and handwritten forgery detection applications [36, 47]. A Siamese network usually takes two inputs, compares them in the feature space, and provides a similarity measure between the two feature vectors. Siamese architectures consist of two identical subnetworks (the feature extractors) with shared weights that are trained using paired samples corresponding to similar (genuine) or dissimilar (imposter) outputs/pairs. During training, the feature extractor subnetworks are forced to provide vectors close to each other when the inputs belong to the same group and are far away from each other if they are from different groups.

In our implementation, shown in figure 5, we used the same layer settings for feature extraction as in the architecture explained in section 2.1. In this setting, the Euclidean distance between the two feature vectors is batch normalized and passed through a single-unit fully connected layer with a Sigmoid activation function. The output of the Sigmoid activation function gives a value close to one for distant feature vectors and a value close to zero for close vectors. In parallel, the feature vectors are also passed into two fully connected layers. The first one has 1344 units followed by a Sigmoid activation function, and the second one is similar to the last layer of the previous architecture (see figure 2), in which, the preprocessed coordinates are calculated. We grouped the samples using the RMSE, which is the root-mean-square of the Euclidean distance between the corresponding shapes. First, we calculated the RMSE for all possible pairs in the training dataset. We then defined the 1st and the 25th percentiles in the RMSE's histogram as thresholds for labeling the samples. These limits were selected after studying various combinations. Samples were labeled as zero (genuine pairs) if the calculated RMSE between two samples was less than the lower limit and as one (imposter pairs) if it was within a 1% range around the upper limit.

The following loss function was used for this Siamese network:

$$\begin{aligned}\text{Loss} = \text{mean}\Big(&\alpha\left((1-y_{\text{true}})\,y_{\text{pred}}^2 + y_{\text{true}}\max(0, M-y_{\text{pred}})^2\right)\\&+(1-\alpha)\big(L(y_A-y_a)+L(y_B-y_b)\big)\Big),\end{aligned} \tag{2}$$

where,

$$L(a) = \begin{cases} 0.5\,\frac{a^2}{\delta} & |a| \leqslant \delta, \\ 0.5\,\delta + (|a| - \delta) & \text{otherwise.} \end{cases} \tag{3}$$

In these equations, α is a scalar coefficient to weight the contributions of the loss values, calculated from the three outputs. δ is a hyperparameter, defining the range for mean-absolute-error, and mean-square-error in L, a modified version of the Huber loss function [48]. M is the margin, y_{true} is the true label of paired samples, y_{pred} is the output of the right arm, y_A and y_B are the true relative coordinates, and y_a and y_b are the predicted relative coordinates in the left and the middle arms of the network (figure 5). The partial loss, calculated from y_{pred}, is the Contrastive loss [49]. Depending on whether the inputs are a genuine/imposter pair, the first or the second part of the Contrastive loss is applied to the output of the network's right arm. The two L loss functions are calculated from predicted relative coordinate values in the left and the middle arms of the network (figure 5). For a genuine pair, the network pushes y_{pred} towards zero, such that the first part of the Contrastive loss also gets closer to zero. If the inputs are an imposter pair, y_{pred} is forced to be larger than the value of M, such that the feature vectors stay separated in the feature space. Similar to sections 2.2 and 2.3, the training hyperparameters for this network, including the optimizer's parameters, M, α, and δ, were tuned using the Hyperband algorithm (refer to table 5 in appendix for the search criteria). We ran the hyperparameter search multiple times and selected the best-performing architecture for final training.

3. Results

Evaluating the performance of the configurations suggested by the Hyperband was done by calculating the shape evaluation metrics between the true and the predicted shapes. Shape evaluation metrics include the tip error (the Euclidean distance between the true and the predicted coordinate of the sensor's tip) and the RMSE. The best-performing architecture among the three hyperparameter search attempts for each normalization method was selected based on the median values of the shape evaluation metrics in the validation dataset.

Table 1 shows the error values for the two input normalization methods. The one-dimensional normalization method, which preserves the distribution profile of the input data, results in a median tip error and a median RMSE of 4.46 mm and 2.74 mm, respectively. In the multi-dimensional normalization approach, the median values are 13.38 mm and 8.11 mm, respectively, which are significantly higher compared to the one-dimensional normalization method. This might be due to the validation loss reaching its plateau quicker when the input data distribution is approximately Gaussian (shown in figure 6). The one-dimensional normalization method was therefore selected as the input data preprocessing step for the remainder of this paper.

The error statistics for shape evaluation metrics when using different preprocessing methods on the target data are shown in table 2. Among the three preprocessing approaches applied to the absolute coordinate values (M1, M2, and M3), the first method (M1) shows the lowest error values with a median tip error of 4.68 mm and a median RMSE of 2.83 mm. However, the network performs better in predicting relative coordinates (M4) compared to absolute values by resulting in the median value of 4.36 mm for the tip error and 2.78 mm for the RMSE. Therefore, the relative coordinate values were selected as the best-performing target data preprocessing. The last row in table 2 shows the shape evaluation parameters of the Siamese network when using the best-performing data preprocessing on both input and target data, one-dimensional normalization, and M4. As can be noted, there is a significant improvement in all error values, and the median tip error is reduced by almost 1.25 mm to 3.11 mm compared to M4 (more information on the significance test is provided in appendix). The median value of RMSE is also reduced to 1.98 mm compared to the M4 method, which is 2.78 mm.

The designed Siamese network uses the RMSprop as the optimizer with a learning rate of 1^{-4}, a momentum of 0.9, and a decay factor of 0.7. The loss function's hyperparameters, including M, α and δ, are 0.5, 0.7, and 2.2, respectively. A typical case of predicted shapes using the designed Siamese network is shown in figure 7. The error statistics based on the Euclidean distance between the true and the predicted shapes are shown in figure 8. It can be noticed that the median of the Euclidean distance between the true and the predicted shapes increases towards the sensor's end. This accumulative error may be due to inaccuracies in predicting the sensor's initial orientation that is caused by a small gap between the V-clamp and the first eFBG sensing plane (as shown in figure 1). The reader is referred to [35] for more details.

Table 1. Shape evaluation errors on the test dataset when the input data were preprocessed using one-dimensional and multi-dimensional normalization methods. IQR: interquartile, one-dim.: one-dimensional normalization, multi-dim.: multi-dimensional normalization.

	Tip error (mm)		RMSE (mm)	
Method	Median	IQR	Median	IQR
One-dim.	4.46	4.30	2.74	2.39
Multi-dim.	13.38	11.34	8.11	5.76

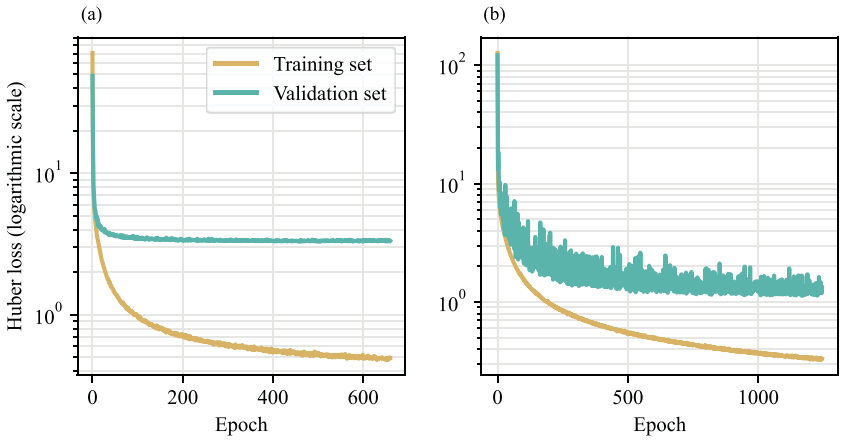

Figure 6. Training histories when the input data was preprocessed using (a) multi-dimensional and (b) one-dimensional normalization methods. The model was trained with early stopping conditions.

Table 2. Shape evaluation errors on the test dataset when the target data were processed using four different methods and when the network architecture was modified based on the Siamese design (indicated in bold). The model's output was first scaled back to absolute coordinates for each method, and then the error values were computed.

	Tip error (mm)		RMSE (mm)	
Method	Median	IQR	Median	IQR
M1	4.68	4.29	2.83	2.20
M2	6.73	5.46	3.97	2.69
M3	6.85	5.37	3.98	2.69
M4	4.36	4.46	2.78	2.56
Siamese	**3.11**	**3.38**	**1.98**	**1.97**

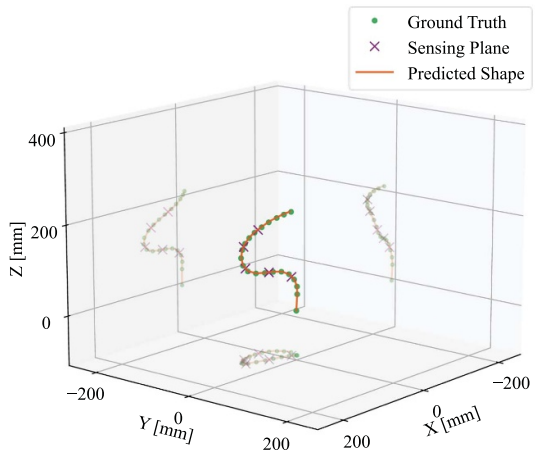

Figure 7. A typical case of predicted shapes using the designed Siamese model. The five sensing planes of the eFBG sensor are illustrated with × signs. The true shape (ground truth) of the sensor is shown with green circles, and the predicted shape using the deep learning model is shown with the orange solid line.

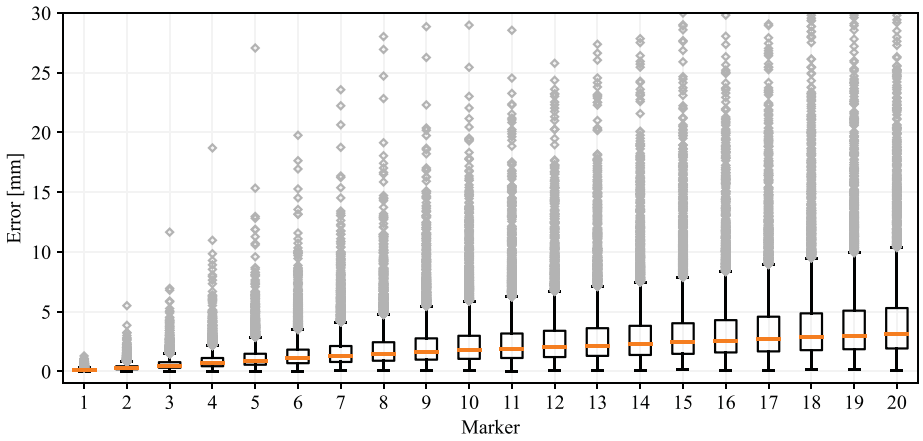

Figure 8. The error statistics are based on the Euclidean distance between the true markers' absolute coordinates and the predicted relative coordinates that are scaled back to the absolute values (the first marker is excluded). On each box, the median is indicated with a central mark, and the 25th and 75th percentiles are indicated with the bottom and top edges of the box, respectively. The whiskers show the minimum and maximum values within each group (outliers are not considered), and the outliers are marked with ⋄ symbols. The total number of outliers are ∼400, out of ∼5300 samples. Some outliers were excluded for better visualization of the median values.

4. Conclusion

In this work, we designed a deep learning-based model to extract the shape information of an eFBG sensor based on its full spectrum. We used the Hyperband algorithm to optimize the hyperparameters of our neural networks. We performed the hyperparameter tuning in two steps to avoid a large search space. First, the parameters related to the conv1D and the pooling layers were optimized. The best-performing architecture contains seven conv1D layers with Sigmoid activation function and four max pooling layers. In the second step, optimization and loss calculation hyperparameters were defined in the Hyperband search space to optimize the network when differently scaled input and output data were used. We showed that the model performs better when the input data is normalized with the one-dimensional z-scaling method and when relative coordinates instead of absolute values are used as the target data. Upgrading the selected architecture to the Siamese design significantly improved the shape prediction accuracy of a 30 cm long sensor, with a median tip error of 3.11 mm and a median RMSE of 1.98 mm in a curvature range of 1.4 m^{-1} to 35.3 m^{-1}. We achieved an improvement of almost 2.7 mm in the median value of the tip error, and 1.4 mm in the median value of the RMSE using the proposed model, compared to the previously designed network architecture in [35], which showed a median tip error of 5.8 mm and a median RMSE value of 3.4 mm. Compared to the MFD method [30, 34], our proposed model can accurately predict the sensor's shape with an order of magnitude lower tip error.

In future work, we will add temporal shape information to the input data to further improve the prediction accuracy. We also tend to continue investigating different architectural designs, including Siamese networks with triplet loss.

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: https://academictorrents.com/details/33ebcf714a480206fa9f76359e48a0355a974757.

Acknowledgment

We gratefully acknowledge the funding of this work by the Werner Siemens Foundation through the MIRACLE Project.

Appendix A. Hyperparameter optimization

This section presents the search criteria for all three hyperparameter optimizations performed in this work. Table 3 shows the search space settings for the first step of hyperparameter optimization, in which, the number of conv1D layers was set to seven. Each conv1D layer was followed by a Sigmoid activation function and batch normalization. For the second step of hyperparameter optimization, the settings of the conv1D

Table 3. Search criteria for the first step of hyperparameter optimization. The optimized kernel size, channel sizes, and pooling layer settings, resulting from this hyperparameter tuning step, are fixed in the next hyperparameter search step.

Hyperparameter	Search space
Dropout rate	min: 0, max: 0.3, step: 0.1
Optimizer	SGDW, AdamW
Learning rate	0.1, 0.01, 0.001, 0.0001
Weight decay	0.1, 0.01, 0.001, 0.0001, 0.00 001
Momentum	min: 0, max: 0.9, step: 0.1
Kernel size (similar for all conv1D layers)	min: 2, max: 10, step: 1
Channel size (different for each conv1D layer)	min: 8, max: 256, step: 8
Choice of max pooling (different after each conv1D layer)	true, false
Pooling size (different for each max pooling layer)	min: 2, max: 3, step: 1

Table 4. Search criteria for the second step of hyperparameter optimization. In this step, the hyperparameter search was performed three times for each data preprocessing approach.

Hyperparameter	Search space
Dropout rate	min: 0, max: 0.3, step: 0.1
Optimizer	SGDW, AdamW, RMSprop, Adadelta, Adamax
Learning rate	0.1, 0.01, 0.001, 0.0001
Weight decay	0.1, 0.01, 0.001, 0.0001, 0.00 001
Momentum	min: 0, max: 0.9, step: 0.1
Loss function	mean absolute error, mean squared error, mean squared logarithmic error, Huber loss, mean absolute percentage error, cosine similarity

Table 5. Hyperparameter search criteria for the Siamese network. α, δ, and M are the loss function's hyperparameters, defined in equations (2) and (3).

Hyperparameter	Search space
α	min: 0, max: 1, step: 0.1
δ	min: 0.1, max: 5, step: 0.1
M	min: 0.5, max: 1, step: 0.1
rho	min: 0.5, max: 0.9, step: 0.1
Momentum	min: 0.5, max: 0.9, step: 0.1

and the pooling layers were fixed. Table 4 shows the search space settings for this hyperparameter tuning. The search criteria for tuning the Siamese network's hyperparameters are presented in table 5.

Appendix B. Point cloud rescaling

This section explains the output data transformation approaches in more detail. Figure 9 shows a 3D shape example of the fiber sensor with 21 markers. For illustration purposes, we only focus on three highlighted markers (numbers 7, 12, and 17) and study their point cloud modifications as we apply M1–M4 rescaling methods. Figures 10 and 11 depict the scatter plots (point clouds) of measured coordinates of the selected markers in ∼5800 different shape examples. To compare the size of the point clouds after rescaling, an estimated sphere of each point cloud is depicted with a black mesh plot. Each sphere is centered at the mean coordinate of the original/rescaled points and has a radius of r. The radius r, in each point cloud, is the mean value of the calculated radial distance between the points and the cloud's center.

In method M1, the rescaling factor is the same for all spheres. Therefore, as can be seen in figures 10 (b_{1-3}), the relative size between the spheres is similar to their original versions (figures 10(a_{1-3})). In method M2, the rescaling factor for each point cloud is the average of calculated r values in that point cloud. Therefore, all point clouds have sphere mesh plots with a radius of one (figures 10(c_{1-3})). In the third method (M3), applying three-dimensional standard scaling to the point clouds makes the transformed data uncorrelated (figures 11(d_{1-3})). Using relative coordinates between the markers (M4), greatly changes the point clouds' appearance. As the markers are fixed on the sensor, the maximum relative distance between two neighboring markers is limited. It can be seen in figures 11(e_{1-3}) that the coordinate points, especially for the markers closer to the sensor's tip, are constrained in terms of volume, and therefore, better form a sphere.

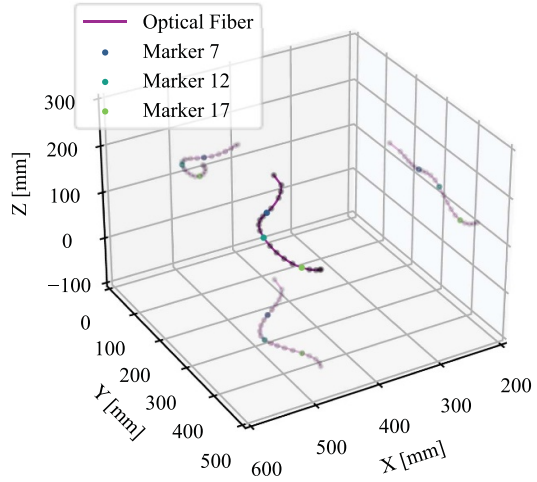

Figure 9. A 3D shape example of the fiber sensor with 21 attached reflective markers is depicted. The three highlighted markers, numbers 7, 12, and 17, are selected for scatter plotting. The remaining reflective markers are shown with gray circles. The coordinate system is rotated for better visualization.

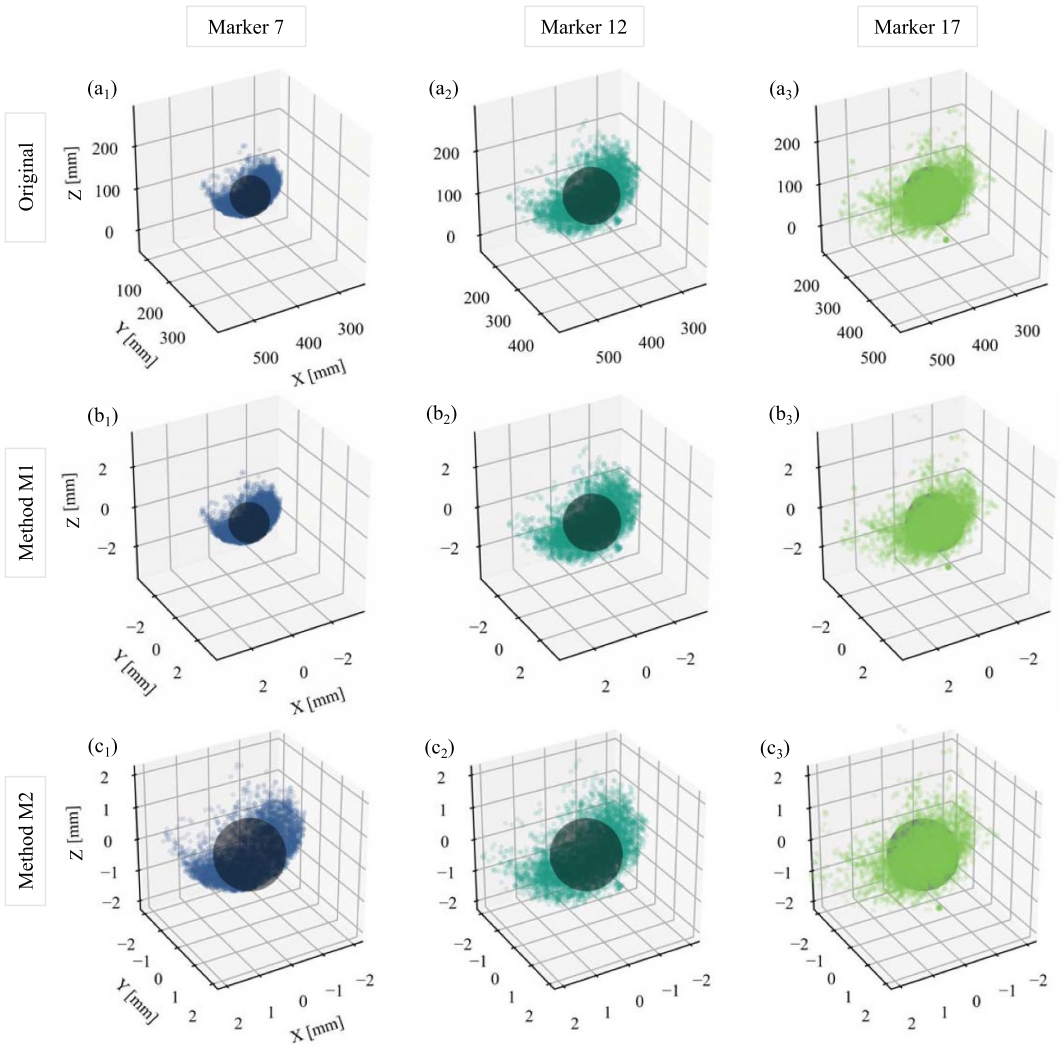

Figure 10. The scatter plot of measured coordinates in ∼5800 shape examples. The estimated sphere of each point cloud is shown with a black mesh plot. The original coordinate values of marker numbers 7 (a_1), 12 (a_2), and 17 (a_3). The rescaled point clouds using M1 method for marker numbers 7 (b_1), 12 (b_2), and 17 (b_3). The rescaled point clouds using M2 method for marker numbers 7 (c_1), 12 (c_2), and 17 (c_3).

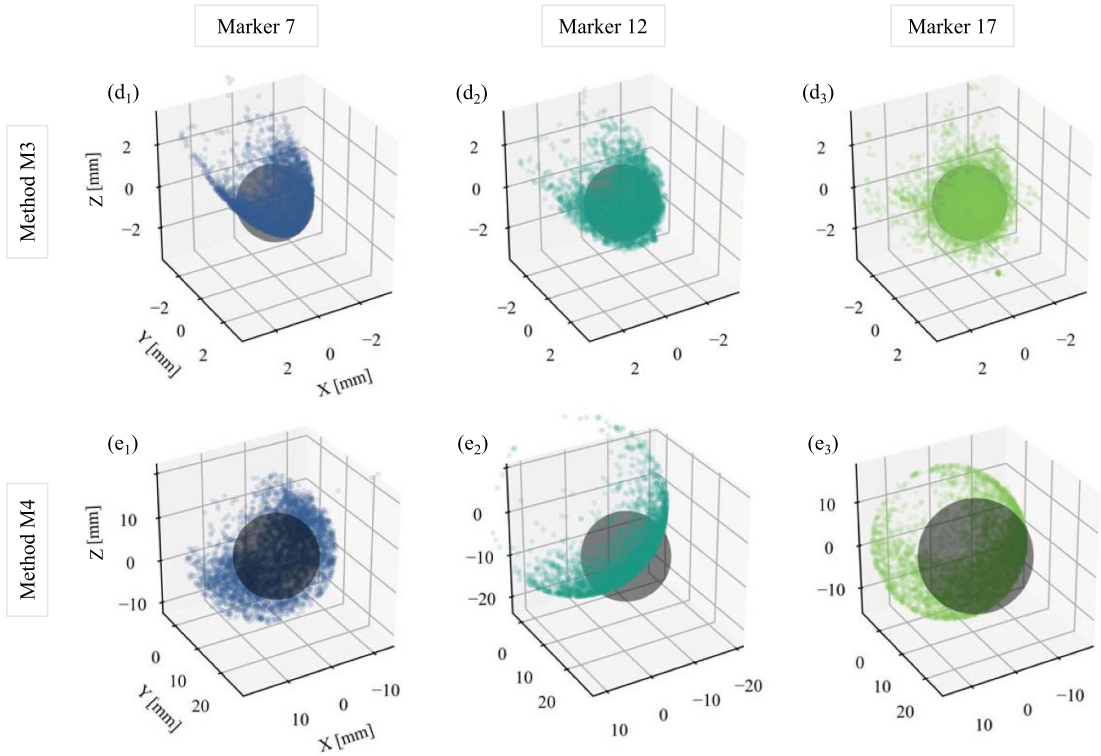

Figure 11. The rescaled point clouds using M3 method for marker numbers 7 (d_1), 12 (d_2), and 17 (d_3). The rescaled point clouds using M4 method for marker numbers 7 (e_1), 12 (e_2), and 17 (e_3).

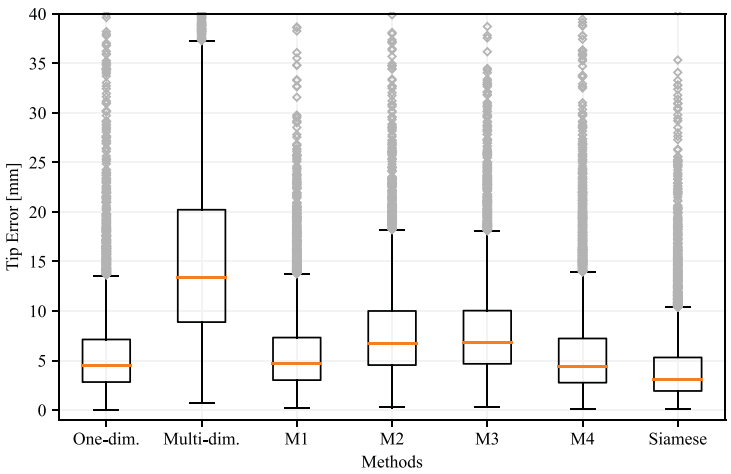

Figure 12. The tip error statistics of the designed networks when different data preprocessing methods and the Siamese architecture were employed. On each box, the median is indicated with a central mark, and the 25th and 75th percentiles are indicated with the bottom and top edges of the box, respectively. The whiskers show the minimum and maximum values within each group (outliers are not considered), and the outliers are marked with ◇ symbols. Some outliers were excluded for better visualization of the median values.

Appendix C. Significance test

In this section, the significance test results, comparing the evaluated models, are presented. Figure 12 shows the tip error box plots of all seven methods evaluated in this paper. As can be clearly noticed, the Siamese network has the least median tip error. We performed Tukey's HSD pairwise group comparisons on the seven methods. The Siamese method shows p-values close to zero compared to the other seven methods, proving that the shape prediction's improvement is statistically significant.

ORCID iDs

Samaneh Manavi Roodsari https://orcid.org/0000-0002-0633-3573
Sara Freund https://orcid.org/0000-0003-2053-2568
Azhar Zam https://orcid.org/0000-0001-7789-5680
Georg Rauter https://orcid.org/0000-0001-9089-8181
Philippe C Cattin https://orcid.org/0000-0001-8785-2713

References

[1] Burgner-Kahrs J, Rucker D C and Choset H 2015 Continuum robots for medical applications: a survey *IEEE Trans. Robot.* **31** 1261–80
[2] Vitiello V, Lee S-L, Cundy T P and Yang G-Z 2012 Emerging robotic platforms for minimally invasive surgery *IEEE Rev. Biomed. Eng.* **6** 111–26
[3] Dogangil G, Davies B L and Rodriguez y Baena F 2010 A review of medical robotics for minimally invasive soft tissue surgery *Proc. Inst. Mech. Eng.* H **224** 653–79
[4] van de Berg N J, van Gerwen D J, Dankelman J and van den Dobbelsteen J J 2014 Design choices in needle steering-a review *IEEE/ASME Trans. Mech.* **20** 2172–83
[5] Wang X and Meng M Q-H 2012 Robotics for natural orifice transluminal endoscopic surgery: a review *J. Robot.* **2012** 1–9
[6] Yeung B P M and Chiu P W Y 2016 Application of robotics in gastrointestinal endoscopy: a review *World J. Gastroenterol.* **22** 1811–25
[7] Payne C J and Yang G-Z 2014 Hand-held medical robots *Ann. Biomed. Eng.* **42** 1594–605
[8] Patel N, Seneci C, Yang G-Z, Darzi A and Teare J 2014 Flexible platforms for natural orifice transluminal and endoluminal surgery *Endosc. Int. Open* **2** E117–23
[9] Webster R J III and Jones B A 2010 Design and kinematic modeling of constant curvature continuum robots: a review *Int. J. Robot. Res.* **29** 1661–83
[10] Xu K and Simaan N 2010 Analytic formulation for kinematics, statics and shape restoration of multibackbone continuum robots via elliptic integrals *J. Mech. Robot.* **2** 011006
[11] Optical measurement/optical tracking - NDI 2022 (available at: www.ndigital.com)
[12] Surgical navigation/optical tracking - kick 2022 (available at: www.brainlab.com)
[13] Wagner M, Schafer S, Strother C and Mistretta C 2016 4D interventional device reconstruction from biplane fluoroscopy *Med. Phys.* **43** 1324–34
[14] Glossop N 2012 Localization and tracking technologies for medical robotics *Medical Robotics* (Woodhead Publishing Series in Biomaterials) 41–58
[15] Ryu S C and Dupont P E 2014 FBG-based shape sensing tubes for continuum robots *IEEE Int. Conf. on Robotics and Automation (ICRA)* 3531–7
[16] Manavi Roodsari S, Witthauer L, Iafolla L, Zam A, Rauter G and Cattin P C 2018 Temperature-compensated FBG-based 3D shape sensor using single-mode fibers *Advanced Photonics (BGPP, IPR, NP, NOMA, Sensors, Networks, SPPCom, SOF)* (Optica Publishing Group) p JTu6C.1
[17] Manavi Roodsari S, Freund S, Zam A, Rauter G and Cattin P C 2022 Fabrication and characterization of a flexible FBG-based shape sensor using single-mode fibers *IEEE Trans. Biomed. Eng.* **69** 2488–98
[18] Roesthuis R J, Janssen S and Misra S 2013 On using an array of fiber Bragg grating sensors for closed-loop control of flexible minimally invasive surgical instruments *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems* pp 2545–51
[19] Jäckle S, Eixmann T, Schulz-Hildebrandt H, Hüttmann G and Pätz T 2019 Fiber optical shape sensing of flexible instruments for endovascular navigation *Int. J. Comput. Assist. Radiol. Surg.* **14** 2137–45
[20] Waltermann C, Koch J, Angelmahr M, Burgmeier J, Thiel M and Schade W 2014 Fiber-Optical 3D Shape Sensing *Planar Waveguides and Other Confined Geometries: Theory, Technology, Production and Novel Applications* vol 189 (New York: Springer) pp 227–50
[21] Yi J, Zhu X, Zhang H, Shen L and Qiao X 2012 Spatial shape reconstruction using orthogonal fiber Bragg grating sensor array *Mechatronics* **22** 679–87
[22] 3D shape sensing 2022 (available at: www.sensuron.com)
[23] Fiber optic shape sensing | the shape sensing company 2022 (available at: https://shapesensing.com)
[24] Waltermann C, Baumann A L, Bethmann K, Doering A, Koch J, Angelmahr M and Schade W 2015 Femtosecond laser processing of evanescence field coupled waveguides in single mode glass fibers for optical 3d shape sensing and navigation *Proc. SPIE Fiber Optic Sensors and Applications XII* vol 9480 pp 202–9
[25] Thomson R R, Bookey H T, Psaila N D, Fender A, Campbell S, Macpherson W N, Barton J S, Reid D T and Kar A K 2007 Ultrafast-laser inscription of a three dimensional fan-out device for multicore fiber coupling applications *Opt. Express* **15** 11691–7
[26] Bao W, Rong Q, Chen F and Qiao X 2018 All-fiber 3D vector displacement (bending) sensor based on an eccentric FBG *Opt. Express* **26** 8619–27
[27] Rong Q, Guo T, Bao W, Shao Z, Peng G-D and Qiao X 2017 Highly sensitive fiber-optic accelerometer by grating inscription in specific core dip fiber *Sci. Rep.* **7** 11856
[28] Feng D, Qiao X and Albert J 2016 Off-axis ultraviolet-written fiber Bragg gratings for directional bending measurements *Opt. Lett.* **41** 1201–4
[29] Zhang L, Qiao X, Liu Q, Shao M, Jiang Y and Huang D 2018 Off-axis ultraviolet-written thin-core fiber Bragg grating for directional bending measurements *Opt. Commun.* **410** 197–201
[30] Waltermann C, Bethmann K, Doering A, Jiang Y, Baumann A L, Angelmahr M and Schade W 2018 Multiple off-axis fiber Bragg gratings for 3D shape sensing *Appl. Opt.* **57** 8125–33
[31] Drexler P and Fiala P 2012 Optical fiber Birefringence effects–sources, utilization and methods of suppression *Recent Progress in Optical Fiber Research* (Rijeka: InTechOpen) (https://doi.org/10.5772/27517)
[32] Thomas J, Jovanovic N, Becker R G, Marshall G D, Withford M J, Tünnermann A, Nolte S and Steel M 2011 Cladding mode coupling in highly localized fiber Bragg gratings: modal properties and transmission spectra *Opt. Express* **19** 325–41

[33] Thomas J U, Jovanovic N, Krämer R G, Marshall G D, Withford M J, Tünnermann A, Nolte S and Steel M J 2012 Cladding mode coupling in highly localized fiber Bragg gratings II: complete vectorial analysis *Opt. Express* **20** 21434–49
[34] Manavi Roodsari S, Freund S, Angelmahr M, Rauter G, Zam A, Schade W and Cattin P C 2022 The secret role of undesired physical effects in eccentric FBGs (arXiv:2210.16316v2 [cs.LG])
[35] Manavi Roodsari S, Renna T, Horvath A, Freund S, Zam A, Rauter G, Schade W and Cattin P C 2021 Using supervised deep-learning to model edge-FBG shape sensors: a feasibility study *Proc. SPIE, Optical Sensors* vol 11772 pp 79–88
[36] Bromley J, Guyon I, LeCun Y, Säckinger E and Shah R 1993 Signature verification using a 'siamese' time delay neural network *Advances in Neural Information Processing Systems* vol 6 (available at: https://proceedings.neurips.cc/paper_files/paper/1993/file/288cc0ff022877bd3df94bc9360b9c5d-Paper.pdf)
[37] Jamieson K and Talwalkar A 2016 Non-stochastic best arm identification and hyperparameter optimization *Proc. of Int. Conf. on Artificial Intelligence and Statistics* vol 51 (PMLR) pp 240–8 (available at: https://proceedings.mlr.press/v51/jamieson16.html)
[38] Li L, Jamieson K, DeSalvo G, Rostamizadeh A and Talwalkar A 2017 Hyperband: a novel bandit-based approach to hyperparameter optimization *J. Mach. Learn. Res.* **18** 6765–816
[39] Bishop C M 1995 *Neural Networks for Pattern Recognition* Advanced Texts in Econometrics (Oxford: Clarendon Press)
[40] Manavi Roodsari S *et al* 2023 The raw data of eFBG fiber sensor and the Jupyter notebook used for loading and splitting the data (available at: https://academictorrents.com/details/33ebcf714a480206fa9f76359e48a0355a974757)
[41] O'Malley T *et al* 2019 Kerastuner (available at: https://github.com/keras-team/keras-tuner)
[42] Kessy A, Lewin A and Strimmer K 2018 Optimal whitening and decorrelation *Am. Stat.* **72** 309–14
[43] Wiatowski T and Bölcskei H 2017 A mathematical theory of deep convolutional neural networks for feature extraction *IEEE Trans. Inf. Theory* **64** 1845–66
[44] Mallat S 2012 Group invariant scattering *Commun. Pure Appl. Math.* **65** 1331–98
[45] Chopra S, Hadsell R and LeCun Y 2005 Learning a similarity metric discriminatively, with application to face verification *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR)* vol 1 pp 539–46
[46] Koch G, Zemel R and Salakhutdinov R 2015 Siamese neural networks for one-shot image recognition *ICML Deep Learning Workshop* vol 2
[47] Dey S, Dutta A, Toledo J I, Ghosh S K, Lladós J and Pal U 2017 Signet: convolutional siamese network for writer independent offline signature verification (arXiv:1707.02131 [cs.CV])
[48] Huber P J 1992 Robust estimation of a location parameter *Breakthroughs in Statistics: Methodology and Distribution* (New York: Springer) 492–518
[49] Hadsell R, Chopra S and LeCun Y 2006 Dimensionality reduction by learning an invariant mapping *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR)* vol 2 pp 1735–42