**Supplementary information**

**Automatic Strain Sensor Design *via* Active Learning and Data Augmentation for Soft Machines**

Haitao Yang,‡a Jiali Li,‡a Kai Zhuo Lim,a Chuanji Pan,a Tien Van Truong,b Qian Wang,a Kerui Li,a Shuo Li,a Meng Ding,a Xiaoli Liu,a Qian Xie,c Pablo Valdivia y Alvarado,b Xiaonan Wang,a\* Po-Yen Chena\*

a Department of Chemical and Biomolecular Engineering, National University of Singapore, 4 Engineering Drive 4, Singapore 117585, Singapore.

b Engineering Product Development, Singapore University of Technology and Design, 8 Somapah Road, Singapore 487372.

c School of Metallurgic Engineering, Anhui University of Technology, 59 Hudong Road, Ma’anshan, P.R. China 243002.

‡ These authors contribute equally to this work.

\* Email: [chewxia@nus.edu.sg](mailto:chewxia@nus.edu.sg) (X. Wang); checp@umd.edu (P.-Y. Chen)

This PDF file includes:

**Experimental Section**

**Note S1** to **S12**

**Fig. S1** to **Fig. S13**

**Table S1** to **S9**

**Supplementary Video 1** and **Video 2**

**Note S1. Different fracture mechanisms** **observed in *G0*, *G1*-1D, and *G1*-2D sensors under** **uniaxial strains.**

As shown in **Fig. S7a**,under 5% strain, the planar ps-MXene layer of *G0* sensor exhibited visible cracks perpendicular to the axis of applied strains. As the strain increased to 10%, the size of surface cracks quickly grew, and the electrical resistances increased to ~1 kΩ. As the strain increased above 10%, larger surface cracks completely cut off the conductive pathways of ps-MXene layer.

As shown in **Fig. S7b**,under 10% strain, the wrinkle-textured ps-MXene layer of *G1*-1D sensor exhibited smaller surface cracks and slower crack propagation, compared to the planar ps-MXene layer of *G0* sensor. As the strain increased to 30%, the conductive pathways were not completely cut off, and the electrical resistance increased to ~4 kΩ.

As shown in **Fig. S7c**,when the applied strain was lower than 40%, no surface cracks were developed on the crumple-textured ps-MXene layer of *G1*-2D sensor, and the isotropic crumples were gradually deformed into periodic wrinkles. When the strain was above 40%, tiny surface cracks were observed on the deformed wrinkles, and the cracks continued to increase. Even under 100% strain, the crumple-textured ps-MXene layer still remained conductive with the electrical resistance of ~5 kΩ.

**Note S2.** **Estimated number of experiments required to build a four-DOF dataset for automatic sensor design.**

Four degrees of freedom (DOF) were recognized in the fabrication process of *Gn* sensors, including PVA loading, SWNT loading, sensing layer thickness, and morphology. We set 2.0 wt.% as the step size for both PVA and SWNT loadings, so the total steps were calculated to be 1,250 (50×50/2) for varying two DOFs of sensing layer composition. We set 50 nm as the step size for varying sensing layer thickness from 200 to 2,000 nm (36 total steps). Three different types of surface morphologies were introduced, including planar, wrinkle-textured, and crumple-textured ps-MXene layers. The number of experiments required to construct a full-map dataset across four DOFs was estimated to be 135,000 (1,250×36×3).

**Note S3. Training of a support-vector machine (SVM) classifier.**

To ensure the decision programs of navigation model to suggest the fabrication recipes with high detachment chances for the next loop of active learning, a SVM classifier was first trained to recognize three detachment cases (i.e., “feasible”, “fractured”, and “fail”) of ps-MXene layers at different PVA and SWNT loadings.

Three steps were implemented using Python to construct a SVM classifier, including (1) selecting a kernel function, (2) importing data points to train a SVM classifier, and (3) optimizing as-trained SVM classifier. In this work, as the collected feasibility grades were not shown to be linear (**Fig. 2b**), we decided to use a kernel function to map low-dimension data points into a higher dimensional feature space to find the optimal hyperplanes with maximal margin distances.5

For the first step, a radial basis function (RBF) was selected as the kernel function to deal with the nonlinear data points. Afterwards, a SVM classifier was trained by inputting 351 feasibility grades. For the last step of classifier optimization, Matthew’s correlation coefficient (MCC) was used to adjust the hyperparameter values.6 MCC is normally used as the main indicator to compare the prediction capability of a model with a random guess, which can be used to measure whether the performance of SVM classifier is affected by class imbalance. If the tested result of MCC is negative, it indicates that the prediction capability of a model is worse than a random guess. If the tested result of MCC is positive, it indicates that the prediction capability of a model is better than a random guess. As MCC reaches to its maximum (i.e., 1.0), the model shows perfect prediction. The formula of MCC is provided in **Equation S1**:

(**S1**)

, where TP means true positive (SVM classifier prediction is true, and real value is true), TN means true negative (SVM classifier prediction is false, and real value is false), FP means false positive (SVM classifier prediction is true, but real value is false), and FN means false negative (SVM classifier prediction is false, but real value is true). Our SVM classifier was trained by 351 data points with the MCC value of 0.964. The open source code to implement SVM in Python is provided in **GitHub** (https://github.com/jiali1025/Automatic\_Strain\_Sensor\_Design).

**Note S4. Terms used in active learning loops.**

Several important terms were used in this work. First, we named the ML model that performed the space exploration during active learning loops as “navigation model”. After 12 loops of active learning, the dataset of navigation model contained 125 data points cumulatively. Each data point had 8 labels, which included 4 fabrication recipes (e.g., composition, thickness, morphology of ps-MXene layer) and 4 sensor characteristics (*ε0*, *ε10*, *ε100*, *εmax*). 6 decision programs were trained by decision tree (DT) and artificial neural network (ANN) algorithms. Each decision program possessed independent training hyperparameters (such as learning rate) to estimate the uncertainty of targeted data points (on basis of *A Score*) for next loop of active learning.

**Note S5. Calculation of *A Score* acquisition function.**

A suitable acquisition function was introduced in the active learning loops to suggest the targeted data points with the highest uncertainty in the sensor design space. We defined the acquisition function as *A Score* in **Equation S2**,

(**S2**)

, where *L*2 denotes the shortest mathematicaldistance (also called Euclidian distance) between current recipe labels (within the dataset of navigation model) and targeted recipe labels (not yet included in the dataset of navigation model). In particular, *L*2 is calculated by **Equation S3**,

(**S3**)

, where *N* is the cumulative number of data points in current dataset, *s*i, *p*i, *t*i, and *m*i represent SWNT loading, PVA loading, sensing layer thickness, and morphology of one known data point (*i*) within the navigation model, and *s*j, *p*j, *t*j, and *m*j are the recipe labels of one targeted data point (*j*) outside the navigation model. On the other hand, denotes the variance of predicted strain labels from 6 decision programs (3 DT-trained and 3 ANN-trained programs), which is defined in **Equation S4**,

(**S4**)

, where *N* is the total number of decision program (*N* = 6), , ,, andare the output strain labels predicted by the *j*th decision program on basis of the recipe labels of a targeted data point, , , andare the average strain labels predicted by 6 decision programs on basis of the recipe labels of a targeted data point.

**Note S6. Comparison of various navigation models using different acquisition functions.**

Three different acquisition functions were examined, where the decision programs suggested the targeted data points with the largest *A Scores*, with the largest variance, or through random selection. Detailed implementations of these acquisition functions in Python are described in **Note S10**–**S12**. To represent the degree of space exploration of the trained navigation models, we utilized the average mathematical distance (abbreviated as ) between collected recipe labels, defined by **Equation S5**,

(**S5**)

, where *N* is the cumulative number of data points, *s*i, *p*i, *t*i, and *m*i are the recipe labels (including SWNT loading, PVA loading, sensing layer thickness, and morphology) of one data point (*i*), and *s*j, *p*j, *t*j, and *m*j are the recipe labels of another data point (*j*). A higher indicates a wider distribution of data points in the sensor design space, and a lower corresponds to close clusters formed among existing data points. As shown in **Table S8** and **S9**, by using variance-based or random suggestion-based acquisition function, the resulting navigation models suggested the targeted data points with very similar recipe labels. After first loop of active learning, in **Fig. 3b**, the values of three trained navigation model were calculated to be 0.06 (on basis of variance), 0.14 (through random selection), and 0.24 (on basis of *A Score*). These results indicated that, with *A Score* as the acquisition function, the navigation model did explore the targeted data points across the multi-dimensional design space and formed a wider data distribution.

On the other hand, the prediction accuracy of trained navigation model was quantified by using mean squared error (MSE) in **Equation S6**,

(**S6**)

, where *N* is the cumulative number of test data (*N* = 30), is the model-predicted strain labels on basis of a test data (*i*), is the actual strain values of a test data (*i*). A smaller MSE value indicates higher prediction accuracy of decision programs and *vice versa*. As a result, in **Fig. 3b**, the MSE values of three trained navigation model were calculated to be 4,503 (on basis of variance), 3,502 (through random selection), and 806 (on basis of *A Score*). These results indicate that *A Score* is a better acquisition function for training the navigation model with both wide data distribution and high prediction accuracy.

**Note S7. Construction of an ultimate prediction model through data augmentation and genetic algorithm selection.**

To construct an ultimate prediction model for automatic sensor design, the data points collected from active learning loops were augmented *via* two different methods, “Synthetic Minority Oversampling Technique for REGression (SMOTE-REG)” and “User Input Principle (UIP)”. The SMOTE-REG method was modified from the original SMOTE method to fit our regression problem.7 The SMOTE-REG method was used to construct virtual data points between two real data points by using linear interpolation. The UIP method was based on the physical principles suggested by the expert users. For example, within a very small change of specific fabrication parameters, the sensor characteristics were approximately invariant. Based on our observation in **Table S4**, with ±2.0 wt.% composition change or ±5 nm thickness changes, the *G1*-1D sensors showed approximately the same sensor characteristics. Based on our 125 data points, we synthesized 10 virtual data points per real data point with slightly different recipe labels (with ±2.0 wt.% composition changes or ±5 nm thickness changes) yet identical strain labels. The open source code to implement SMOTE-REG and UIP methods in Python is provided in **GitHub** (https://github.com/jiali1025/Automatic\_Strain\_Sensor\_Design).

After data augmentation *via* SMOTE-REG and UIP methods, the virtual and real data points were used to train two pools of decision programs (over 300 programs for each pool, by DT and ANN algorithms). Afterwards, we adopted the genetic algorithm (GA) selection to select a set of decision programs with optimal prediction performance. In particular, GA selection performed many iterations with a population of Boolean vectors (with each entry containing the vector of being 0 (not included) or 1 (included)). For each Boolean vector, the 10-fold cross-validation error was calculated. At the end of each iteration, tournament selection was performed to select the vectors with the lowest mean relative error (MRE, see calculation in **Equation S7**), while mutation and crossover operations were used to introduce diversity.8

(**S7**)

, where *N* is the cumulative number of test data (*N* = 30), , , and are the strain labels predicted by selected decision programs on basis of a test data (*i*), , , and are the actual strain values of a test data (*i*). Constructed through UIP method followed by GA selection, the model with best performance was assembled by 16 decision programs (3 by DT and 13 by ANN), which demonstrated the lowest MRE of 24%. The open source code to implement GA selection in Python is provided in **GitHub** (https://github.com/jiali1025/Automatic\_Strain\_Sensor\_Design).

**Note S8. Implementation of statistical analyses based on Pearson’s coefficients in Python.**

Pearson’s coefficient (Pearson’s *r*) describes the degree of linear correlation between two sets of parameters.9 The value of Pearson’s *r* ranges from –1 to +1 for perfect negative to perfect positive correlations. Meanwhile, *p* value is usually calculated to evaluate whether the correlation (either negative or positive) is significant. Pearson’s *r* () between fabrication parameters and sensor characteristics was calculated by **Equation S8**,

(**S8**)

, where is the number of data points, is the selected fabrication parameter (e.g., SWNT loading), is the average value of selected fabrication parameter, is the selected sensor characteristic (e.g., *εmax*), and is the average value of selected sensor characteristic. The open source code to implement statistical analyses based on Pearson’s *r* in Python is provided in **GitHub** (https://github.com/jiali1025/Automatic\_Strain\_Sensor\_Design).

**Note S9.** **Automatic strain sensor design.**

First, the required sensor characteristics to monitor a specific soft machine were entered into the ultimate prediction model. For example, in order to monitor the actuating behaviors of a soft gripper, we entered “*ε100* < 15% and *εmax* > 30%” into the ultimate prediction model. To monitor the actuating behaviors of a soft swimmer robot, we entered “*ε100* < 0.8% and *εmax* > 1.0%”. Afterwards, we utilized a particle swarm optimization method (PSO) to find feasible fabrication recipes that led to the sensor characteristics with minimal deviations from the input requests. The open source code to implement inverse sensor design in Python is provided in **GitHub** (https://github.com/jiali1025/Automatic\_Strain\_Sensor\_Design).

**Note S10. Implementation of *A Score*-based active learning loops in Python.**

Four major steps were implemented in Python to conduct *A Score*-based active learning loops. The first step was to train 6 decision programs by DT and ANN with different hyperparameters; 3 programs were trained by DT, and 3 programs were trained by ANN. The second step was to calculate *A Scores* (defined in **Equation S2**) over a large number of targeted data points outside the navigation model. The third step is to select the targeted data points with the highest *A Scores*. The fourth step is to apply the trained SVM classifier to filter these targeted data points, and the recipe labels with high detachment chances were then suggested for next loop of active learning. The open source code to implement *A Score*-based active learning loops in Python is provided in **GitHub** (https://github.com/jiali1025/Automatic\_Strain\_Sensor\_Design).

**Note S11. Implementation of variance-based active learning loops in Python.**

Four major steps were implemented in Python to conduct variance-based active learning loop to benchmark the performance of the developed framework. The first step was to train 6 decision programs by DT and ANN with different hyperparameters; 3 programs were trained by DT, and 3 programs were trained by ANN. The second step was to calculate the variance(, defined in **Equation S4**) over a large number of targeted data points outside the navigation model. The third step is to select the targeted data points with the highest variance values. The fourth step is to apply the trained SVM classifier to filter these targeted data points, and the recipe labels with high detachment chances were then suggested for next loop of active learning. The open source code to implement variance-based active learning loops in Python is provided in **GitHub** (https://github.com/jiali1025/Automatic\_Strain\_Sensor\_Design).

**Note S12. Implementation of random suggestion-based active learning loops in Python.**

Three major steps were implemented in Python to conduct active learning loop by random selection to benchmark the performance of the developed framework. First, a pool of *N* targeted data points was generated through a random function of Python. For the second step, the trained SVM classifier was applied to remove the data points with low detachment chances. Third, the recipe labels with high detachment chances were then suggested for next loop of active learning. The open source code to implement active learning loops based on random suggestion in Python is provided in **GitHub** (https://github.com/jiali1025/Automatic\_Strain\_Sensor\_Design).

**Table S1 Feasibility grades of delaminating ps-MXene layers from PVDF membranes.** “Feasible” cases refer to the conditions that the filteredps-MXene layers were completely detached from PVDF membranes. “Fragile” cases refer to the conditions that the filtered ps-MXene layers exhibited visible fractures after the detachment. “Fail” cases refer to the conditions that the filtered ps-MXene layers were stuck on PVDF membranes. A total of 351 delamination tests were conducted.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Feasibility Grades of ps-MXene Layer Delamination | | | | | | | | | | |
| Index | **MXene Loading**  **(wt.%)** | **SWNT Loading**  **(wt.%)** | **PVA Loading**  **(wt.%)** | **Case**  **(–)** |  | **Index** | **MXene**  **Loading**  **(wt.%)** | **SWNT**  **Loading**  **(wt.%)** | **PVA**  **Loading**  **(wt.%)** | **Case**  **(–)** |
| 1 | 100 | 0 | 0 | Feasible |  | **177** | 12 | 28 | 60 | Fail |
| 2 | 96 | 0 | 4 | Feasible |  | **178** | 8 | 28 | 64 | Fail |
| 3 | 92 | 0 | 8 | Feasible |  | **179** | 4 | 28 | 68 | Fail |
| 4 | 88 | 0 | 12 | Feasible |  | **180** | 0 | 28 | 72 | Fail |
| 5 | 84 | 0 | 16 | Feasible |  | **181** | 68 | 32 | 0 | Feasible |
| 6 | 80 | 0 | 20 | Fragile |  | **182** | 64 | 32 | 4 | Feasible |
| 7 | 76 | 0 | 24 | Fragile |  | **183** | 60 | 32 | 8 | Feasible |
| 8 | 72 | 0 | 28 | Fragile |  | **184** | 56 | 32 | 12 | Feasible |
| 9 | 68 | 0 | 32 | Fragile |  | **185** | 52 | 32 | 16 | Feasible |
| 10 | 64 | 0 | 36 | Fail |  | **186** | 48 | 32 | 20 | Feasible |
| 11 | 60 | 0 | 40 | Fail |  | **187** | 44 | 32 | 24 | Fragile |
| 12 | 56 | 0 | 44 | Fail |  | **188** | 40 | 32 | 28 | Fail |
| 13 | 52 | 0 | 48 | Fail |  | **189** | 36 | 32 | 32 | Fail |
| 14 | 48 | 0 | 52 | Fail |  | **190** | 32 | 32 | 36 | Fail |
| 15 | 44 | 0 | 56 | Fail |  | **191** | 28 | 32 | 40 | Fail |
| 16 | 40 | 0 | 60 | Fail |  | **192** | 24 | 32 | 44 | Fail |
| 17 | 36 | 0 | 64 | Fail |  | **193** | 20 | 32 | 48 | Fail |
| 18 | 32 | 0 | 68 | Fail |  | **194** | 16 | 32 | 52 | Fail |
| 19 | 28 | 0 | 72 | Fail |  | **195** | 12 | 32 | 56 | Fail |
| 20 | 24 | 0 | 76 | Fail |  | **196** | 8 | 32 | 60 | Fail |
| 21 | 20 | 0 | 80 | Fail |  | **197** | 4 | 32 | 64 | Fail |
| 22 | 16 | 0 | 84 | Fail |  | **198** | 0 | 32 | 68 | Fail |
| 23 | 12 | 0 | 88 | Fail |  | **199** | 64 | 36 | 0 | Feasible |
| 24 | 8 | 0 | 92 | Fail |  | **200** | 60 | 36 | 4 | Feasible |
| 25 | 4 | 0 | 96 | Fail |  | **201** | 56 | 36 | 8 | Feasible |
| 26 | 0 | 0 | 100 | Fail |  | **202** | 52 | 36 | 12 | Feasible |
| 27 | 96 | 4 | 0 | Feasible |  | **203** | 48 | 36 | 16 | Feasible |
| 28 | 92 | 4 | 4 | Feasible |  | **204** | 44 | 36 | 20 | Feasible |
| 29 | 88 | 4 | 8 | Feasible |  | **205** | 40 | 36 | 24 | Feasible |
| 30 | 84 | 4 | 12 | Feasible |  | **206** | 36 | 36 | 28 | Fail |
| 31 | 80 | 4 | 16 | Feasible |  | **207** | 32 | 36 | 32 | Fragile |
| 32 | 76 | 4 | 20 | Fragile |  | **208** | 28 | 36 | 36 | Fail |
| 33 | 72 | 4 | 24 | Fragile |  | **209** | 24 | 36 | 40 | Fail |
| 34 | 68 | 4 | 28 | Fragile |  | **210** | 20 | 36 | 44 | Fail |
| 35 | 64 | 4 | 32 | Fail |  | **211** | 16 | 36 | 48 | Fail |
| 36 | 60 | 4 | 36 | Fail |  | **212** | 12 | 36 | 52 | Fail |
| 37 | 56 | 4 | 40 | Fail |  | **213** | 8 | 36 | 56 | Fail |
| 38 | 52 | 4 | 44 | Fail |  | **214** | 4 | 36 | 60 | Fail |
| 39 | 48 | 4 | 48 | Fail |  | **215** | 0 | 36 | 64 | Fail |
| 40 | 44 | 4 | 52 | Fail |  | **216** | 60 | 40 | 0 | Feasible |
| 41 | 40 | 4 | 56 | Fail |  | **217** | 56 | 40 | 4 | Feasible |
| 42 | 36 | 4 | 60 | Fail |  | **218** | 52 | 40 | 8 | Feasible |
| 43 | 32 | 4 | 64 | Fail |  | **219** | 48 | 40 | 12 | Feasible |
| 44 | 28 | 4 | 68 | Fail |  | **220** | 44 | 40 | 16 | Feasible |
| 45 | 24 | 4 | 72 | Fail |  | **221** | 40 | 40 | 20 | Feasible |
| 46 | 20 | 4 | 76 | Fail |  | **222** | 36 | 40 | 24 | Feasible |
| 47 | 16 | 4 | 80 | Fail |  | **223** | 32 | 40 | 28 | Fragile |
| 48 | 12 | 4 | 84 | Fail |  | **224** | 28 | 40 | 32 | Fragile |
| 49 | 8 | 4 | 88 | Fail |  | **225** | 24 | 40 | 36 | Fragile |
| 50 | 4 | 4 | 92 | Fail |  | **226** | 20 | 40 | 40 | Fragile |
| 51 | 0 | 4 | 96 | Fail |  | **227** | 16 | 40 | 44 | Fail |
| 52 | 92 | 8 | 0 | Feasible |  | **228** | 12 | 40 | 48 | Fail |
| 53 | 88 | 8 | 4 | Feasible |  | **229** | 8 | 40 | 52 | Fail |
| 54 | 84 | 8 | 8 | Feasible |  | **230** | 4 | 40 | 56 | Fail |
| 55 | 80 | 8 | 12 | Feasible |  | **231** | 0 | 40 | 60 | Fail |
| 56 | 76 | 8 | 16 | Feasible |  | **232** | 56 | 44 | 0 | Feasible |
| 57 | 72 | 8 | 20 | Fragile |  | **233** | 52 | 44 | 4 | Feasible |
| 58 | 68 | 8 | 24 | Fail |  | **234** | 48 | 44 | 8 | Feasible |
| 59 | 64 | 8 | 28 | Fail |  | **235** | 44 | 44 | 12 | Feasible |
| 60 | 60 | 8 | 32 | Fail |  | **236** | 40 | 44 | 16 | Feasible |
| 61 | 56 | 8 | 36 | Fail |  | **237** | 36 | 44 | 20 | Feasible |
| 62 | 52 | 8 | 40 | Fail |  | **238** | 32 | 44 | 24 | Feasible |
| 63 | 48 | 8 | 44 | Fail |  | **239** | 28 | 44 | 28 | Feasible |
| 64 | 44 | 8 | 48 | Fail |  | **240** | 24 | 44 | 32 | Feasible |
| 65 | 40 | 8 | 52 | Fail |  | **241** | 20 | 44 | 36 | Fragile |
| 66 | 36 | 8 | 56 | Fail |  | **242** | 16 | 44 | 40 | Fragile |
| 67 | 32 | 8 | 60 | Fail |  | **243** | 12 | 44 | 44 | Fail |
| 68 | 28 | 8 | 64 | Fail |  | **244** | 8 | 44 | 48 | Fail |
| 69 | 24 | 8 | 68 | Fail |  | **245** | 4 | 44 | 52 | Fail |
| 70 | 20 | 8 | 72 | Fail |  | **246** | 0 | 44 | 56 | Fail |
| 71 | 16 | 8 | 76 | Fail |  | **247** | 52 | 48 | 0 | Feasible |
| 72 | 12 | 8 | 80 | Fail |  | **248** | 48 | 48 | 4 | Feasible |
| 73 | 8 | 8 | 84 | Fail |  | **249** | 44 | 48 | 8 | Feasible |
| 74 | 4 | 8 | 88 | Fail |  | **250** | 40 | 48 | 12 | Feasible |
| 75 | 0 | 8 | 92 | Fail |  | **251** | 36 | 48 | 16 | Feasible |
| 76 | 88 | 12 | 0 | Feasible |  | **252** | 32 | 48 | 20 | Feasible |
| 77 | 84 | 12 | 4 | Feasible |  | **253** | 28 | 48 | 24 | Feasible |
| 78 | 80 | 12 | 8 | Feasible |  | **254** | 24 | 48 | 28 | Feasible |
| 79 | 76 | 12 | 12 | Feasible |  | **255** | 20 | 48 | 32 | Feasible |
| 80 | 72 | 12 | 16 | Feasible |  | **256** | 16 | 48 | 36 | Feasible |
| 81 | 68 | 12 | 20 | Fragile |  | **257** | 12 | 48 | 40 | Fragile |
| 82 | 64 | 12 | 24 | Fail |  | **258** | 8 | 48 | 44 | Fail |
| 83 | 60 | 12 | 28 | Fail |  | **259** | 4 | 48 | 48 | Fail |
| 84 | 56 | 12 | 32 | Fail |  | **260** | 0 | 48 | 52 | Fail |
| 85 | 52 | 12 | 36 | Fail |  | **261** | 48 | 52 | 0 | Feasible |
| 86 | 48 | 12 | 40 | Fail |  | **262** | 44 | 52 | 4 | Feasible |
| 87 | 44 | 12 | 44 | Fail |  | **263** | 40 | 52 | 8 | Feasible |
| 88 | 40 | 12 | 48 | Fail |  | **264** | 36 | 52 | 12 | Feasible |
| 89 | 36 | 12 | 52 | Fail |  | **265** | 32 | 52 | 16 | Feasible |
| 90 | 32 | 12 | 56 | Fail |  | **266** | 28 | 52 | 20 | Feasible |
| 91 | 28 | 12 | 60 | Fail |  | **267** | 24 | 52 | 24 | Feasible |
| 92 | 24 | 12 | 64 | Fail |  | **268** | 20 | 52 | 28 | Feasible |
| 93 | 20 | 12 | 68 | Fail |  | **269** | 16 | 52 | 32 | Feasible |
| 94 | 16 | 12 | 72 | Fail |  | **270** | 12 | 52 | 36 | Feasible |
| 95 | 12 | 12 | 76 | Fail |  | **271** | 8 | 52 | 40 | Fragile |
| 96 | 8 | 12 | 80 | Fail |  | **272** | 4 | 52 | 44 | Fail |
| 97 | 4 | 12 | 84 | Fail |  | **273** | 0 | 52 | 48 | Fail |
| 98 | 0 | 12 | 88 | Fail |  | **274** | 44 | 56 | 0 | Feasible |
| 99 | 84 | 16 | 0 | Feasible |  | **275** | 40 | 56 | 4 | Feasible |
| 100 | 80 | 16 | 4 | Feasible |  | **276** | 36 | 56 | 8 | Feasible |
| 101 | 76 | 16 | 8 | Feasible |  | **277** | 32 | 56 | 12 | Feasible |
| 102 | 72 | 16 | 12 | Feasible |  | **278** | 28 | 56 | 16 | Feasible |
| 103 | 68 | 16 | 16 | Feasible |  | **279** | 24 | 56 | 20 | Feasible |
| 104 | 64 | 16 | 20 | Fragile |  | **280** | 20 | 56 | 24 | Feasible |
| 105 | 60 | 16 | 24 | Fragile |  | **281** | 16 | 56 | 28 | Feasible |
| 106 | 56 | 16 | 28 | Fail |  | **282** | 12 | 56 | 32 | Feasible |
| 107 | 52 | 16 | 32 | Fail |  | **283** | 8 | 56 | 36 | Feasible |
| 108 | 48 | 16 | 36 | Fail |  | **284** | 4 | 56 | 40 | Feasible |
| 109 | 44 | 16 | 40 | Fail |  | **285** | 0 | 56 | 44 | Fail |
| 110 | 40 | 16 | 44 | Fail |  | **286** | 40 | 60 | 0 | Feasible |
| 111 | 36 | 16 | 48 | Fail |  | **287** | 36 | 60 | 4 | Feasible |
| 112 | 32 | 16 | 52 | Fail |  | **288** | 32 | 60 | 8 | Feasible |
| 113 | 28 | 16 | 56 | Fail |  | **289** | 28 | 60 | 12 | Feasible |
| 114 | 24 | 16 | 60 | Fail |  | **290** | 24 | 60 | 16 | Feasible |
| 115 | 20 | 16 | 64 | Fail |  | **291** | 20 | 60 | 20 | Feasible |
| 116 | 16 | 16 | 68 | Fail |  | **292** | 16 | 60 | 24 | Feasible |
| 117 | 12 | 16 | 72 | Fail |  | **293** | 12 | 60 | 28 | Feasible |
| 118 | 8 | 16 | 76 | Fail |  | **294** | 8 | 60 | 32 | Feasible |
| 119 | 4 | 16 | 80 | Fail |  | **295** | 4 | 60 | 36 | Feasible |
| 120 | 0 | 16 | 84 | Fail |  | **296** | 0 | 60 | 40 | Feasible |
| 121 | 80 | 20 | 0 | Feasible |  | **297** | 36 | 64 | 0 | Feasible |
| 122 | 76 | 20 | 4 | Feasible |  | **298** | 32 | 64 | 4 | Feasible |
| 123 | 72 | 20 | 8 | Feasible |  | **299** | 28 | 64 | 8 | Feasible |
| 124 | 68 | 20 | 12 | Feasible |  | **300** | 24 | 64 | 12 | Feasible |
| 125 | 64 | 20 | 16 | Feasible |  | **301** | 20 | 64 | 16 | Feasible |
| 126 | 60 | 20 | 20 | Feasible |  | **302** | 16 | 64 | 20 | Feasible |
| 127 | 56 | 20 | 24 | Fail |  | **303** | 12 | 64 | 24 | Feasible |
| 128 | 52 | 20 | 28 | Fail |  | **304** | 8 | 64 | 28 | Feasible |
| 129 | 48 | 20 | 32 | Fail |  | **305** | 4 | 64 | 32 | Feasible |
| 130 | 44 | 20 | 36 | Fail |  | **306** | 0 | 64 | 36 | Feasible |
| 131 | 40 | 20 | 40 | Fail |  | **307** | 32 | 68 | 0 | Feasible |
| 132 | 36 | 20 | 44 | Fail |  | **308** | 28 | 68 | 4 | Feasible |
| 133 | 32 | 20 | 48 | Fail |  | **309** | 24 | 68 | 8 | Feasible |
| 134 | 28 | 20 | 52 | Fail |  | **310** | 20 | 68 | 12 | Feasible |
| 135 | 24 | 20 | 56 | Fail |  | **311** | 16 | 68 | 16 | Feasible |
| 136 | 20 | 20 | 60 | Fail |  | **312** | 12 | 68 | 20 | Feasible |
| 137 | 16 | 20 | 64 | Fail |  | **313** | 8 | 68 | 24 | Feasible |
| 138 | 12 | 20 | 68 | Fail |  | **314** | 4 | 68 | 28 | Feasible |
| 139 | 8 | 20 | 72 | Fail |  | **315** | 0 | 68 | 32 | Feasible |
| 140 | 4 | 20 | 76 | Fail |  | **316** | 28 | 72 | 0 | Feasible |
| 141 | 0 | 20 | 80 | Fail |  | **317** | 24 | 72 | 4 | Feasible |
| 142 | 76 | 24 | 0 | Feasible |  | **318** | 20 | 72 | 8 | Feasible |
| 143 | 72 | 24 | 4 | Feasible |  | **319** | 16 | 72 | 12 | Feasible |
| 144 | 68 | 24 | 8 | Feasible |  | **320** | 12 | 72 | 16 | Feasible |
| 145 | 64 | 24 | 12 | Feasible |  | **321** | 8 | 72 | 20 | Feasible |
| 146 | 60 | 24 | 16 | Feasible |  | **322** | 4 | 72 | 24 | Feasible |
| 147 | 56 | 24 | 20 | Feasible |  | **323** | 0 | 72 | 28 | Feasible |
| 148 | 52 | 24 | 24 | Fail |  | **324** | 24 | 76 | 0 | Feasible |
| 149 | 48 | 24 | 28 | Fail |  | **325** | 20 | 76 | 4 | Feasible |
| 150 | 44 | 24 | 32 | Fail |  | **326** | 16 | 76 | 8 | Feasible |
| 151 | 40 | 24 | 36 | Fail |  | **327** | 12 | 76 | 12 | Feasible |
| 152 | 36 | 24 | 40 | Fail |  | **328** | 8 | 76 | 16 | Feasible |
| 153 | 32 | 24 | 44 | Fail |  | **329** | 4 | 76 | 20 | Feasible |
| 154 | 28 | 24 | 48 | Fail |  | **330** | 0 | 76 | 24 | Feasible |
| 155 | 24 | 24 | 52 | Fail |  | **331** | 20 | 80 | 0 | Feasible |
| 156 | 20 | 24 | 56 | Fail |  | **332** | 16 | 80 | 4 | Feasible |
| 157 | 16 | 24 | 60 | Fail |  | **333** | 12 | 80 | 8 | Feasible |
| 158 | 12 | 24 | 64 | Fail |  | **334** | 8 | 80 | 12 | Feasible |
| 159 | 8 | 24 | 68 | Fail |  | **335** | 4 | 80 | 16 | Feasible |
| 160 | 4 | 24 | 72 | Fail |  | **336** | 0 | 80 | 20 | Feasible |
| 161 | 0 | 24 | 76 | Fail |  | **337** | 16 | 84 | 0 | Feasible |
| 162 | 72 | 28 | 0 | Feasible |  | **338** | 12 | 84 | 4 | Feasible |
| 163 | 68 | 28 | 4 | Feasible |  | **339** | 8 | 84 | 8 | Feasible |
| 164 | 64 | 28 | 8 | Feasible |  | **340** | 4 | 84 | 12 | Feasible |
| 165 | 60 | 28 | 12 | Feasible |  | **341** | 0 | 84 | 16 | Feasible |
| 166 | 56 | 28 | 16 | Feasible |  | **342** | 12 | 88 | 0 | Feasible |
| 167 | 52 | 28 | 20 | Feasible |  | **343** | 8 | 88 | 4 | Feasible |
| 168 | 48 | 28 | 24 | Fail |  | **344** | 4 | 88 | 8 | Feasible |
| 169 | 44 | 28 | 28 | Fail |  | **345** | 0 | 88 | 12 | Feasible |
| 170 | 40 | 28 | 32 | Fail |  | **346** | 8 | 92 | 0 | Feasible |
| 171 | 36 | 28 | 36 | Fail |  | **347** | 4 | 92 | 4 | Feasible |
| 172 | 32 | 28 | 40 | Fail |  | **348** | 0 | 92 | 8 | Feasible |
| 173 | 28 | 28 | 44 | Fail |  | **349** | 4 | 96 | 0 | Feasible |
| 174 | 24 | 28 | 48 | Fail |  | **350** | 0 | 96 | 4 | Fail |
| 175 | 20 | 28 | 52 | Fail |  | **351** | 0 | 100 | 0 | Fail |
| 176 | 16 | 28 | 56 | Fail |  |  |  |  |  |  |

**Table S2 125 data points collected from 12 loops of active learning for navigation model construction.** A total of 12 loops of active learning were executed, and 125 sensors were fabricated cumulatively.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Index** | **Learning**  **Loop** | **Model-Suggested Recipe Labels** | | | | | **Measured Sensor Characteristics** | | | | |
| MXene  Loading  (wt.%) | SWNT  Loading  (wt.%) | PVA  Loading  (wt.%) | Layer  Thickness  (nm) | Layer  Morphology  (**–**) | *ε0*  (%) | *ε10*  (%) | *ε100*  (%) | *εmax*  (%) |
| **1** | 0 | 90 | 5 | 5 | 900 | *G0* | 0.0 | 0.3 | 0.3 | 10.0 |
| **2** | 0 | 90 | 5 | 5 | 300 | *G0* | 0.0 | 2.0 | 3.3 | 30.0 |
| **3** | 0 | 100 | 0 | 0 | 800 | *G1*-2D | 0.0 | 6.7 | 10.0 | 120.0 |
| **4** | 0 | 95 | 5 | 0 | 800 | *G1*-2D | 0.0 | 3.3 | 23.3 | 166.7 |
| **5** | 0 | 100 | 0 | 0 | 800 | *G1*-1D | 0.0 | 1.0 | 3.3 | 36.7 |
| **6** | 0 | 95 | 5 | 0 | 800 | *G1*-1D | 0.0 | 1.0 | 3.3 | 36.7 |
| **7** | 0 | 95 | 0 | 5 | 800 | *G1*-1D | 0.0 | 1.0 | 3.3 | 70.0 |
| **8** | 0 | 100 | 0 | 0 | 800 | *G*0 | 0.0 | 0.3 | 1.0 | 10.0 |
| **9** | 0 | 90 | 5 | 5 | 800 | *G1*-2D | 0.0 | 3.3 | 30.0 | 213.3 |
| **10** | 0 | 90 | 10 | 0 | 800 | *G0* | 0.0 | 3.3 | 3.3 | 13.3 |
| **11** | 0 | 92 | 4 | 4 | 800 | *G0* | 0.0 | 0.2 | 2.0 | 13.3 |
| **12** | 1 | 0 | 60 | 40 | 2,000 | *G0* | 0.0 | 9.1 | 9.5 | 10.0 |
| **13** | 1 | 0 | 60 | 40 | 1,380 | *G0* | 0.0 | 5.3 | 5.9 | 10.5 |
| **14** | 1 | 83 | 17 | 0 | 1,700 | *G0* | 0.0 | 1.1 | 1.8 | 4.2 |
| **15** | 1 | 2 | 58 | 40 | 2,000 | *G1*-1D | 0.0 | 10.6 | 10.6 | 10.6 |
| **16** | 1 | 4 | 56 | 40 | 2,000 | *G1*-2D | 0.0 | 13.3 | 21.7 | 22.8 |
| **17** | 2 | 83 | 0 | 17 | 2,000 | *G0* | 0.0 | 7.0 | 11.3 | 13.4 |
| **18** | 2 | 9 | 52 | 39 | 1,980 | *G0* | 0.0 | 6.8 | 7.4 | 10.5 |
| **19** | 2 | 83 | 0 | 17 | 1,440 | *G0* | 0.0 | 15.2 | 17.1 | 20.0 |
| **20** | 2 | 0 | 96 | 4 | 200 | *G1*-2D | 0.0 | 11.1 | 14.1 | 25.3 |
| **21** | 2 | 83 | 0 | 17 | 1,990 | *G1*-2D | 0.0 | 11.9 | 58.7 | 115.9 |
| **22** | 3 | 3 | 57 | 40 | 220 | *G1*-1D | 0.0 | 7.8 | 9.4 | 20.0 |
| **23** | 3 | 4 | 56 | 40 | 200 | *G0* | 0.0 | 18.0 | 19.8 | 32.0 |
| **24** | 3 | 49 | 30 | 21 | 216 | *G0* | 0.0 | 0.7 | 1.7 | 5.0 |
| **25** | 3 | 16 | 48 | 36 | 200 | *G1*-2D | 0.0 | 11.1 | 18.5 | 20.0 |
| **26** | 3 | 47 | 31 | 22 | 215 | *G1*-1D | 0.0 | 8.2 | 17.3 | 36.4 |
| **27** | 3 | 83 | 0 | 17 | 217 | *G1*-2D | 0.0 | 42.2 | 100.0 | 133.3 |
| **28** | 3 | 83 | 0 | 17 | 213 | *G1*-1D | 0.0 | 14.0 | 24.0 | 60.0 |
| **29** | 3 | 11 | 51 | 38 | 1,171 | *G1*-1D | 0.0 | 16.7 | 20.0 | 33.3 |
| **30** | 4 | 83 | 0 | 17 | 1,839 | *G1*-1D | 0.0 | 1.0 | 20.0 | 33.0 |
| **31** | 4 | 7 | 93 | 0 | 264 | *G1*-1D | 0.0 | 4.0 | 9.0 | 30.0 |
| **32** | 4 | 5 | 95 | 0 | 389 | *G0* | 0.0 | 2.0 | 14.0 | 60.0 |
| **33** | 4 | 25 | 43 | 32 | 907 | *G0* | 0.0 | 5.0 | 6.0 | 12.0 |
| **34** | 4 | 70 | 25 | 5 | 1,744 | *G0* | 0.0 | 0.6 | 4.4 | 6.7 |
| **35** | 4 | 55 | 24 | 21 | 1,405 | *G1*-1D | 0.0 | 18.0 | 20.0 | 28.0 |
| **36** | 4 | 31 | 41 | 28 | 966 | *G1*-2D | 0.0 | 21.0 | 22.0 | 24.0 |
| **37** | 4 | 4 | 94 | 2 | 1,266 | *G1*-1D | 0.0 | 9.5 | 10.0 | 12.2 |
| **38** | 5 | 12 | 88 | 0 | 1,127 | *G1*-2D | 0.0 | 2.5 | 3.0 | 10.0 |
| **39** | 5 | 73 | 27 | 0 | 1,438 | *G1*-2D | 0.0 | 21.4 | 27.1 | 50.0 |
| **40** | 5 | 1 | 95 | 4 | 1,131 | *G0* | 0.0 | 2.1 | 4.7 | 11.6 |
| **41** | 5 | 60 | 38 | 2 | 686 | *G1*-2D | 0.0 | 30.0 | 45.0 | 53.0 |
| **42** | 5 | 36 | 62 | 2 | 205 | *G1*-2D | 0.0 | 21.3 | 33.8 | 81.3 |
| **43** | 5 | 63 | 36 | 1 | 1,651 | *G1*-1D | 0.0 | 2.0 | 4.0 | 26.7 |
| **44** | 5 | 0 | 60 | 40 | 1,642 | *G1*-2D | 0.0 | 15.5 | 16.0 | 18.5 |
| **45** | 5 | 80 | 17 | 3 | 1,875 | *G1*-1D | 0.0 | 2.0 | 32.7 | 35.3 |
| **46** | 6 | 50 | 50 | 0 | 800 | *G0* | 0.0 | 2.0 | 3.3 | 6.7 |
| **47** | 6 | 20 | 80 | 0 | 800 | *G0* | 0.0 | 1.0 | 3.3 | 10.0 |
| **48** | 6 | 72 | 28 | 0 | 800 | *G0* | 0.0 | 1.0 | 3.3 | 6.7 |
| **49** | 6 | 12 | 88 | 0 | 800 | *G0* | 0.0 | 3.3 | 6.7 | 20.0 |
| **50** | 6 | 28 | 72 | 0 | 800 | *G0* | 0.0 | 1.0 | 3.3 | 13.3 |
| **51** | 6 | 96 | 4 | 0 | 800 | *G0* | 0.0 | 1.0 | 3.3 | 16.7 |
| **52** | 6 | 88 | 0 | 12 | 800 | *G0* | 0.0 | 0.1 | 0.5 | 3.3 |
| **53** | 6 | 99 | 0 | 1 | 800 | *G0* | 0.0 | 3.3 | 6.7 | 13.3 |
| **54** | 6 | 45 | 45 | 10 | 800 | *G0* | 0.0 | 3.3 | 6.7 | 36.7 |
| **55** | 6 | 72 | 14 | 14 | 560 | *G0* | 0.0 | 1.0 | 6.7 | 43.3 |
| **56** | 6 | 32 | 62 | 6 | 640 | *G0* | 0.0 | 1.0 | 3.3 | 16.7 |
| **57** | 6 | 0 | 100 | 0 | 800 | *G0* | 0.0 | 13.3 | 16.7 | 23.3 |
| **58** | 6 | 55 | 33 | 12 | 720 | *G0* | 0.0 | 1.0 | 3.3 | 10.0 |
| **59** | 6 | 60 | 20 | 20 | 800 | *G0* | 0.0 | 1.0 | 2.0 | 4.0 |
| **60** | 6 | 56 | 28 | 16 | 800 | *G0* | 0.0 | 3.8 | 4.0 | 6.7 |
| **61** | 6 | 52 | 32 | 16 | 800 | *G0* | 0.0 | 2.0 | 3.3 | 4.0 |
| **62** | 6 | 90 | 5 | 5 | 1,200 | *G0* | 0.0 | 1.0 | 1.0 | 6.7 |
| **63** | 6 | 90 | 5 | 5 | 1,500 | *G0* | 0.0 | 1.0 | 1.0 | 6.7 |
| **64** | 6 | 90 | 5 | 5 | 1,800 | *G0* | 0.0 | 0.5 | 2.0 | 3.3 |
| **65** | 6 | 90 | 5 | 5 | 800 | *G1*-1D | 0.0 | 3.3 | 26.7 | 50.0 |
| **66** | 6 | 84 | 0 | 16 | 800 | *G1*-1D | 0.0 | 3.3 | 13.3 | 66.7 |
| **67** | 6 | 80 | 20 | 0 | 800 | *G1*-1D | 0.0 | 3.3 | 13.3 | 40.0 |
| **68** | 6 | 70 | 25 | 5 | 800 | *G0* | 0.0 | 1.2 | 4.7 | 14.1 |
| **69** | 6 | 70 | 25 | 5 | 800 | *G1*-1D | 0.0 | 4.4 | 8.9 | 38.9 |
| **70** | 6 | 59 | 21 | 20 | 515 | *G1*-2D | 0.0 | 40.0 | 55.0 | 84.0 |
| **71** | 6 | 68 | 14 | 18 | 1,063 | *G1*-2D | 0.0 | 25.7 | 37.1 | 42.9 |
| **72** | 6 | 66 | 16 | 18 | 1,847 | *G1*-2D | 0.0 | 50.0 | 60.0 | 65.0 |
| **73** | 6 | 87 | 10 | 3 | 1,241 | *G1*-2D | 0.0 | 50.0 | 65.0 | 85.0 |
| **74** | 6 | 77 | 6 | 17 | 1,438 | *G1*-2D | 0.0 | 19.0 | 20.0 | 21.0 |
| **75** | 6 | 1 | 91 | 8 | 1,973 | *G1*-2D | 0.0 | 32.0 | 32.0 | 33.3 |
| **76** | 6 | 78 | 22 | 0 | 266 | *G1*-2D | 0.0 | 28.8 | 46.9 | 121.9 |
| **77** | 6 | 83 | 13 | 4 | 1,877 | *G1*-2D | 0.0 | 41.0 | 85.7 | 91.4 |
| **78** | 7 | 87 | 6 | 7 | 469 | *G1*-2D | 0.0 | 60.0 | 123.5 | 225.0 |
| **79** | 7 | 87 | 5 | 8 | 201 | *G1*-2D | 0.0 | 95.0 | 225.0 | 350.0 |
| **80** | 7 | 25 | 72 | 3 | 691 | *G1*-2D | 0.0 | 23.8 | 35.0 | 47.5 |
| **81** | 7 | 78 | 5 | 17 | 1,420 | *G1*-2D | 0.0 | 150.0 | 250.0 | 250.0 |
| **82** | 7 | 80 | 3 | 17 | 891 | *G1*-2D | 0.0 | 87.5 | 175.0 | 200.0 |
| **83** | 7 | 92 | 4 | 4 | 1,687 | *G1*-2D | 0.0 | 80.0 | 175.0 | 225.0 |
| **84** | 7 | 88 | 7 | 5 | 1,939 | *G1*-2D | 0.0 | 50.0 | 75.0 | 143.8 |
| **85** | 7 | 90 | 6 | 4 | 1,431 | *G1*-2D | 0.0 | 75.0 | 162.5 | 175.0 |
| **86** | 8 | 81 | 4 | 15 | 1,225 | *G1*-2D | 0.0 | 75.0 | 165.0 | 240.0 |
| **87** | 8 | 81 | 4 | 15 | 1,691 | *G1*-2D | 0.0 | 35.0 | 100.0 | 140.0 |
| **88** | 8 | 80 | 4 | 16 | 682 | *G1*-2D | 0.0 | 85.0 | 165.0 | 320.0 |
| **89** | 8 | 45 | 41 | 14 | 221 | *G1*-2D | 0.0 | 25.0 | 35.0 | 105.0 |
| **90** | 8 | 90 | 5 | 5 | 1,031 | *G1*-2D | 0.0 | 33.3 | 93.3 | 166.7 |
| **91** | 8 | 3 | 90 | 7 | 1,998 | *G1*-1D | 0.0 | 8.0 | 9.0 | 14.0 |
| **92** | 8 | 82 | 4 | 14 | 1,049 | *G1*-2D | 0.0 | 40.0 | 115.0 | 155.0 |
| **93** | 8 | 97 | 3 | 0 | 206 | *G1*-2D | 0.0 | 40.0 | 102.9 | 171.4 |
| **94** | 9 | 81 | 5 | 14 | 429 | *G1*-1D | 0.0 | 1.0 | 38.0 | 155.0 |
| **95** | 9 | 82 | 5 | 13 | 1,211 | *G1*-1D | 0.0 | 2.0 | 18.0 | 70.0 |
| **96** | 9 | 88 | 5 | 7 | 515 | *G0* | 0.0 | 1.0 | 5.0 | 23.0 |
| **97** | 9 | 81 | 4 | 15 | 354 | *G0* | 0.0 | 2.0 | 6.0 | 45.0 |
| **98** | 9 | 49 | 44 | 7 | 1,923 | *G1*-2D | 0.0 | 28.0 | 30.0 | 35.0 |
| **99** | 9 | 85 | 5 | 10 | 603 | *G1*-1D | 0.0 | 2.0 | 5.0 | 85.0 |
| **100** | 9 | 81 | 4 | 15 | 1,550 | *G1*-1D | 0.0 | 2.0 | 8.0 | 46.0 |
| **101** | 9 | 12 | 81 | 7 | 402 | *G1*-2D | 0.0 | 15.0 | 25.0 | 60.0 |
| **102** | 10 | 35 | 60 | 5 | 438 | *G1*-1D | 0.0 | 2.5 | 6.3 | 45.0 |
| **103** | 10 | 28 | 60 | 12 | 1,372 | *G1*-2D | 0.0 | 17.0 | 22.0 | 29.0 |
| **104** | 10 | 64 | 18 | 18 | 214 | *G1*-2D | 0.0 | 31.7 | 63.0 | 150.0 |
| **105** | 10 | 2 | 73 | 25 | 445 | *G1*-1D | 0.0 | 8.0 | 36.0 | 72.0 |
| **106** | 10 | 2 | 76 | 22 | 438 | *G0* | 0.0 | 8.0 | 12.0 | 34.0 |
| **107** | 10 | 0 | 86 | 14 | 431 | *G1*-1D | 0.0 | 8.0 | 18.0 | 62.0 |
| **108** | 10 | 27 | 48 | 25 | 425 | *G0* | 0.0 | 8.0 | 28.0 | 56.0 |
| **109** | 10 | 60 | 37 | 3 | 432 | *G1*-1D | 0.0 | 2.0 | 18.0 | 105.0 |
| **110** | 11 | 8 | 91 | 1 | 1,940 | *G0* | 0.0 | 1.0 | 5.7 | 8.6 |
| **111** | 11 | 38 | 59 | 3 | 1,586 | *G0* | 0.0 | 4.3 | 4.8 | 11.3 |
| **112** | 11 | 3 | 57 | 40 | 679 | *G1*-2D | 0.0 | 25.0 | 28.0 | 31.0 |
| **113** | 11 | 93 | 4 | 3 | 278 | *G1*-1D | 0.0 | 1.0 | 18.0 | 90.0 |
| **114** | 11 | 79 | 5 | 16 | 530 | *G0* | 0.0 | 3.0 | 6.0 | 18.0 |
| **115** | 11 | 78 | 5 | 17 | 708 | *G0* | 0.0 | 1.0 | 5.0 | 12.0 |
| **116** | 11 | 80 | 5 | 15 | 205 | *G0* | 0.0 | 3.8 | 13.3 | 47.6 |
| **117** | 11 | 94 | 5 | 1 | 1,761 | *G1*-1D | 0.0 | 2.0 | 18.0 | 80.0 |
| **118** | 12 | 72 | 28 | 0 | 1,046 | *G1*-2D | 0.0 | 45.0 | 60.0 | 130.0 |
| **119** | 12 | 2 | 75 | 23 | 970 | *G1*-1D | 0.0 | 5.0 | 8.0 | 22.0 |
| **120** | 12 | 58 | 25 | 17 | 838 | *G1*-2D | 0.0 | 40.0 | 50.0 | 100.0 |
| **121** | 12 | 38 | 38 | 24 | 1,553 | *G1*-2D | 0.0 | 23.8 | 33.3 | 40.0 |
| **122** | 12 | 82 | 17 | 1 | 632 | *G1*-2D | 0.0 | 65.0 | 120.0 | 290.0 |
| **123** | 12 | 79 | 4 | 17 | 416 | *G1*-2D | 0.0 | 70.0 | 170.0 | 330.0 |
| **124** | 12 | 68 | 14 | 18 | 802 | *G1*-2D | 0.0 | 65.0 | 140.0 | 180.0 |
| **125** | 12 | 39 | 36 | 25 | 1,950 | *G1*-1D | 0.0 | 8.0 | 8.0 | 12.0 |

**Table S3 30 test data for evaluation of ML model’s prediction accuracy.**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test Data | | | | | | | | | |
| Index | **Fabrication Recipes** | | | | | **Measured Sensor Characteristics** | | | |
| MXene  Loading  (wt.%) | SWNT  Loading  (wt.%) | PVA  Loading  (wt.%) | Layer  Thickness  (nm) | Layer  Morphology  (**–**) | *ε0*  (%) | *ε10*  (%) | *ε100*  (%) | *εmax*  (%) |
| 1 | 13 | 79 | 8 | 1,947 | *G0* | 0.0 | 4.5 | 5.0 | 7.5 |
| 2 | 12 | 84 | 4 | 465 | *G1*-1D | 0.0 | 3.0 | 9.0 | 29.0 |
| 3 | 16 | 55 | 29 | 1,179 | *G1*-1D | 0.0 | 4.5 | 5.0 | 14.0 |
| 4 | 100 | 0 | 0 | 200 | *G1*-2D | 0.0 | 25.0 | 65.0 | 150.0 |
| 5 | 1 | 95 | 4 | 2,000 | *G1*-2D | 0.0 | 32.0 | 34.0 | 40.0 |
| 6 | 83 | 0 | 17 | 200 | *G0* | 0.0 | 3.8 | 15.1 | 30.2 |
| 7 | 5 | 95 | 0 | 226 | *G0* | 0.0 | 6.1 | 30.3 | 54.5 |
| 8 | 83 | 0 | 17 | 1,921 | *G1*-2D | 0.0 | 70.6 | 147.1 | 211.8 |
| 9 | 1 | 95 | 4 | 266 | *G1*-2D | 0.0 | 25.0 | 35.0 | 50.0 |
| 10 | 100 | 0 | 0 | 200 | *G1*-1D | 0.0 | 1.7 | 30.0 | 100.0 |
| 11 | 100 | 0 | 0 | 1,629 | *G0* | 0.0 | 0.5 | 2.0 | 5.0 |
| 12 | 100 | 0 | 0 | 2,000 | *G1*-1D | 0.0 | 3.0 | 17.0 | 36.3 |
| 13 | 7 | 54 | 39 | 1,126 | *G1*-2D | 0.0 | 30.0 | 35.0 | 45.0 |
| 14 | 9 | 52 | 39 | 928 | *G0* | 0.0 | 4.7 | 5.0 | 8.0 |
| 15 | 5 | 95 | 0 | 2,000 | *G1*-1D | 0.0 | 5.0 | 7.7 | 10.0 |
| 16 | 27 | 47 | 26 | 200 | *G1*-2D | 0.0 | 31.3 | 68.8 | 162.5 |
| 17 | 80 | 3 | 17 | 1,100 | *G1*-1D | 0.0 | 7.1 | 40.0 | 128.6 |
| 18 | 81 | 19 | 0 | 1,087 | *G1*-2D | 0.0 | 45.0 | 55.0 | 140.0 |
| 19 | 26 | 43 | 31 | 200 | *G1*-1D | 0.0 | 8.0 | 24.0 | 64.0 |
| 20 | 24 | 50 | 26 | 2,000 | *G1*-2D | 0.0 | 38.0 | 39.0 | 45.0 |
| 21 | 54 | 46 | 0 | 372 | *G0* | 0.0 | 2.0 | 5.0 | 45.0 |
| 22 | 17 | 83 | 0 | 1,140 | *G1*-2D | 0.0 | 30.0 | 65.0 | 66.0 |
| 23 | 1 | 95 | 4 | 1,113 | *G0* | 0.0 | 4.0 | 5.0 | 7.0 |
| 24 | 40 | 48 | 12 | 2,000 | *G1*-1D | 0.0 | 4.0 | 8.0 | 19.0 |
| 25 | 55 | 45 | 0 | 1,378 | *G0* | 0.0 | 2.0 | 3.0 | 11.0 |
| 26 | 34 | 39 | 27 | 2,000 | *G0* | 0.0 | 5.0 | 5.1 | 5.2 |
| 27 | 2 | 95 | 3 | 1,219 | *G1*-1D | 0.0 | 10.0 | 11.0 | 23.0 |
| 28 | 69 | 13 | 18 | 954 | *G0* | 0.0 | 2.8 | 3.0 | 5.0 |
| 29 | 64 | 36 | 0 | 716 | *G1*-1D | 0.0 | 4.0 | 8.0 | 40.0 |
| 30 | 0 | 63 | 37 | 200 | *G0* | 0.0 | 12.3 | 16.0 | 40.0 |

**Table S4 Fabrication recipes with small changes of fabrication parameters resulted in nearly the same sensor characteristics.**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Index | Fabrication Recipes | | | | | Measured Sensor Characteristics | | | |
| MXene  Loading  (wt.%) | SWNT  Loading  (wt.%) | PVA  Loading  (wt.%) | Layer  Thickness  (nm) | Layer  Morphology  (**–**) | *ε0*  (%) | *ε10*  (%) | *ε100*  (%) | *εmax*  (%) |
| 1 | 70 | 25 | 5 | 800 | *G1*-1D | 0.0 | 4.0 | 9.0 | 39.0 |
| 2 | 69 | 26 | 5 | 800 | *G1*-1D | 0.0 | 4.2 | 9.2 | 39.0 |
| 3 | 69 | 25 | 6 | 800 | *G1*-1D | 0.0 | 4.0 | 9.0 | 39.2 |
| 4 | 70 | 25 | 5 | 802 | *G1*-1D | 0.0 | 4.1 | 9.0 | 39.5 |
| 5 | 70 | 25 | 5 | 785 | *G1*-1D | 0.0 | 4.0 | 8.8 | 39.0 |

**Table S5. Prediction performance of an ultimate prediction model (after UIP method and GA selection) using 30 test data.** By comparing the model-predicted strain labels with actual strain values of test data (in **Table S3**), the MRE of an ultimate prediction model(after UIP method and GA selection)was calculated to be 24%.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Evaluation of Ultimate Prediction Model (After UIP Method and GA Selection) | | | | | | | | | |
| Index | **Fabrication Recipes in 30 Test Data** | | | | | **Model-Predicted Sensor Labels** | | | |
| MXene  Loading  (wt.%) | SWNT  Loading  (wt.%) | PVA  Loading  (wt.%) | Layer  Thickness  (nm) | Layer  Morphology  (**–**) | *ε0*  (%) | *ε10*  (%) | *ε100*  (%) | *εmax*  (%) |
| 1 | 13 | 79 | 8 | 1,947 | *G0* | 0.0 | 3.7 | 6.2 | 10.1 |
| 2 | 12 | 84 | 4 | 465 | *G1*-1D | 0.0 | 5.4 | 10.6 | 35.4 |
| 3 | 16 | 55 | 29 | 1,179 | *G1*-1D | 0.0 | 9.8 | 12.4 | 21.9 |
| 4 | 100 | 0 | 0 | 200 | *G1*-2D | 0.0 | 34.5 | 72.0 | 135.8 |
| 5 | 1 | 95 | 4 | 2,000 | *G1*-2D | 0.0 | 18.3 | 20.3 | 24.3 |
| 6 | 83 | 0 | 17 | 200 | *G0* | 0.0 | 4.7 | 9.8 | 28.2 |
| 7 | 5 | 95 | 0 | 226 | *G0* | 0.0 | 2.9 | 9.7 | 34.2 |
| 8 | 83 | 0 | 17 | 1,921 | *G1*-2D | 0.0 | 26.0 | 54.1 | 92.6 |
| 9 | 1 | 95 | 4 | 266 | *G1*-2D | 0.0 | 11.9 | 16.1 | 30.7 |
| 10 | 100 | 0 | 0 | 200 | *G1*-1D | 0.0 | 4.1 | 16.0 | 63.4 |
| 11 | 100 | 0 | 0 | 1,629 | *G0* | 0.0 | 0.3 | 1.5 | 6.7 |
| 12 | 100 | 0 | 0 | 2,000 | *G1*-1D | 0.0 | 3.3 | 17.9 | 51.8 |
| 13 | 7 | 54 | 39 | 1,126 | *G1*-2D | 0.0 | 17.5 | 20.4 | 24.4 |
| 14 | 9 | 52 | 39 | 928 | *G0* | 0.0 | 5.4 | 6.9 | 11.9 |
| 15 | 5 | 95 | 0 | 2,000 | *G1*-1D | 0.0 | 7.7 | 9.4 | 14.2 |
| 16 | 27 | 47 | 26 | 200 | *G1*-2D | 0.0 | 17.4 | 25.5 | 44.7 |
| 17 | 80 | 3 | 17 | 1,100 | *G1*-1D | 0.0 | 4.7 | 15.2 | 52.0 |
| 18 | 81 | 19 | 0 | 1,087 | *G1*-2D | 0.0 | 34.6 | 52.7 | 86.5 |
| 19 | 26 | 43 | 31 | 200 | *G1*-1D | 0.0 | 8.4 | 13.6 | 30.4 |
| 20 | 24 | 50 | 26 | 2,000 | *G1*-2D | 0.0 | 20.1 | 26.5 | 31.4 |
| 21 | 54 | 46 | 0 | 372 | *G0* | 0.0 | 1.2 | 4.7 | 21.0 |
| 22 | 17 | 83 | 0 | 1,140 | *G1*-2D | 0.0 | 8.2 | 10.6 | 16.7 |
| 23 | 1 | 95 | 4 | 1,113 | *G0* | 0.0 | 2.8 | 5.3 | 11.6 |
| 24 | 40 | 48 | 12 | 2,000 | *G1*-1D | 0.0 | 6.5 | 9.0 | 14.9 |
| 25 | 55 | 45 | 0 | 1,378 | *G0* | 0.0 | 2.5 | 3.8 | 9.3 |
| 26 | 34 | 39 | 27 | 2,000 | *G0* | 0.0 | 5.5 | 7.1 | 9.9 |
| 27 | 2 | 95 | 3 | 1,219 | *G1*-1D | 0.0 | 8.1 | 9.8 | 15.2 |
| 28 | 69 | 13 | 18 | 954 | *G0* | 0.0 | 1.7 | 3.0 | 6.5 |
| 29 | 64 | 36 | 0 | 716 | *G1*-1D | 0.0 | 3.2 | 9.7 | 39.0 |
| 30 | 0 | 63 | 37 | 200 | *G0* | 0.0 | 10.4 | 12.7 | 22.1 |

**Table S6. Recipe labels suggested by an ultimate prediction model for a soft gripper.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Index | Model-Suggested Recipe Labels | | | | |
| MXene  Loading  (wt.%) | SWNT  Loading  (wt.%) | PVA  Loading  (wt.%) | Layer  Thickness  (nm) | Layer  Morphology  (**–**) |
| 1 | 53 | 45 | 2 | 924 | *G1*-1D |
| 2 | 39 | 60 | 1 | 635 | *G1*-1D |
| 3 | 38 | 60 | 2 | 693 | *G1*-1D |
| 4 | 53 | 43 | 4 | 877 | *G1*-1D |
| 5 | 8 | 85 | 7 | 274 | *G0* |
| 6 | 57 | 38 | 5 | 1,032 | *G1*-1D |
| 7 | 61 | 34 | 5 | 1,114 | *G1*-1D |
| 8 | 47 | 45 | 8 | 859 | *G1*-1D |
| 9 | 4 | 85 | 11 | 240 | *G0* |
| 10 | 69 | 30 | 1 | 1,104 | *G1*-1D |

**Table S7. Recipe labels suggested by an ultimate prediction model for a soft swimmer robot.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Index | Model-Suggested Recipe Labels | | | | |
| MXene  Loading  (wt.%) | SWNT  Loading  (wt.%) | PVA  Loading  (wt.%) | Layer  Thickness  (nm) | Layer  Morphology  (**–**) |
| 1 | 95 | 5 | 0 | 1,484 | *G0* |
| 2 | 95 | 5 | 0 | 1,534 | *G0* |
| 3 | 95 | 5 | 0 | 1,482 | *G0* |
| 4 | 95 | 5 | 0 | 1,545 | *G0* |
| 5 | 95 | 5 | 0 | 1,530 | *G0* |
| 6 | 95 | 5 | 0 | 1,475 | *G0* |
| 7 | 95 | 5 | 0 | 1,500 | *G0* |
| 8 | 95 | 5 | 0 | 1,476 | *G0* |
| 9 | 95 | 5 | 0 | 1,536 | *G0* |
| 10 | 95 | 5 | 0 | 1,472 | *G0* |

**Table S8 Suggestion of targeted data points from a variance-based navigation model.** Based on the evaluation of variance values, the navigation model suggested the targeted data points with very similar recipe labels.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Targeted Data Points from Navigation Model Based on Calculation of Variance | | | | | | | | | |
| Index | **Model-Suggested Recipe Labels** | | | | | **Measured Sensor Characteristics** | | | |
| MXene  Loading  (wt.%) | SWNT  Loading  (wt.%) | PVA  Loading  (wt.%) | Layer  Thickness  (nm) | Layer  Morphology  (**–**) | *ε0*  (%) | *ε10*  (%) | *ε100*  (%) | *εmax*  (%) |
| 1 | 0 | 63 | 37 | 1,747 | *G1*-1D | 0.0 | 4.1 | 11.2 | 13.0 |
| 2 | 5 | 56 | 39 | 1,983 | *G1*-1D | 0.0 | 2.9 | 9.2 | 17.3 |
| 3 | 8 | 54 | 38 | 1,993 | *G1*-1D | 0.0 | 8.8 | 9.0 | 9.0 |
| 4 | 1 | 60 | 39 | 1,852 | *G1*-1D | 0.0 | 4.2 | 8.6 | 14.0 |
| 5 | 7 | 56 | 37 | 1,933 | *G1*-1D | 0.0 | 8.7 | 8.7 | 9.0 |

**Table S9 Suggestion of targeted data points from a random-based navigation model.** Through random suggestion, the navigation model suggested the targeted data points that led to similar sensor characteristics with all *εmax* < 40%.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Targeted Data Points from Navigation Model without Calculation of *A Score* | | | | | | | | | |
| Index | **Model-Suggested Recipe Labels** | | | | | **Measured Sensor Characteristics** | | | |
| MXene  Loading  (wt.%) | SWNT  Loading  (wt.%) | PVA  Loading  (wt.%) | Layer  Thickness  (nm) | Layer  Morphology  (**–**) | *ε0*  (%) | *ε10*  (%) | *ε100*  (%) | *εmax*  (%) |
| 1 | 57 | 23 | 20 | 1,748 | *G0* | 0.0 | 5.2 | 5.2 | 6.0 |
| 2 | 71 | 24 | 5 | 917 | *G0* | 0.0 | 6.3 | 10.0 | 13.5 |
| 3 | 13 | 80 | 7 | 263 | *G0* | 0.0 | 3.3 | 4.0 | 23.6 |
| 4 | 5 | 80 | 15 | 225 | *G0* | 0.0 | 4.0 | 7.6 | 18.0 |
| 5 | 10 | 53 | 37 | 855 | *G1*-1D | 0.0 | 5.0 | 18.0 | 38.2 |

**Supplementary Video 1.** A soft swimmer robot with model-suggested strain sensors for underwater exploration mission.

**Supplementary Video 2.** Real-time monitoring of a soft swimmer robot.

**Supporting References**

1 Alhabeb, M. *et al.* Guidelines for synthesis and processing of two-dimensional titanium carbide (Ti3C2Tx MXene). *Chem. Mater.* **29**, 7633-7644 (2017).

2 Shenton, M. J., Lovell-Hoare, M. C. & Stevens, G. C. Adhesion enhancement of polymer surfaces by atmospheric plasma treatment. *Journal of Physics D: Applied Physics* **34**, 2754-2760 (2001).

3 Low, J. H. *et al.* Hybrid tele-manipulation system using a sensorized 3-D-printed soft robotic gripper and a soft fabric-based haptic glove. *IEEE Robot. Autom. Let.* **2**, 880-887 (2017).

4 Truong, T. V., Viswanathan, V. K., Joseph, V. S. & Alvarado, P. V. y. Design and characterization of a fully autonomous under-actuated soft batoid-like robot. in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* 5826-5831.

5 Smola, A. J. & Schölkopf, B. A tutorial on support vector regression. *Stat. Comput.* **14**, 199-222 (2004).

6 Matthews, B. W. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochim. Biophys. Acta, Protein Struct.* **405**, 442-451 (1975).

7 Liu, X. *et al.* A robust low data solution: dimension prediction of semiconductor nanorods. *arXiv preprint arXiv:2010.14111* (2020).

8 Whitley, D. A genetic algorithm tutorial. *Stat. Comput.* **4**, 65-85 (1994).

9 Schober, P., Boer, C. & Schwarte, L. A. Correlation coefficients: appropriate use and interpretation. *Anesth. Analg.* **126**, 1763-1768 (2018).