

## SOFT ROBOTS

## Soft optoelectronic sensory foams with proprioception

I. M. Van Meerbeek<sup>1</sup>, C. M. De Sa<sup>2</sup>, R. F. Shepherd<sup>1,3\*</sup>

In a step toward soft robot proprioception, and therefore better control, this paper presents an internally illuminated elastomer foam that has been trained to detect its own deformation through machine learning techniques. Optical fibers transmitted light into the foam and simultaneously received diffuse waves from internal reflection. The diffuse reflected light was interpreted by machine learning techniques to predict whether the foam was twisted clockwise, twisted counterclockwise, bent up, or bent down. Machine learning techniques were also used to predict the magnitude of the deformation type. On new data points, the model predicted the type of deformation with 100% accuracy and the magnitude of the deformation with a mean absolute error of 0.06°. This capability may impart soft robots with more complete proprioception, enabling them to be reliably controlled and responsive to external stimuli.

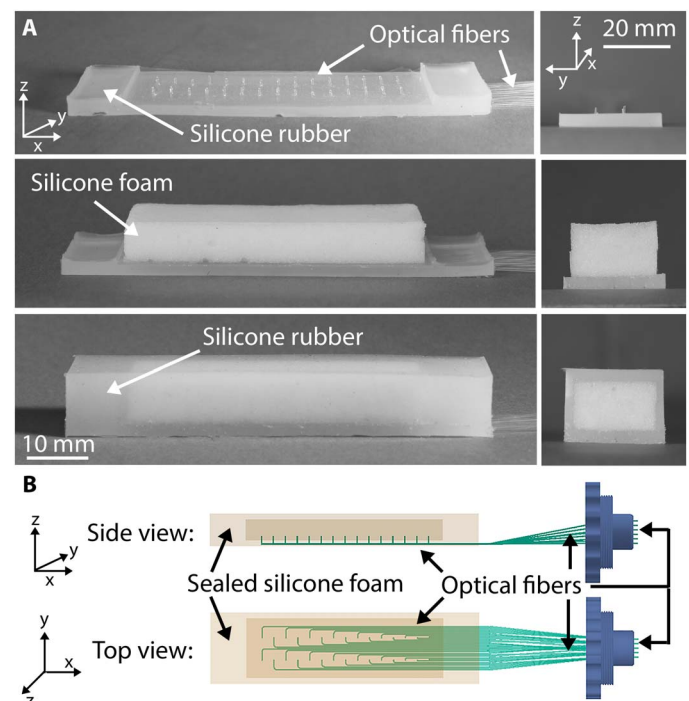
## INTRODUCTION

Since its inception, the field of soft robotics has advanced from one-degree-of-freedom contractile actuators with open-loop control [i.e., McKibben artificial muscles (1, 2)] to active three-degree-of-freedom mechanisms (3–7), devices with closed-loop control (8–10), and high-force actuators (11, 12). Contemporary elastomeric machines can also have both exteroception and proprioception through embedded strain and pressure sensors (13–17), enabling them to sense and respond to external forces (18). As elastomeric machines continue to grow in complexity and as roboticists push the boundary of soft robot functionality, more sophisticated sensing will become necessary.

For a soft robot to robustly interact with its environment, it must know its current shape in three dimensions (3D). To know its own configuration, an inherently compliant system must be able to sense deformation—whether it is self-induced through actuation or externally inflicted. The most commonly used sensors in soft robots are either surface mounted for pressure and touch detection (14, 17, 19, 20) or embedded along neutral bending axes to measure the global curvature of a robot limb (8, 21–23). These types of sensors are typically integrated to measure a specific type of deformation (e.g., pressure at a certain point and bending along a certain axis), which limits the information that they can give about a robot's configuration. To fully know a soft robot's shape, we may need to fabricate sensors that can detect arbitrary deformations; however, it may suffice to pattern high densities of currently available sensors and either derive a complex analytical model or apply machine learning (ML) techniques. Such an approach has been used on sensor systems to fabricate devices such as a gesture recognition device, a pressure sensor, and a robotic skin (24–27). In a step toward soft actuator proprioception, we present an elastomeric foam that can sense macroscopic deformation via embedded optical waveguides and the use of ML and statistical techniques to interpret transmitted light intensities.

Here, we present an elastomeric foam sensor system that we have trained to sense when it is being bent and twisted. To achieve this goal, we embedded an array of optical fiber terminals into the base layer of an elastomeric foam (Fig. 1). The fibers served to illuminate the foam and to detect diffuse reflected light. We bent and twisted the foam to known angles and gathered the intensity of the diffuse reflected light leaving

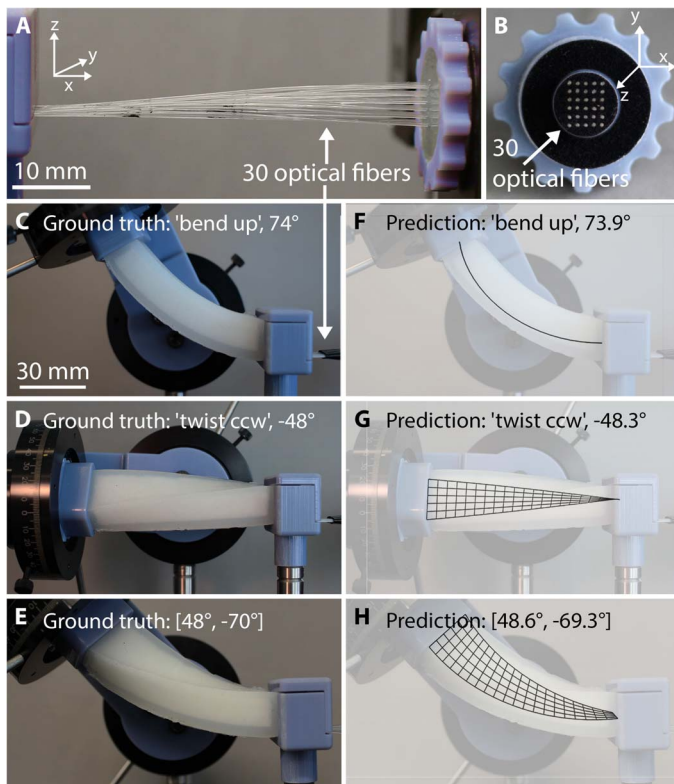
each fiber. To produce models that predict the foam's deformation state from the internally reflected light, we applied ML techniques to the data (Fig. 2 and movie S1). We chose to use ML instead of deriving a theoretical model, because doing the latter would have been very difficult given the large number of independent variables, many of which would have been difficult to accurately measure. Those independent variables include foam porosity, foam geometry, strut geometry, optical fiber placement, optical fiber terminal orientation, refractive index of the silicone, loss of the optical fibers, and absorption of the silicone. Diffusing wave spectroscopy (DWS) in cellular and colloidal substances has been used previously to gather information about microstructural statistics (28); however, this technique does not yield macroscopic shape specificity and has not been applied to robotics.



**Fig. 1. Foam assembly design.** (A) Left: Foam and optical fiber assembly in three stages of fabrication. Right: Cross section of foam and optical fiber assembly in three stages of fabrication. (B) Diagram of foam and optical fiber assembly.

<sup>1</sup>Department of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY 14853, USA. <sup>2</sup>Department of Computer Science, Cornell University, Ithaca, NY 14853, USA. <sup>3</sup>Department of Materials Science and Engineering, Cornell University, Ithaca, NY 14853, USA.

\*Corresponding author. Email: rfs247@cornell.edu



**Fig. 2. Sensor functionality.** (A and B) Optical fiber terminals from which light intensity is read. (C to E) Real images of deformed foam and optical fiber assembly. (F to H) Real images of deformed foam and optical fiber assembly overlaid with computer reconstruction of the assembly's state.

We combined this platform of DWS with ML to create a soft robotic sensor that can sense whether it is being bent, twisted, or both and to what degree(s).

To detect sensor deformation, we selected and evaluated two distinct approaches. The first approach used single-output classification to detect whether the sensor was being bent or twisted, followed by single-output regression to predict the magnitude. This approach allowed us to detect one deformation mode at a time. The second approach enabled us to detect bending and twisting simultaneously by using multi-output regression. To model the foam's state for the first approach, we defined two variables: deformation mode and angle. Deformation mode is a categorical variable that can hold one of the following four values: bend positive, bend negative, twist positive, or twist negative. Angle is a real-valued number corresponding to the magnitude of the bend or twist experienced by the foam. By using the values of deformation mode as training data labels, we trained a single-output categorical model to predict the type of deformation. Then, by using the values of angle as training data labels, we trained four single-output regression models (one for each deformation mode) to predict the magnitude of the deformation after the deformation had been categorized. We compared three classifiers [k-nearest neighbors (kNN), support vector machines (SVMs), and decision trees] and six regression models [kNN, SVMs, decision trees, Gaussian processes (GPs), linear models, and multilayer perceptrons (MLPs; also known as neural networks)]. The best classifiers had a test error rate of 0, and the best regression models had a test mean absolute error of 0.06°. For the second, multi-output approach, we modeled the foam's

**Table 1. Classifier model error rate.** Error rates of the classification models.

| kNN | SVMs | Tree |
|-----|------|------|
| 0   | 0    | 0.05 |

**Table 2. Single-output regression model errors.** Mean absolute errors for each deformation mode. CW, clockwise; CCW, counterclockwise.

|             | kNN         | GPs         | MLPs        | SVMs        | Tree        | Linear      |
|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| Bend up     | 0.07        | 1.59        | 1.74        | 3.10        | 5.02        | 6.50        |
| Bend down   | 0.08        | 1.88        | 2.72        | 2.92        | 5.25        | 8.92        |
| Twist CW    | 0           | 1.81        | 2.36        | 1.98        | 4.50        | 5.87        |
| Twist CCW   | 0.07        | 2.05        | 2.39        | 3.27        | 4.43        | 5.92        |
| <b>Mean</b> | <b>0.06</b> | <b>1.83</b> | <b>2.30</b> | <b>2.82</b> | <b>4.80</b> | <b>6.80</b> |

**Table 3. Multi-output regression model errors.** Mean absolute errors.

| kNN  | MLPs | Linear |
|------|------|--------|
| 0.01 | 1.43 | 13.9   |

state as a 2D vector of real-valued numbers representing the bend and twist angles experienced by the foam. With this label format, we trained a multi-output regression model to predict the bend and twist angles simultaneously. We compared three multi-output regression models—kNN, linear models, and MLPs—and found that the best model had a test mean absolute error of 0.01°.

## RESULTS

### Model performance on test data

We trained each single-output classifier on a training dataset of 2020 observations. We evaluated their performance on a set of 290 unseen observations. In both datasets, the range of bend angles was  $-80^\circ$  to  $90^\circ$  and the range of twist angles was  $-82^\circ$  to  $90^\circ$ . These bounds represent the physical limits of our testing apparatus. Of the classifiers tried, the kNN and SVM models performed best with a classification test error rate of 0. To predict the magnitude of the deformation mode, we partitioned the training dataset by deformation mode and trained one regression model for each partition (four total models). We repeated this process for each of the six regression models that we wanted to compare. The kNN regression model had the lowest mean absolute error of 0.06°. Tables 1 and 2 display the full results of these evaluations. We qualitatively demonstrated one of the composite model's performances by deforming the sensor in real time and displaying a geometric reconstruction of the foam's deformation state based on the model's predictions (movie S1). The prediction models used for this demonstration were the kNN classifier and the GP regression model.

The multi-output regression models were trained on a dataset of 956 observations. To evaluate their performance, we gathered a test dataset of 239 observations, and of the models tried, kNN performed best again, with a mean absolute error of 0.01° on the test dataset. Table 3 displays the full results for these trials. In machine learning, model parameters are those whose values are set during training (i.e., learning). Some examples are the slope and intercept for a linear model or the hyperplane and margin size for SVMs. A hyperparameter, by contrast, is a parameter whose value must be set before training. Hyperparameters help define the structure of the model being used. An example is the number of hidden layers in an MLP (i.e., neural net). For both single-output and multi-output models that had hyperparameters, we optimized them by using random search (29). All reported values come from the best hyperparameter sets that we found. Table S1 displays the hyperparameters that we used for each model.

**Cross-validation**

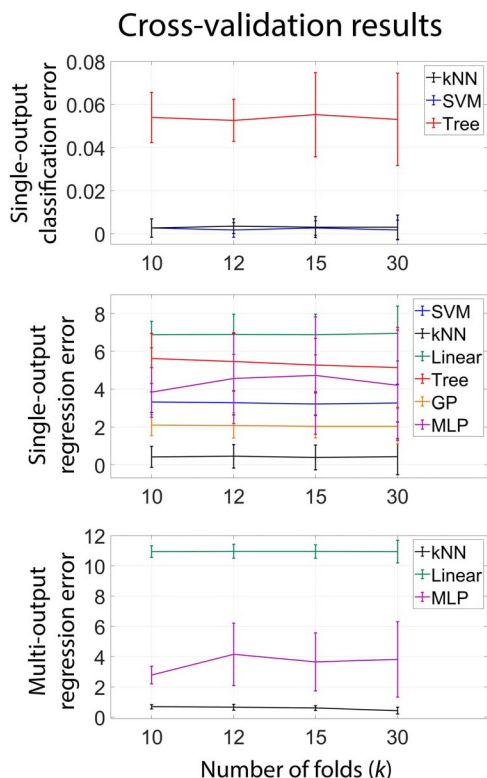
To assess how well our models would perform on new data, we performed nonexhaustive, *k*-fold cross-validation (30). Cross-validation is a method used to estimate model error on unobserved data by splitting the available data into subsets for training and subsets for evaluation. In *k*-fold cross-validation, the available data were randomly partitioned into *k* evenly sized subsets. Next, we reserved one subset for evaluation and trained a model by using the union of the remaining *k* - 1 subsets. We repeated this process for each subset—*k* times—such that *k* models were trained and evaluated. The errors of the *k* models on their corresponding validation sets were estimates of the test error of a model trained on the entire dataset. The variance between the *k* models indicates how much model error varied

with the training data. The results of the *k*-fold cross-validation are displayed in Fig. 3. We found that the models had low error, indicating a likelihood of low test error. Most of the models also had relatively little variance, suggesting that they did not depend heavily on the training data used. With this knowledge, we created our final models using all the gathered data as training data.

**Training data size**

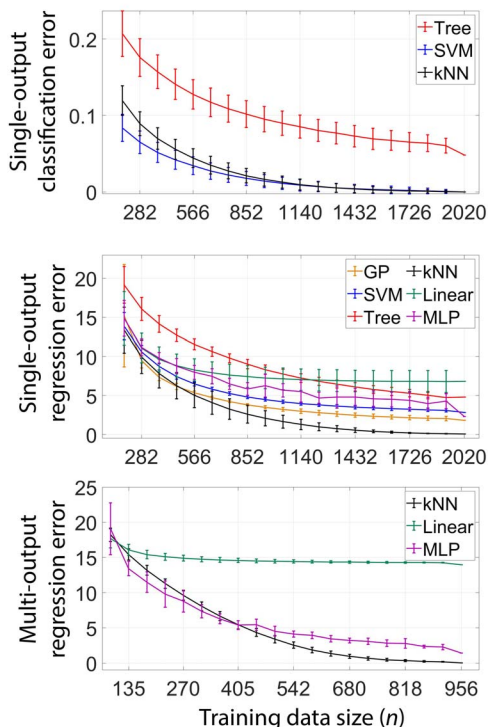
To determine how many training observations (*n*) were required to obtain useful models, we took increasingly smaller subsets of the training data and generated new models based on those smaller training datasets. For single-output prediction, an exhaustive search of all possible training data subsets would have required  $\sum_{i=1}^{2020} \binom{2020}{i} = \sum_{i=1}^{2020} \frac{2020!}{i!(2020-i)!} \gg 10^{307}$  trials for each prediction model (three classifiers and six regression models). We did not have the computational capacity to do the exhaustive search; therefore, for each model, we performed 300 trials for 19 different subset sizes each, for a total of 9 models × 300 trials × 19 subset sizes = 51,300 trials, which was the number of trials that our machine could process in 10 hours (e.g., overnight). The MLP trials took about 20 times longer; therefore, we performed only 15 trials for each subset size. We evaluated each trial’s model on the test dataset (290 data points). We performed a similar process for the multi-output prediction models.

As is typical, we found that, as the number of training observations increased, model performance improved (Fig. 4). The kNN



**Fig. 3. Results from *k*-fold cross-validation.** Error bars represent SD across the *k* models.

**Model performance on test data vs. training data size**



**Fig. 4. Effect of training data size.** Classification and regression performance on test data as a function of training data size. Each plot point represents the mean across random trials, and the error bars represent SD across 300 trials (15 for MLPs). Classification error is 0-1 loss.

classification error rate remained below 0.01 for models trained on datasets as small as 56% of the original training dataset size, and the single-output kNN regression mean absolute error remained below  $1.0^\circ$  for models trained on sets as small as 62% of the original training set size. The multi-output kNN regression mean absolute error also remained below  $1.0^\circ$  for training datasets as small as 75% of the original set. For most model types, the error appears to be approaching a plateau at the maximum training data size that we used, suggesting that the performance of our largest training dataset may be slightly, but not greatly, improved by collecting more data.

### Feature set size

To determine the relationship between model performance and optical fiber detector density, we removed randomly generated subsets of features (i.e., fiber intensity data) from the training and test data. We trained and evaluated new models by using the modified training and test data. The complete, unmodified training data had a feature set size ( $d$ ) of 30 (for the 30 fibers). To exhaustively search all possible feature subsets, we would have needed to generate and test  $\sum_{i=1}^{30} \binom{30}{i} = \sum_{i=1}^{30} \frac{30!}{i!(30-i)!} \approx 1.07 \times 10^9$  models for each of the models that we compared. We did not have the computational power for these many trials; therefore, for each feature set size, we randomly selected a quantity of subsets equal to the smaller of 300 and the maximum number of possible subsets for that feature set size ( $d$ -choose- $f$  where  $f$  was the size of the feature subset). Again, we chose this number of trials based on a

computation time of 10 hours, and the number of MLPs tested for each feature set size was reduced to 11 because of its greater runtime.

Figure 5 displays the full results for this experiment. We found that the single-output kNN classification error remained below 0.1 for feature set sizes as small as 10 and that the kNN regression model error remained below  $1.0^\circ$  for feature set sizes as small as 12 fibers. The multi-output kNN regression error remained below  $1.0^\circ$  for models with as few as nine fibers (i.e., features). These results suggest that our system could be redesigned to have as little as a third of the reported fiber density, which could be useful when designing a full soft robot embedded with this sensing system.

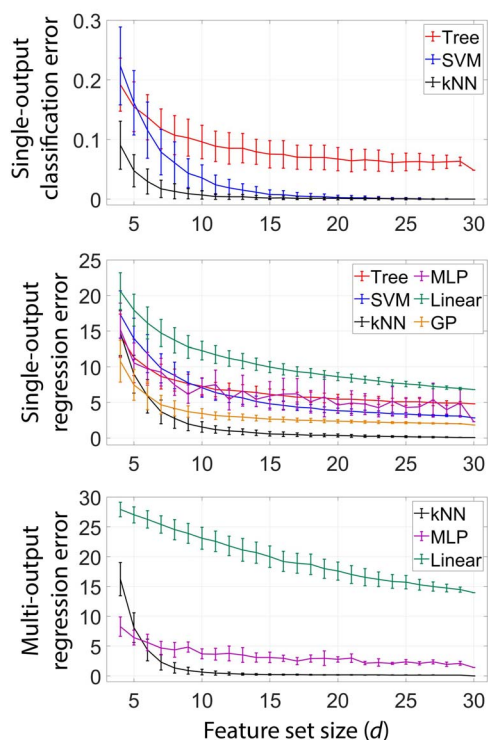
To determine whether certain fibers affected model performance more than others, we conducted another experiment in which we removed fibers and then trained and tested new models. For this experiment, however, instead of randomly removing subsets of fibers, we searched for the fiber that, when removed, produced a model with the highest test error and removed it from the data. We repeated this process until four fibers remained. Figure S1 shows these results plotted with the results from the randomly removed feature subsets for each model. We found that greedily removing fibers from the data produced slightly better models than randomly removing fibers, suggesting that some fibers affect error slightly more than others.

### DISCUSSION

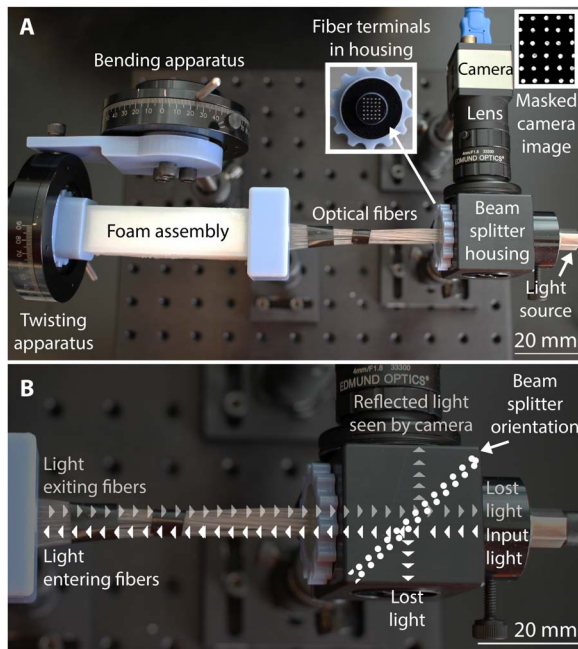
In general, ML model error can have three main causes: (i) The training data may not fully represent the unobserved data, (ii) the data may be noisy, and (iii) the model assumptions may be incorrect (e.g., assuming that the data are linear when they are not). The cross-validation results displayed in Fig. 3 show that our models had relatively low variance, indicating that the training data may represent the full space well. The mean signal-to-noise ratio of all fibers (signal mean divided by the SD of the noise) is 185, and when we propagated the signal noise through our models, we found little to no change in model prediction. Given the low cross-validation variance and limited effect of noise on prediction, the main contributor to model error in our experiments may be incorrect model assumptions (i.e., model bias). If model bias is the main contributor to prediction error, then kNN's lower average error across all trials may be due to it having the best model assumption of the models that we compared (i.e., similar inputs have similar outputs). In addition, kNN has been shown to have low model bias. Cover and Hart (31) showed that, as the training data size approaches infinity, for  $k = 1$ , kNN error is no more than twice the error of the best possible classifier.

kNN's lower error may make it the most effective model for this application; however, there are other factors to consider. The cross-validation results show that some models had lower variance than others. Specifically, the kNN, GP, and SVM test errors varied little between training datasets, while the MLPs and Tree models in particular showed much more variation. If the training data are limited for some reason, then one may want to pick the models with lower variance. One may also consider the evaluation time. Table S2 shows the mean time to evaluate one observation for each model. Although kNN models can often have slow evaluation times when the training set is large, since the training sets in this research remained small, the evaluation times were the same order of magnitude as most of the other models. Given that kNN showed desirable traits regarding error, variance, and evaluation speed, it stands out as one of the most useful models in this application. For a robotics system that makes decisions based partly on prediction

### Model performance on test data vs. feature set size



**Fig. 5. Effect of feature set size.** Classification and regression performance on test data as a function of feature set size. Each plot point represents the mean across random trials, and the error bars represent SD across 300 trials (11 for MLPs).

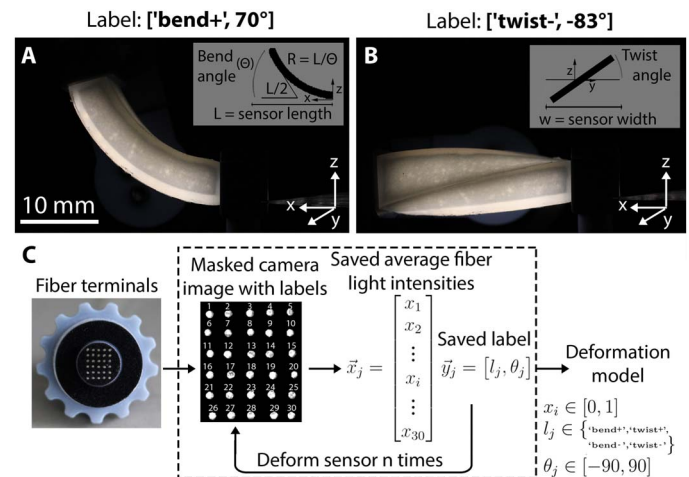


**Fig. 6. Experimental setup.** (A) Bird's eye view of experimental setup. (B) Diagram illustrating how each fiber serves as an illuminator and light detector via a beam splitter.

confidences, however, the GP model could be the most useful because it outputs predictions and confidence values for each prediction.

Several studies examining human proprioceptive capability through limb-matching tasks have found that wrist, finger, and elbow joint angle absolute errors lie between  $1^\circ$  and  $12^\circ$  (32–36). In particular, proprioception of the proximal interphalangeal joint angle has an absolute error between  $4^\circ$  and  $9^\circ$  (36). These results suggest that this level of error in proprioception is acceptable for tasks such as writing and reaching for objects. Given that the human index finger is on average 82 mm in length (37, 38) and that our sensor is 80 mm in length, we can loosely compare the performance of our sensor to that of the proximal interphalangeal joint on the human index finger. Scientists have shown that the proximal interphalangeal joint is located at about the midpoint of the finger (38); therefore, a joint angle error of  $4^\circ$  to  $9^\circ$  corresponds to a fingertip position error of 3 to 6 mm. Given the geometry of our sensor's measured bending angle (Fig. 6), the mean bend error obtained by, for example, the GP models ( $1.74^\circ$ ) corresponds to an error of about 2 mm in the position of the sensor's movable end. The kNN regression model has a smaller bend error, which corresponds to an even greater accuracy. With this comparison, we believe that our foam sensor system has the potential to greatly improve soft robotic control.

To apply this system to a soft robot, one would need to design the integration of optical fibers into the soft actuators. They would also need to integrate the illumination and detection devices. We used a large illuminator and a camera; however, the illuminator could be replaced with light-emitting diodes, and the camera could be replaced with photodiodes. The beam splitter setup could be miniaturized or removed; removing the beam splitter would require the number of embedded fibers to be doubled. We chose not to do these integrations because we wanted to keep the system fabrication simple and highlight the performance of the prediction models rather than the engineering



**Fig. 7. Gathering data.** (A and B) Real images of foam and optical fiber assembly during deformation in darkness. (C) Schematic of training data collection process.

challenges. To remove ambient light interference, we would also need to use an optically opaque elastomer skin. We did not do this here to facilitate troubleshooting. Last, one would need to design a mount for the robot to gather accurate data for the ML models.

Our current system detects four deformation modes; however, we believe that other deformation modes could be added as needed. We also suspect that, with more sophisticated ML techniques, the sensor system could detect deformations that were not predetermined by the experimenters. Further research will investigate this possibility to achieve arbitrary deformation detection in soft robots.

We have seen in biology (39–41) and in engineering (42, 43) that more complete and accurate sensing enables better control. This work is a step toward making soft robots more reliably controllable and more responsive to their environment. With this kind of sensing capability, soft robots could protect themselves by responding to excessive deformation. Walking soft robots could improve their locomotion by learning better walking gaits through proprioception. In addition, they could relearn to walk after experiencing limb damage.

We present an optical robotic device that can sense multiple deformation types without sensors that have been specifically patterned for each type. Although the deformation classification is limited to four modes (i.e., bend positive, bend negative, twist positive, and twist negative), we hypothesize that this method could be used for many deformation types. We also believe that classifying arbitrary deformation may be possible with more sophisticated ML techniques.

## MATERIALS AND METHODS

### Research objectives and design

Our objective is to demonstrate that elastomeric foam (and by extension, robotic elastomeric foam actuators) can be imparted with proprioception through optical sensing and the application of basic ML and statistical methods. To make the work accessible, we chose a readily available, soft lithography process to fabricate the sensor system and implemented commonly built-in ML algorithms.

### Sensor design and fabrication

We wanted the sensor system to be easily integrated into a soft robotic actuator; therefore, our sensor design is identical to that of our previously

published soft foam actuators (3, 44): We fabricated an open-cell, lost salt silicone foam block, which we embedded with optical fibers and sealed with a solid silicone skin. This fabrication technique can be used to create 3D shapes, which makes this work generalizable to other soft mechanisms. We chose optical sensing because, unlike its resistive and capacitive counterparts, it requires no embedded electronics, can sample a large volume with few probes, and is minimally affected by changes in temperature. For the embedded optics, we used plastic optical fibers with radius  $r \approx 0.25$  mm ([www.thefiber opticstore.com](http://www.thefiber opticstore.com)), which experience low loss ( $\Gamma < 0.25$  dB  $m^{-1}$  for  $\lambda \approx 650$  nm) and can be used both to illuminate and to detect light scattered in the foam. Plastic optical fibers can also be thermally shaped, which facilitates fiber terminal placement inside the foam. The input light came from a constant output, visible light source (115 V; MI-150 Fiber Optic Illuminator, Edmund Optics) to enable consistent results and to facilitate troubleshooting, respectively. For manufacturing simplicity, we used a camera (EO-13122C Color USB 3.0, Edmund Optics) to detect the diffuse reflected light exiting the fibers.

We used soft lithography to fabricate the sensor, which allowed us to pattern the optical fibers as a layer of the fabrication process. Because we selected silicone rubber as the base material, the sensor can achieve high extensibility and experience little hysteresis. In addition, silicone comes in a large range of elastic moduli, enabling the generalization of our design to a variety of applications. We chose Smooth-On's Ecoflex 0030 specifically for its translucence and low tangent moduli, facilitating internal illumination and enabling large deformations for small forces, respectively. We fabricated the optical foam assembly by first thermally forming optical fibers to form a planar array of fiber terminals and then casting and curing silicone rubber around those fibers (Fig. 1A, top). Next, we casted a mix of table salt and uncured silicone on top of the exposed fiber terminals, allowed the silicone to cure, and then dissolved the salt out in water (Fig. 1A, middle). In the last soft lithography step, we sealed the foam with solid silicone skin (Fig. 1A, bottom). For reproducible training, we mounted one end of the optical foam into a bending and twisting apparatus and mounted the other end to a rigid post (Figs. 2 and 6A). Using a 3D printed (Objet30 Scholar, Stratasys Inc.; VeroBlue material) connector and epoxy, the loose fiber terminals were directed into a chamber containing a beam splitter (50R/50T Plate Beamsplitter, Edmund Optics). We also pointed the illumination source and the camera into the beam splitter chamber in a configuration that separated the light entering the fibers from the reflected diffuse light exiting the fibers (Fig. 6B).

## Experimental design

### Model selection

One way to reconstruct the shape of a deformed elastomeric foam by using the waveguide output intensities would be to derive a complete theoretical model of the system. To effectively achieve this goal, however, we would at minimum need accurate dynamic models of the complex interactions among features such as foam porosity, strut size, strut shape, refractive index, absorbance, reflectivity, input light wavelength, optical fiber position, and optical fiber orientation. In the absence of a complete and accurate theoretical model, we used ML and statistical techniques to generate our models. We compared several ML models, all of which can be implemented using built-in software packages. We chose to use MATLAB to facilitate transferring of data from the camera to the prediction models, and to implement the different ML techniques, we used the following toolboxes: Statistics and Machine Learning Toolbox, Curve Fitting Toolbox, and Deep Learning Toolbox.

### ML implementation

To gather data, we covered the optical setup (Fig. 6) to avoid interference due to changes in ambient light. With the illuminator on, we deformed the foam to a known bend or twist angle, saved an image of the fiber terminals, calculated the average intensity of each fiber terminal, and saved those scalar values in a vector of length 30. For single-output prediction, we repeated this measurement 2020 times for bend and twist angles in the range  $-80^\circ$  to  $90^\circ$  and  $-82^\circ$  to  $90^\circ$ , respectively, resulting in a feature matrix,  $\mathbf{X}$ , of dimension 2020 by 30 and in a label matrix,  $\mathbf{Y}$ , of dimension 2020 by 2 (Fig. 7). The first column of  $\mathbf{Y}$  held the values for deformation mode, and the second column held the values for angle. We gathered 290 test data points in the same manner. For multi-output prediction, we repeated the above process to gather 956 training data points and 239 test points.

To train the categorical classifiers, we used the built-in MATLAB functions `fitcknn` for kNN and `fitcecoc` for both SVMs and the decision tree. To train the regression models, we grouped the training data by deformation mode and then generated four regression models—two for each deformation mode using the built-in MATLAB functions `knnsearch` for kNN, `fitrsvm` for SVMs, `fitrtree` for the decision tree, `feedforwardnet` and `train` for MLPs, `fitlm` for the linear model, and `fitrgp` for GPs. To train the multi-output models, we used `knnsearch` for kNN, `feedforwardnet` and `train` for MLPs, and `mvregress` for the linear model. For each model, we performed a random hyperparameter search to find the best model. Table S1 displays the best hyperparameter sets found by our search.

### SUPPLEMENTARY MATERIALS

[robotics.sciencemag.org/cgi/content/full/3/24/eaau2489/DC1](http://robotics.sciencemag.org/cgi/content/full/3/24/eaau2489/DC1)

Fig. S1. Random versus greedy feature removal.

Table S1. Model parameters for best prediction models.

Table S2. Model evaluation times.

Movie S1. Real-time deformation prediction.

### REFERENCES AND NOTES

- C.-P. Chou, B. Hannaford, Measurement and modeling of McKibben pneumatic artificial muscles. *IEEE Trans. Rob. Autom.* **12**, 90–102 (1996).
- B. Tondu, P. Lopez, Modeling and control of McKibben artificial muscle robot actuators. *IEEE Control Syst. Mag.* **20**, 15–38 (2000).
- B. C. Mac Murray, X. An, S. S. Robinson, I. M. van Meerbeek, K. W. O'Brien, H. Zhao, R. F. Shepherd, Poroelastic foams for simple fabrication of complex soft robots. *Adv. Mater.* **27**, 6334–6340 (2015).
- B. N. Peele, T. J. Wallin, H. Zhao, R. F. Shepherd, 3D printing antagonistic systems of artificial muscle using projection stereolithography. *Bioinspir. Biomim.* **10**, 055003 (2015).
- R. V. Martinez, J. L. Branch, C. R. Fish, L. Jin, R. F. Shepherd, R. M. Nunes, Z. Suo, G. M. Whitesides, Robotic tentacles with three-dimensional mobility based on flexible elastomers. *Adv. Mater.* **25**, 205–212 (2013).
- C. Laschi, L. Margheri, M. Cianchetti, P. Dario, C. Laschi, Soft robot arm inspired by the octopus: II. From artificial requirements to innovative technological solutions. *Adv. Robot.* **26**, 025005 (2012).
- K. Suzumori, S. Iikura, H. Tanaka, Development of flexible microactuator and its applications to robotic mechanisms. *IEEE Int. Conf. Robot. Autom.* 1622–1627 (1991).
- H. Zhao, J. Jalving, R. Huang, R. Knepper, A. Ruina, R. Shepherd, A helping hand: Soft orthosis with integrated optical strain sensors and EMG control. *IEEE Robot. Autom. Mag.* **23**, 55–64 (2016).
- H. Zhao, R. Huang, R. F. Shepherd, Curvature control of soft orthotics via low cost solid-state optoelectronics. *IEEE Int. Conf. Robot. Autom.* 4008–4013 (2016).
- A. D. Marchese, R. Tedrake, D. Rus, Dynamics and trajectory optimization for a soft spatial fluidic elastomer manipulator. *Int. J. Rob. Res.* **35**, 1000–1019 (2016).
- H. Zhao, Y. Li, A. Elsamadisi, R. Shepherd, Scalable manufacturing of high force wearable soft actuators. *Extrem. Mech. Lett.* **3**, 89–104 (2015).
- H. K. Yap, H. Y. Ng, C.-H. Yeow, High-force soft printable pneumatics for soft robotic applications. *Soft Robot.* **3**, 144–158 (2016).

13. Y.-L. Park, C. Majidi, R. Kramer, P. Bérard, R. J. Wood, Hyperelastic pressure sensing with a liquid-embedded elastomer. *J. Micromech. Microeng.* **20**, 125029 (2010).
14. L. Wang, T. Ding, P. Wang, Thin flexible pressure sensor array based on carbon black/silicone rubber nanocomposite. *IEEE Sens. J.* **9**, 1130–1135 (2009).
15. Y.-L. Park, B.-R. Chen, R.-J. Wood, Design and fabrication of soft artificial skin using embedded microchannels and liquid conductors. *IEEE Sens. J.* **12**, 2711–2718 (2012).
16. J. L. Schneider, T. B. Sheridan, An optical tactile sensor for manipulators. *Robot. Comput. Integr. Manuf.* **1**, 65–71 (1984).
17. S. S. Robinson, K. W. O'Brien, H. Zhao, B. N. Peele, C. M. Larson, B. C. Mac Murray, I. M. Van Meerbeek, S. N. Dunham, R. F. Shepherd, Integrated soft sensors and elastomeric actuators for tactile machines with kinesthetic sense. *Extrem. Mech. Lett.* **5**, 47–53 (2015).
18. H. Zhao, K. O'Brien, S. Li, R. F. Shepherd, Optoelectronically innervated soft prosthetic hand via stretchable optical waveguides. *Sci. Robot.* **1**, eaai7529 (2016).
19. C. Larson, B. Peele, S. Li, S. Robinson, M. Totaro, L. Beccai, B. Mazzolai, R. Shepherd, Highly stretchable electroluminescent skin for optical signaling and tactile sensing. *Science* **351**, 1071–1074 (2016).
20. Y.-L. Park, K. Chau, R. J. Black, M. R. Cutkosky, Force sensing robot fingers using embedded fiber Bragg grating sensors and shape deposition manufacturing, in *Proceedings 2007 IEEE International Conference on Robotics and Automation (2007)*, pp. 1510–1516.
21. C. To, T. L. Hellebrekers, Y.-L. Park, Highly stretchable optical sensors for pressure, strain, and curvature measurement, in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2015)*, pp. 5898–5903.
22. F. L. Hammond, Y. Mengüç, R. J. Wood, Toward a modular soft sensor-embedded glove for human hand motion and tactile pressure measurement, in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (2014)*, pp. 4000–4007.
23. J. W. Boley, E. L. White, G. T.-C. Chiu, R. K. Kramer, Direct writing of gallium-indium alloy for stretchable electronics. *Adv. Funct. Mater.* **24**, 3501–3507 (2014).
24. C. Larson, J. Spjut, R. Knepper, R. Shepherd, A deformable interface for human touch recognition using stretchable carbon nanotube dielectric elastomer sensors and deep neural networks. arXiv:1706.02542 (8 June 2017).
25. H. A. Sonar, M. C.-S. Yuen, R. Kramer-bottiglio, J. Paik, An any-resolution distributed pressure localization scheme using a capacitive soft sensor skin, in *2018 IEEE International Conference on Soft Robotics (RoboSoft) (2018)*, pp. 1–6.
26. D. Hughes, J. Lammie, N. Correll, A robotic skin for collision avoidance and affective touch recognition. *IEEE Robot. Autom. Lett.* **3**, 1386–1393 (2018).
27. Y. Sugiura, G. Kakehi, A. Withana, C. Lee, D. Sakamoto, M. Sugimoto, M. Inami, T. Igarashi, Detecting shape deformation of soft objects using directional photoreflexivity measurement, in *Proceedings of the 24th annual ACM symposium on user interface software and technology (UIST'11) (2011)*, pp. 509.
28. S. John, F. C. MacKintosh, Diffusing wave spectroscopy and multiple scattering of light in correlated random media. *Phys. Rev. B. Condens. Matter* **40**, 2383–2406 (1989).
29. J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization. *J. Mach. Learn. Res.* **13**, 281–305 (2012).
30. R. Kohavi, A study of Cross validation and bootstrap for accuracy estimation and model selection, in *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCI95) (1995)*.
31. T. M. Cover, P. E. Hart, Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory* **1**, 21–27 (1967).
32. D. E. Adamo, B. J. Martin, S. H. Brown, Age-related differences in upper limb proprioceptive acuity. *Percept. Mot. Skills* **104**, 1297–1309 (2007).
33. D. J. Goble, S. H. Brown, Task-dependent asymmetries in the utilization of proprioceptive feedback for goal-directed movement. *Exp. Brain Res.* **180**, 693–704 (2007).
34. D. E. Adamo, N. D. Alexander, S. H. Brown, The influence of age and physical activity on upper limb proprioceptive ability. *J. Aging Phys. Act.* **17**, 272–293 (2009).
35. D. J. Goble, J. P. Coxon, N. Wenderoth, A. Van Impe, S. P. Swinnen, Proprioceptive sensibility in the elderly: Degeneration, functional consequences and plastic-adaptive processes. *Neurosci. Biobehav. Rev.* **33**, 271–278 (2009).
36. W. R. Ferrell, A. Smith, Position sense at the proximal interphalangeal joint of the human index finger. *J. Physiol.* **399**, 49–61 (1988).
37. M. Peters, K. Mackenzie, P. Bryden, Finger length and distal finger extent patterns in humans. *Am. J. Phys. Anthropol.* **117**, 209–217 (2002).
38. A. Buryanov, V. Kotiuk, Proportions of hand segments. *Int. J. Morphol.* **28**, 755–758 (2010).
39. Z. Hasan, D. G. Stuart, Animal solutions to problems of movement control: the role of proprioceptors. *Annu. Rev. Neurosci.* **11**, 199–223 (1988).
40. F. Horak, D. M. Corcos, F. Horak, H. Poizner, M. Shapiro, P. Tuite, J. Volkman, M. Maschke, Proprioception and motor control in Parkinson's disease. *J. Mot. Behav.* **41**, 543–552 (2009).
41. B. L. Riemann, S. M. Lephart, The sensorimotor system, Part II: The role of proprioception in motor control and functional joint stability. *J. Athl. Train.* **37**, 80–84 (2002).
42. M. Bergerman, Y. Xu, Robust joint and cartesian control of underactuated manipulators. *J. Dyn. Syst. Meas. Control* **118**, 557–565 (1996).
43. H. Arai, S. Tachi, Position control of manipulator with passive joints using dynamic coupling. *IEEE Trans. Rob. Autom.* **7**, 528–534 (1991).
44. A. Argiolas, B. C. Mac Murray, I. Van Meerbeek, J. Whitehead, E. Sinibaldi, B. Mazzolai, R. F. Shepherd, Sculpting soft machines. *Soft Robot.* **3**, 101–108 (2016).

**Acknowledgments:** We thank G. Hoffman for input on the sensor design. **Funding:** This work was supported in part by the Air Force Office of Scientific Research (award number FA9550-18-1-0243), the NSF Graduate Research Fellowship Program (grant number DGE-1144153), and a grant from the Alfred P. Sloan Foundation. **Author contributions:** I.M.V.M. designed and fabricated the sensor system, designed and conducted the experiments, analyzed the data, and wrote the manuscript. R.F.S. initialized the concept, supervised the experiments, and wrote the manuscript. C.M.D.S. partially designed the ML experiments. **Competing interests:** R.F.S. and I.M.V.M. are inventors on a patent application (no. 62/444,581) submitted by Cornell University that covers elastomeric foam sensors. C.M.D.S. declares no competing interests. **Data and materials availability:** All data needed to evaluate the conclusion in this paper are present in the paper or the Supplementary Materials.

Submitted 22 May 2018  
 Accepted 9 November 2018  
 Published 28 November 2018  
 10.1126/scirobotics.aau2489

**Citation:** I. M. Van Meerbeek, C. M. De Sa, R. F. Shepherd, Soft optoelectronic sensory foams with proprioception. *Sci. Robot.* **3**, eaau2489 (2018).

## Soft optoelectronic sensory foams with proprioception

I. M. Van Meerbeek, C. M. De Sa and R. F. Shepherd

*Sci. Robotics* **3**, eaau2489.  
DOI: 10.1126/scirobotics.aau2489

### ARTICLE TOOLS

<http://robotics.sciencemag.org/content/3/24/eaau2489>

### SUPPLEMENTARY MATERIALS

<http://robotics.sciencemag.org/content/suppl/2018/11/26/3.24.eaau2489.DC1>

### REFERENCES

This article cites 35 articles, 1 of which you can access for free  
<http://robotics.sciencemag.org/content/3/24/eaau2489#BIBL>

### PERMISSIONS

<http://www.sciencemag.org/help/reprints-and-permissions>

Use of this article is subject to the [Terms of Service](#)