Computational Mechanics
https://doi.org/10.1007/s00466-023-02284-0

ORIGINAL PAPER

A machine learning-based probabilistic computational framework for uncertainty quantification of actuation of clustered tensegrity structures

Yipeng Ge[1] · Zigang He[1] · Shaofan Li[2] · Liang Zhang[1] · Litao Shi[1,3]

Received: 27 August 2022 / Accepted: 8 February 2023

Abstract
Clustered tensegrity structures integrated with continuous cables are lightweight, foldable, and deployable. Thus, they can be used as flexible manipulators or soft robots. The actuation process of such soft structure has high probabilistic sensitivity. It is essential to quantify the uncertainty of actuated responses of the tensegrity structures and to modulate their deformation accurately. In this work, we propose a comprehensive data-driven computational approach to study the uncertainty quantification (UQ) and probability propagation in clustered tensegrity structures, and we have developed a surrogate optimization model to control the flexible structure deformation. An example of clustered tensegrity beam subjected to a clustered actuation is presented to demonstrate the validity of the approach and its potential application. The three main novelties of the data-driven framework are: (1) The proposed model can avoid the difficulty of convergence in nonlinear Finite Element Analysis (FEA), by two machine learning methods, the Gauss Process Regression (GPR) and Neutral Network (NN). (2) A fast real-time prediction on uncertainty propagation can be achieved by the surrogate model, and (3) Optimization of the actuated deformation comes true by using both Sequence Quadratic Programming (SQP) and Bayesian optimization methods. The results have shown that the proposed data-driven computational approach is powerful and can be extended to other UQ models or alternative optimization objectives.

Keywords Clustered tensegrity structure · Machine learning · Uncertainty quantification and propagation · Optimization inverse problem · Bayesian optimization

1 Introduction

Tensegrity structures have many advantages, such as lightweight, deployable, and changeable stiffness, and thus they find many applications in architecture, civil, aerospace, and biomedical engineering [1–5]. Recent novel applications are involved in soft robots that can walk on rough land, due to their excellent resistance to impact [6–10]. A new trend to combine tensegrity structures and artificial intelligence technology is emerging with the rapid developments in robotics and data science.

Tensegrity structures can be classified into two categories: classical and clustered. The clustered tensegrity is integrated by clustered (continuous) cables over the whole structure, whereas the classical structure does not. In the past two decades, extensive mechanics analyses and numerical simulations were conducted by many authors [11–17] to investigate the response of such structures. Existing computational methods for clustered tensegrity include the Force Density Method (FDM) [12], the Dynamic Relaxation Method (DRM) [14], and Finite Element Method (FEM) [15–17]. Material and geometric nonlinearities are both included in the numerical analysis. Material nonlinearity arises from the slack of cables or buckling of struts, while geometric nonlinearity comes from the large deformation and displacement between its initial configuration and its deformed configuration. The study conducted by Moored et al. [12] indicates that

✉ Liang Zhang
zhangl@cqu.edu.cn

1 College of Aerospace Engineering, Chongqing University, Chongqing 400044, People's Republic of China

2 Department of Civil and Environmental Engineering, University of California at Berkeley, Berkeley, CA 74720, USA

3 Shanghai Academy of Spaceflight Technology, Shanghai 201100, People's Republic of China

Published online: 05 March 2023

the number of internal mechanisms of clustered tensegrity structures increases significantly in comparison to classical structures [12]. For example, the state of prestress in the structure determines internal deformation mechanisms as well as the stiffness of the clustered tensegrity. Focusing on clustered tensegrity structures, Zhang et al. proposed a concise co-rotational formulation that has a quadratic convergence behavior [16]. The merit of the co-rotational formulation is that alternative constitutive laws such as plasticity can be considered directly in the local frame, and thus the existing finite element and constitutive formulations can be directly incorporated into the co-rotational framework. Kan et al. developed an efficient ANC FEM to deal with the friction between clustered cables and pulleys [17], in which the dynamic process of deployment was simulated, and the effect of friction on deployment was investigated numerically. These studies revealed that the convergence property of the numerical computations is sensitively dependent on the state of prestress. A low prestress producing more internal mechanisms is detrimental to the convergence of the computation algorithm. That is the main setback for the state-of-the-art numerical analysis of clustered tensegrity structures.

On the other hand, recently, a variety of data-driven computational methods were proposed for mechanical analysis as well as material and structural designs [18–26]. Kirchdoerfer and Ortiz developed a data-driven computational approach that was carried out directly from experimental material data and pertinent constraints and conservation laws [18]. An enhanced computational mechanics was proposed by Oishi and Yagawa to optimize quadrature for the FEM stiffness matrices, using a deep learning technology [19]. Oden and his co-authors developed a dynamic data-driven computational framework based on Bayesian inference and Karlman filtering. The framework included uncertainty quantification, model selection, and validation, and it was applied to the prediction of damage in composites, tumor growth, and evolution of COVID-19 [20–22]. Bessa et al. proposed a comprehensive data-driven computational framework to assist in the design and modeling of material properties and microstructure of composites [23]. The data-driven framework was extended to the design of ultra-thin shell structures in the stochastic post-buckling range by using Bayesian machine learning and optimization [24, 25]. A data-driven approach was proposed by Li et al. for the design of phononic crystals with anticipated band gaps [26]. A multi-layer perceptron was trained to establish an inherent relation between band gaps and topological features, and then the trained models were used for optimization.

Data-based machine learning methods were also proposed for the gait planning of tensegrity robots. For example, Kim et al. compared two techniques (greedy search and multi-generation Monte Carlo method), which were employed to find desired deformations and actuation strategies resulting in robust rolling locomotion of the robots [7]. Surovik et al. proposed Guided Policy Search (GPS), a sample-efficient and model-free hybrid framework for optimization and reinforcement learning, which has been used to produce periodic locomotion for a spherical 6-bar tensegrity robot on flat or slightly varied surfaces [8]. Kimber et al. demonstrated how unsupervised learning algorithms could be used to produce vibration-based locomotion [9]. By using a machine learning algorithm able to discover effective gaits with a minimal number of physical trials, Rieffel et al. made an easy-to-assemble tensegrity-based soft robot capable of highly dynamic locomotive gaits and studied the resilience behavior in the face of physical damage [10].

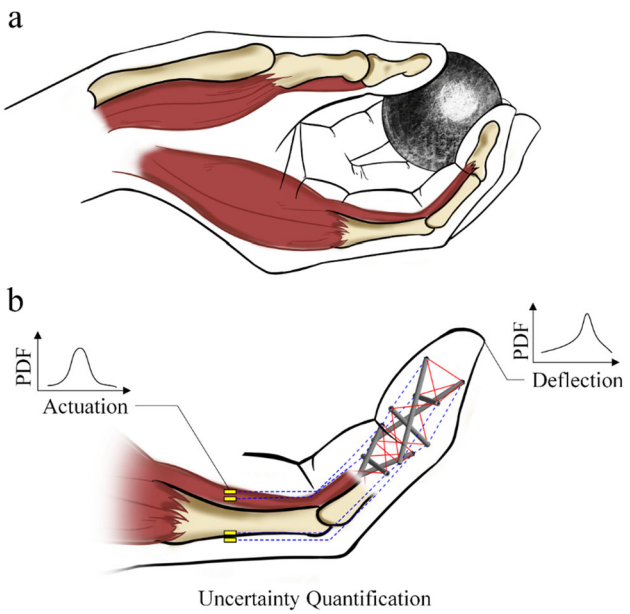

Fig. 1 Sketch of clustered tensegrity fingers for capturing

However, these studies mentioned above are limited to the classical tensegrity. The actuation design for a clustered tensegrity with a prescribed function has been hardly seen so far. One of the important reasons may be attributed to the unstable convergence of numerical analysis, which prevents the optimization design based on finite element analysis. In addition, there are many uncertain factors, such as uncertainties of material property, load, boundary and interaction in real engineering [27–29]. Uncertainty quantification and propagation in tensegrity structures were never studied before. Quantifying these uncertainties how to propagate in a clustered tensegrity structure could find an application in the design of tensegrity fingers for capturing, which is shown in

Fig. 1, in which PDF stands for probability density function and the mechanism is actuated by shortening the long cable.

This paper aims at developing a novel data-driven computational approach for both uncertainty quantification and optimization designs of actuated clustered tensegrity structures. The rest of this paper is organized as follows. In Sect. 2, we introduce a co-rotational finite element formulation of clustered tensegrity structures, and the numerical results are discussed. In Sect. 3, the data-driven computational framework is presented in detail, and an example of a clustered beam is employed to illustrate the application. The proposed computational framework is a comprehensive one, which includes a series of knowledge-based components and technologies, such as the design of experiments, uncertainty propagation, surrogate models, sensitivity analysis, and real-time prediction. As an application of the proposed model, subsequently, an optimization design of the clustered tensegrity beam based on the surrogate model is discussed in Sect. 4. Finally, we conclude the paper with some remarks and perspectives.

2 Co-rotational finite element formulation

2.1 Tangent stiffness matrix

Tensegrity structures could include clustered (continuous) cables (as shown in Fig. 2a) or not (as shown in Fig. 2b). The structure including clustered cables is called clustered tensegrity, whereas the other is called classical tensegrity. While static analysis of clustered tensegrity is considered, three basic assumptions are usually adopted to simplify the model as follows [12, 14, 16],

- Pulleys are frictionless.
- All the loads are applied at pin-joints.
- The displacement is large, but the strain is such small that a linear stress–strain relation can be used.

Since pulleys are frictionless, each sub-element in one clustered cable (which can be regarded as a super element) have the same internal force. By the co-rotational approach, the tangent stiffness matrix of a clustered cable can be derived by variation calculation for the internal force vector. The formulation was proposed by the authors before [16]. Here we present it directly. For a clustered cable including n sub-elements, the stiffness matrix can be given by

$$\mathbf{K}_e = \tilde{\mathbf{K}}_{t1} + \tilde{\tilde{\mathbf{K}}}_{t1} + \tilde{\mathbf{K}}_{t2}, \tag{1}$$

$$\tilde{\mathbf{K}}_{t1} = \sum_{ei=1}^{n} \mathbf{T}_{ei}^{\mathrm{T}} \mathbf{K}_l \mathbf{T}_{ei}, \tag{2}$$

$$\tilde{\tilde{\mathbf{K}}}_{t1} = \sum_{ei=1}^{n} \sum_{ej=1}^{n} \mathbf{T}_{ei}^{\mathrm{T}} \mathbf{K}_l \mathbf{T}_{ej} \ \ (ei \neq ej), \tag{3}$$

$$\tilde{\mathbf{K}}_{t2} = \frac{A_0 \sigma}{l_e} \left[\mathbf{A} - \frac{1}{l_e^2} \mathbf{c}_e(x') \mathbf{c}_e(x')^T \right]. \tag{4}$$

In Eqs. 2–4, the subscript ei and ej denote the sub-elements included in the super element e. $\mathbf{T}$ is a transformation matrix consisting of direction cosines and sines. $\mathbf{K}_l$ is the local stiffness matrix of each sub-element, $\mathbf{A}$ is a constant matrix $\begin{bmatrix} \mathbf{I}_{3\times3} & -\mathbf{I}_{3\times3} \\ -\mathbf{I}_{3\times3} & \mathbf{I}_{3\times3} \end{bmatrix}$ and $\mathbf{c}_e(x')$ is an elementwise direction cosine vector. A, l, E and σ are the area of cross-section, rest length, Young's modulus, and engineering stress of a cable, respectively. Other matrices and vectors can be found in the authors' previous work [16]. In Eq. 1, the first part $\tilde{\mathbf{K}}_{t1}$ denotes the material stiffness of the n sub-elements. The second part $\tilde{\tilde{\mathbf{K}}}_{t1}$ describes the stiffness effect of one sub-element on others, and it is induced by the relative sliding between clustered cables and pulleys. The third part $\tilde{\mathbf{K}}_{t2}$ is the stress stiffness that is included in a conventional geometrically nonlinear formulation. It should be pointed out that if the second part is removed from Eq. 1, $\mathbf{K}_e$ will change to be the stiffness matrix of n classical (short) cables connected by pin-joints.

The nodal internal force of a super element e (a clustered cable) is given by

$$\mathbf{f}_e = \sum_{ei=1}^{n} \mathbf{T}_{ei}^{\mathrm{T}} \mathbf{f}_{ei}. \tag{5}$$

$\mathbf{f}_{ei}$ is the internal force vector of each sub-element. With the tangent stiffness matrix and internal force vector, nonlinear finite element analysis can be conducted for a clustered tensegrity structure by the Newton–Raphson scheme. Note that all the cables have no resistance to compression, and it must be dealt with in the computer program.

2.2 Actuation and response of a clustered tensegrity beam

A clustered tensegrity beam is selected as an example in the paper to illustrate the application of a data-driven computational approach. As shown in Fig. 3, the structure includes 4 clustered cables, 32 short cables, and 12 struts. The structure is self-equilibrated by prestress that is introduced by shortening the two top clustered cables by 2.55% of the rest length. To deform the structure, actuations are applied to four clustered cables. The actuation is assumed to be quasi-static, and thus the inertial effect is neglected. Numerical simulation is carried out by using the co-rotational (CR) finite element formulation. The two top clustered cables are shortened by

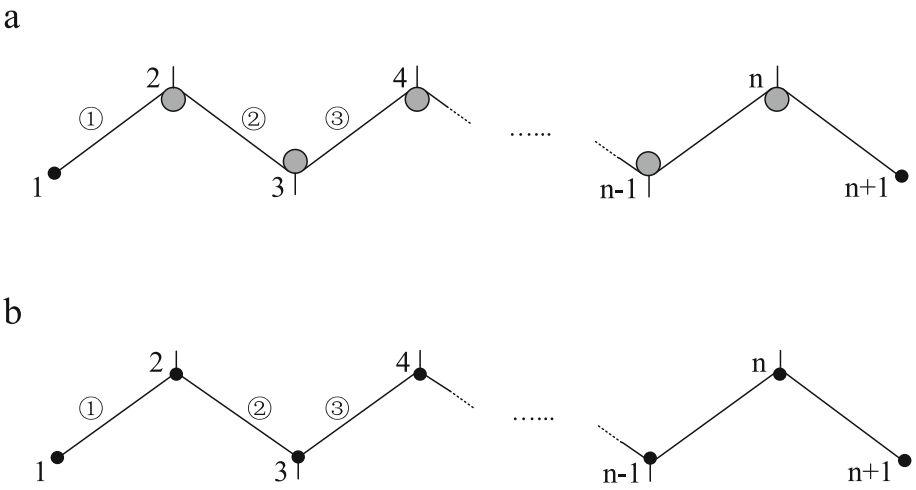

Fig. 2 Clustered and classical cables with n + 1 nodes: **a** The clustered cable grouped by n cable elements; **b** The classical pin-joint structure composed of n cable elements

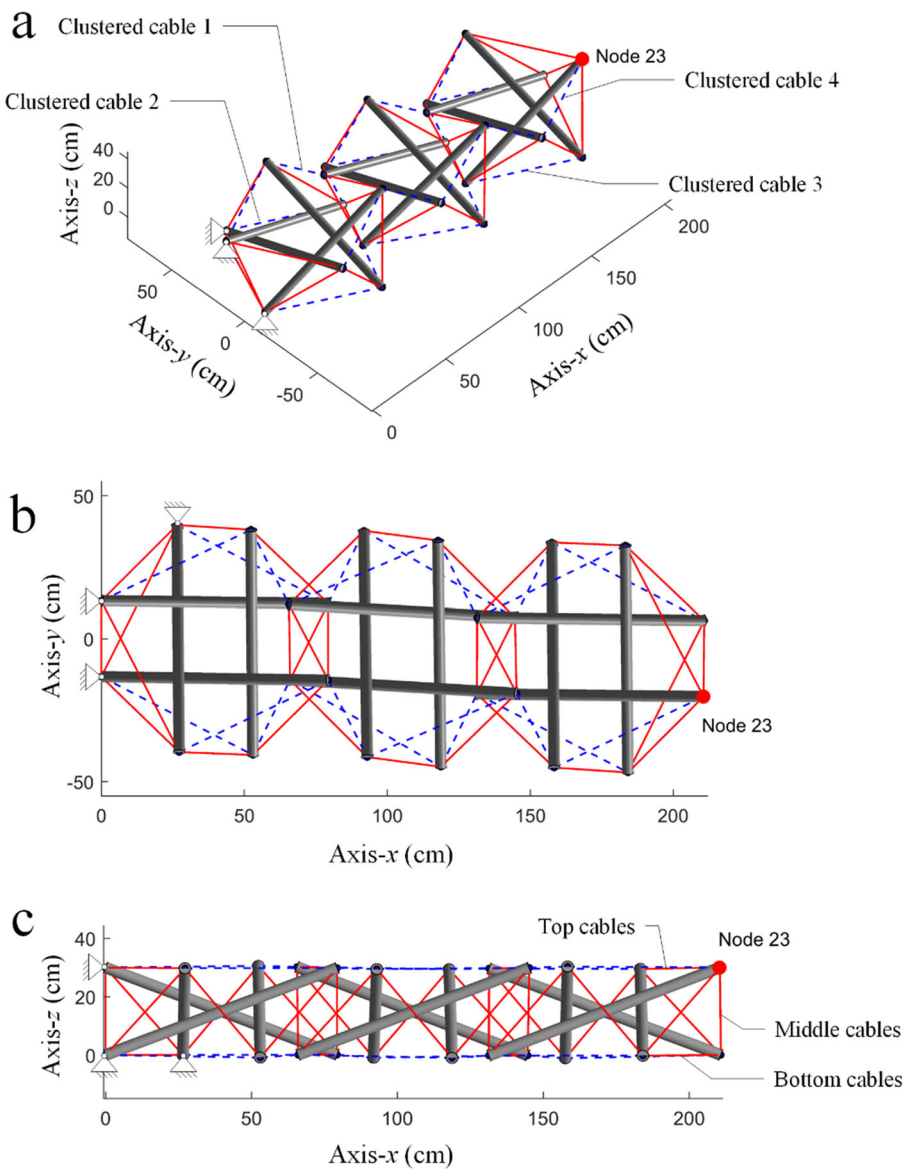

Fig. 3 A clustered tensegrity beam including four clustered cables: **a** A perspective view; **b** A top view; **c** A side view

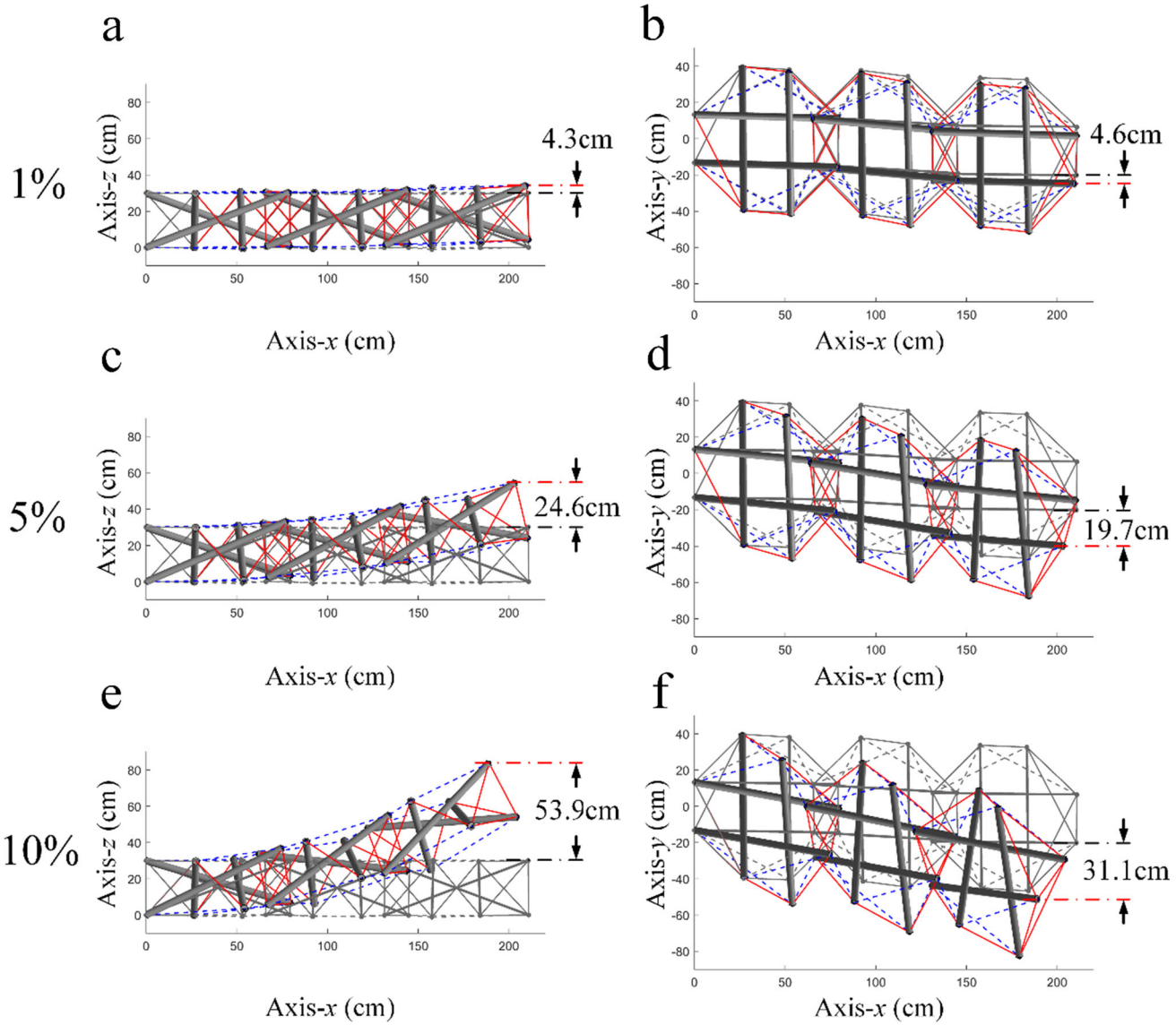

Fig. 4 Deformed shapes of the clustered tensegrity beam under different actuations: **a**, **c** and **e** A side view; (**b**), **d** and **f** A top view

10%, while the two bottom ones are extended by 10%. The actuation is implemented gradually by an increment of 1%. Three snapshots of deformed shapes produced by 1%, 5%, and 10% actuations are plotted in Fig. 4. A large deflection occurs along axis-z when the actuation is up to 10%. Besides, a deflection along axis-y also becomes larger and larger with the increase of actuation.

Unfortunately, the FEA of the clustered tensegrity often encounters the problem of convergence, which is caused by mechanism modes [12]. FEA of the clustered tensegrity beam (shown in Fig. 3) could get multiple solutions, as shown in Fig. 5. Figure 5a shows an upward deflection that is desired for the application of a flexible manipulator, while Fig. 5b shows an irregular deformed configuration that cannot be used. The two configurations are obtained by setting different lengths of loading steps.

Further, we try on large amounts of simulations for a thorough investigation. In the first group of simulations, gravity is considered. 100,000 actuations are changed randomly within the interval [1%,1 0%]. The top two clustered cables have the same actuation A_14, while the bottom two have A_23. A statistical pie chart of the first group of simulations is plotted in Fig. 6a. It can be found that the result of an upward deflection takes up 98.4%, the irregular deformation 1.4%, and the unconvergence only 0.2%. In the second group, gravity is neglected. 100,000 actuation combinations for the four independent clustered cables, (A_1, A_2, A_3, and A_4), are randomly drawn from a uniform distribution within the interval of [1%,30%]. The statistical pie chart is plotted in

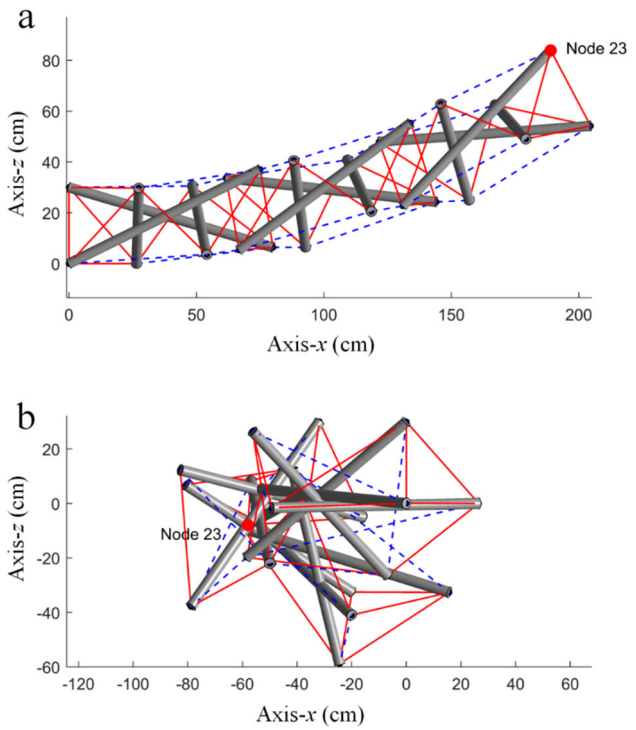

Fig. 5 Deformed configuration produced by actuations: **a** A stable structure; **b** A collapsed irregular structure

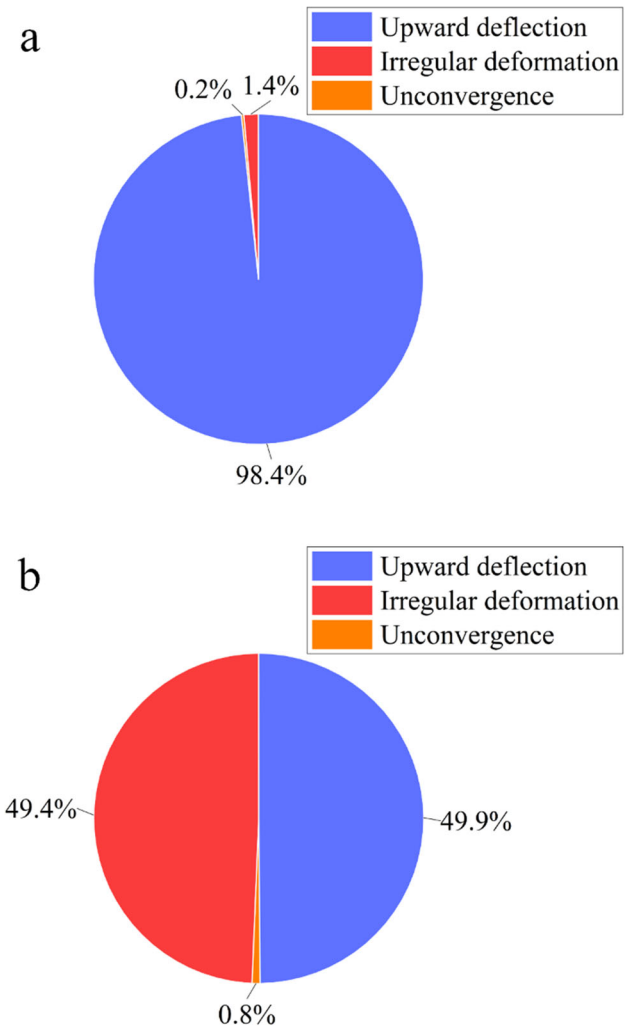

Fig. 6 Statistical pie chart of 100,000 FEM simulations: **a** With gravity; **b** Without gravity

Fig. 6b. It can be seen that about half of the simulations produce an upward deflection, and irregular deformation takes up 49.4%. To overcome such difficulty an effective surrogate model is needed. Bayesian method has its advantages because they deal with noisy observations by inferring a smooth average response and quantifying uncertainty[25]. Without losing generality, the neural network model as an example of another way of building a surrogate model has also been purposed and discussed.

In addition, the unstable convergence prevents the optimization design of actuation based on FEM, in that the finite element equilibrium equation needs to be solved within each iteration to update the objective function. This problem will be solved later in this paper, by proposing an efficient data-driven computational approach that is available for the optimization of actuation.

3 Data-driven computational framework

A comprehensive data-driven computational approach is proposed for analysis and design of clustered tensegrity structures in this section. It includes three parts, which are the design of experiments, surrogate models, and uncertainty propagation. A flowchart of the framework is plotted in Fig. 7. Next, each part will be explained in detail.

3.1 Design of experiments

The design variables are the actuation of clustered cables and temperature in classical cables. As shown in Fig. 3a, the four clustered cables can be divided into two groups. The first group includes the two clustered cables (No. 1 and No. 4) on the top surface, while the second group includes the two (No. 2 and No. 3) on the bottom. Actuations are denoted by A_14 and A_23, respectively. Note that clustered cables No. 1 and No. 4 are always shortened, while No. 2 and No. 3 are always extended. Temperature loads are divided into three groups. The temperature load applied to classical cables on the top surface is denoted by T_top. Similarly, T_bottom is for classical cables on the bottom surface, while T_middle is for the cables between the top and bottom surfaces. The input variable $\mathbf{x}$ is five-dimensional, i.e., $\mathbf{x} = [\text{A\_ 14, A\_ 23, T\_ top, T\_ middle, T\_ bottom}]$. The range of actuation is $[9\%, 10\%]$, and the temperature changes

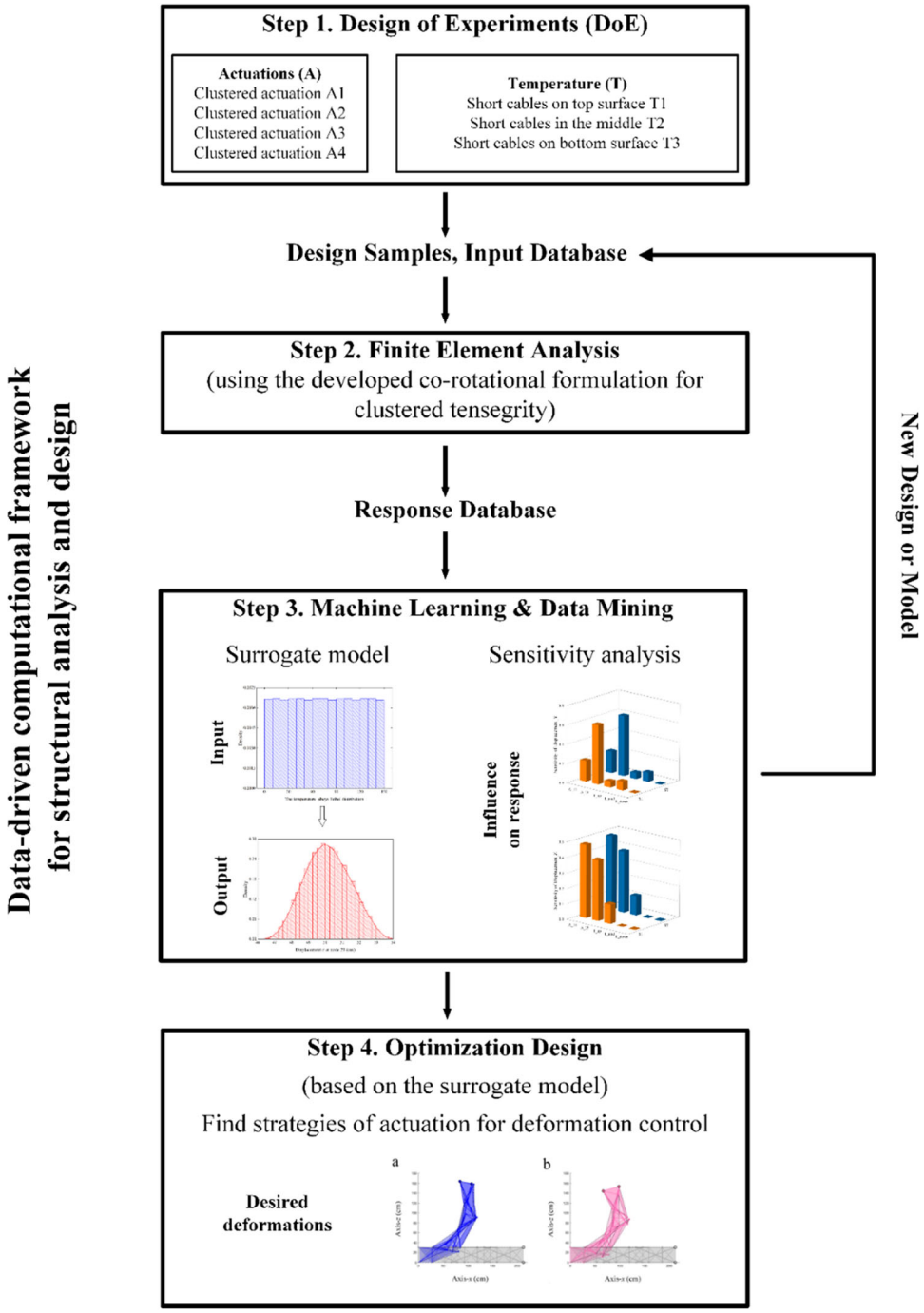

Fig. 7 Flow chart of the data-driven computational approach

within [0, 150°C]. The dataset of input variables is designed as $A \times T = 100 \times 125 = 12,500$. Without any prior knowledge of the problem, space-filling designs treating all the input variable domains uniformly are the first choice [30]. Among all the sampling methods, Latin Hypercube Sampling and the Sobol sequence are widely adopted because of their high efficiency and ease of use [31, 32]. Here we employ the Sobol sequence to produce the input data. Figure 8 shows DoEs for the two-dimensional actuation variables and three-dimensional temperature variables obtained by the Sobol sequence within the bounds given before. After the space of input design variables is defined, the next step is to collect its related quantities of interests (QoIs), $\mathbf{q}$. The data pairs and $\mathbf{q}$, $\left[\left(\mathbf{x}^1, \mathbf{q}^1\right), \left(\mathbf{x}^2, \mathbf{q}^2\right), \ldots, \left(\mathbf{x}^N, \mathbf{q}^N\right)\right]$ create the database that is required in the training of machine learning methods. N is the total number of samples in the design space. In this study,

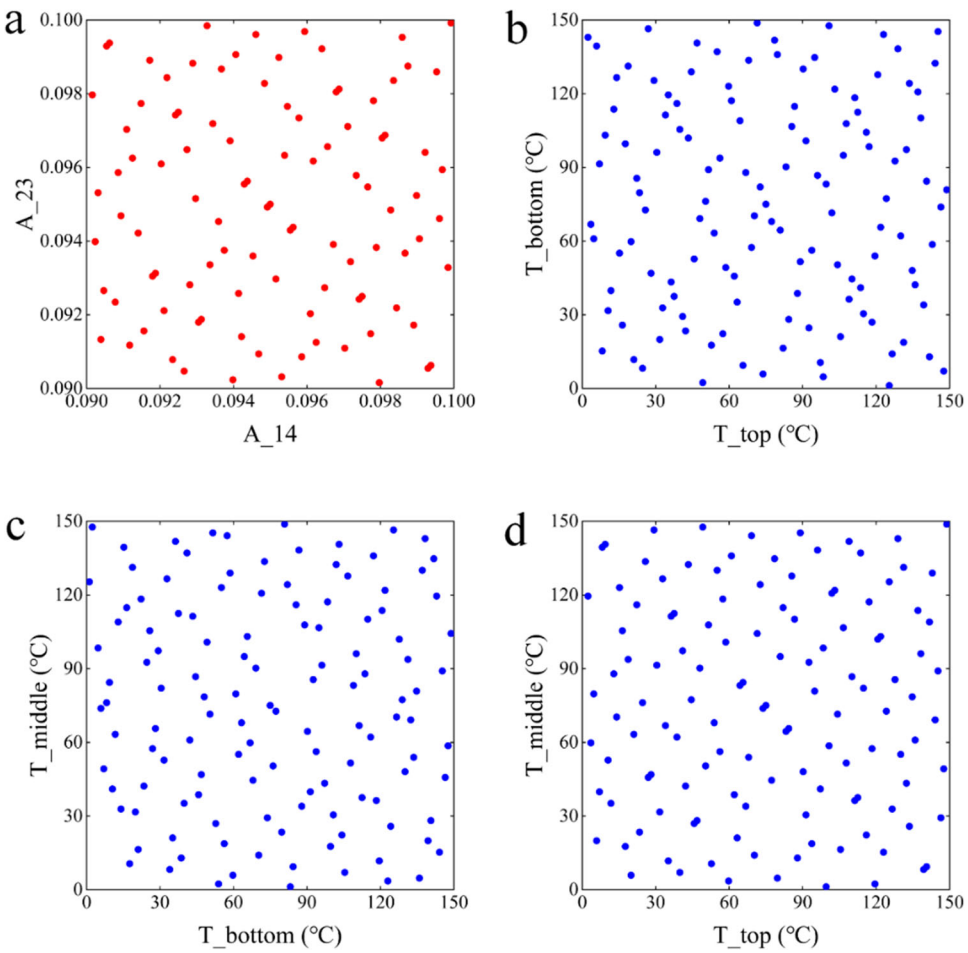

Fig. 8 Design of experiments with 100 actuations and 125 temperatures, produced by Sobol sequence: **a** A_ 14 and A_ 23; **b** T_top and T_bottom; **c** T_bottom and T_middle; **d** T_top and T_middle

we focus on the displacements at the free end (the displacements at node 23) of the tensegrity beam, with $\mathbf{q} = [u, v, w]$.

12,500 simulations are carried out. The material property of cables and bars are presented in Table 1. One simulation costs about 0.3 s, and the preparation of input dataset costs about 1 h. The uncertainty of deformation (the displacement at node 23) is shown in Fig. 9. For clarity, the states of deformation are sorted in ascending order by the magnitude of displacement. In the plot, a box plot is used, where the central mark is the median of quantity. It shows the 25^{th} and 75^{th} percentiles with the box edge, and the top and bottom dash line represent the maximum and minimum. Each box is obtained by the combination of a unique actuation and different temperatures. In summary, Fig. 9 indicates that the effect of uncertainty of temperature (T_top) on deflection-z is high, while the effect of temperature on deflection-y is negligible.

3.2 Surrogate models

Uncertainty quantification and propagation for a problem with high dimensionality require a huge amount of computation on DoEs. However, conventional FEA is too expensive for such stochastic study which usually involves thousand to million of simulations. Machine Learning methods allow building efficient surrogate models by using a predetermined DoEs analysis dataset. Even though the training dataset obtained by DoE analysis is still relatively large, it needs to be prepared only once, and the constructed surrogate model can be used for prediction and optimization design. For the surrogate models, the inputs are actuations of four long cables and the output is the deflection of the tensegrity beam (average displacement-z at nodes 18 and 23). In this paper, both Gaussian Process Regression (GPR) [33] and Neural Network (NN) [34] are employed to construct surrogate models, and their performance is compared.

To build the surrogate models, the 12,500 FEM simulations are separated into a 10,000 sample training set and a 2,500 sample validation set.

Table 1 Material property of members

Member	Material	Cross-sectiosn area(cm^2)	Young's modulus(kN/cm^2)	Weight(kN/cm^3)
Struts	Aluminum	255	7000	2.7×10^{-5}
Cables	Stainless-steel	0.5026	11,500	7.85×10^{-5}

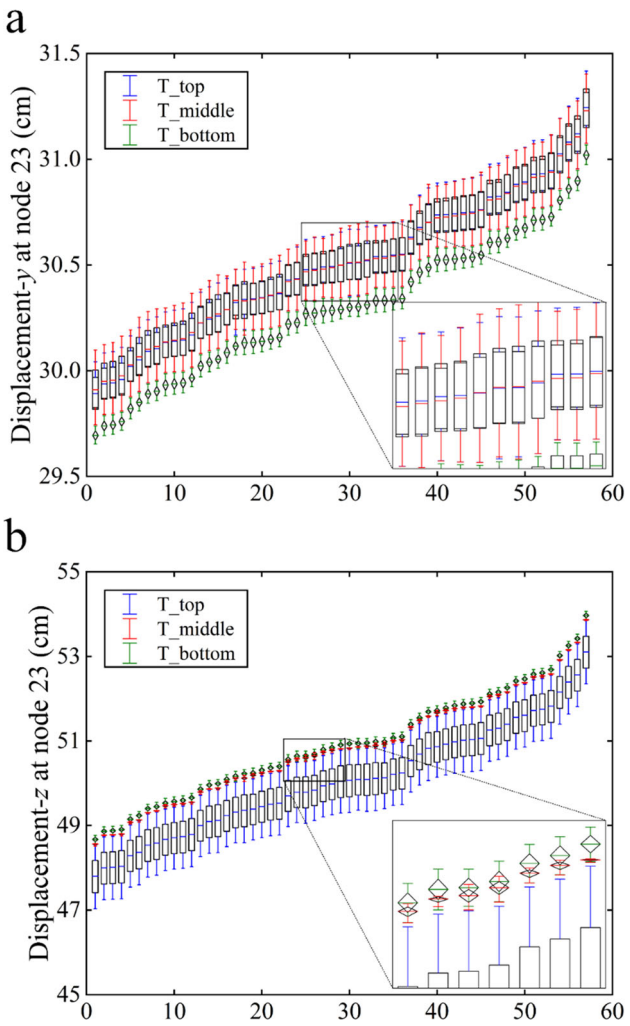

Fig. 9 Uncertainty quantification of displacements subjected to temperature variation: **a** displacement-y; **b** displacement-z

Gaussian Process Regression (GPR) is a type of Bayesian non-parametric regression method that allows flexible representation of the posterior distribution of independent variables by utilizing Gaussian Processes (GPs) which are distributions over functions [32], such that

$$f \sim GP(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}\prime)). \tag{6}$$

$\mu : R^d \rightarrow R$ is the mean function and $k : R^d \times R^d \rightarrow R$ is the covariance function. In this paper, k is chosen to be the Radial Basis Function (RBF) kernel. GPR significantly contributes toward modeling various problems flexibly. Here a separate GPR is trained for each output-dependent variable. However, the computation is expensive for a large-scale problem.

Neural Network (NN) maps the input vector $\mathbf{x}$ into output $\mathbf{y}$, through a number of linear transformations. The mapping is achieved through the activation function $\sigma(Wx+b)$, where are the node weight vector and b the offset bias. Both must be found during the learning process. Interested readers are referred to the literature [35] for the details. In this study, a Neural Network with 2 hidden layers, each consisting of 50 neurons, is constructed. Mean square error is selected as the loss function. A two-stage optimization strategy is adopted for training. The Neural Network model is firstly trained with the Adam optimization method [36] for 5,000 iterations. In the second stage, a derivative-based L-BFGS optimization is used to further improve the accuracy of model fitting [37].

Relative errors such as E and R^2 are selected as the error metrics to estimate the accuracy of the two methods. Specifically, they are calculated as

$$E_{q_i} = \frac{1}{N}\sum_{1}^{N}\frac{|\hat{q}_i - q_i|}{|q_i|}, \tag{7}$$

$$R^2_{q_i} = \frac{\sum_{1}^{N}\left(\hat{q}_i - \overline{q}_i\right)^2}{\sum_{1}^{N}\left(q_i - \hat{q}_i\right)^2}. \tag{8}$$

where $q_i, \overline{q}_i$ is the quantity of interests and its mean, respectively, while $\hat{q}_i$ is estimated by the surrogate model. E shows the relative error between FEM and surrogate models, while assessing how strong the linear relationship is. Figure 10 presents the errors of displacement-y and displacement-z at node 23 obtained by the two methods, using different sizes of DoEs. It can be found that the errors are both very small as the number of training data increases to 1,000. The relative error of GPR reaches around $1 \times 10^{-5}\%$, while NN has an error about $1 \times 10^{-2}\%$. Figure 11 shows the comparison between FEM and surrogate models trained by 1,000 samples. GPR has $R^2 = 0.9999$ for both displacement-y and -z, while NN has $R^2 = 0.9932$ and $R^2 = 0.9989$ respectively for displacement-y and -z. It is concluded that GPR has better accuracy than NN when they are both used for a

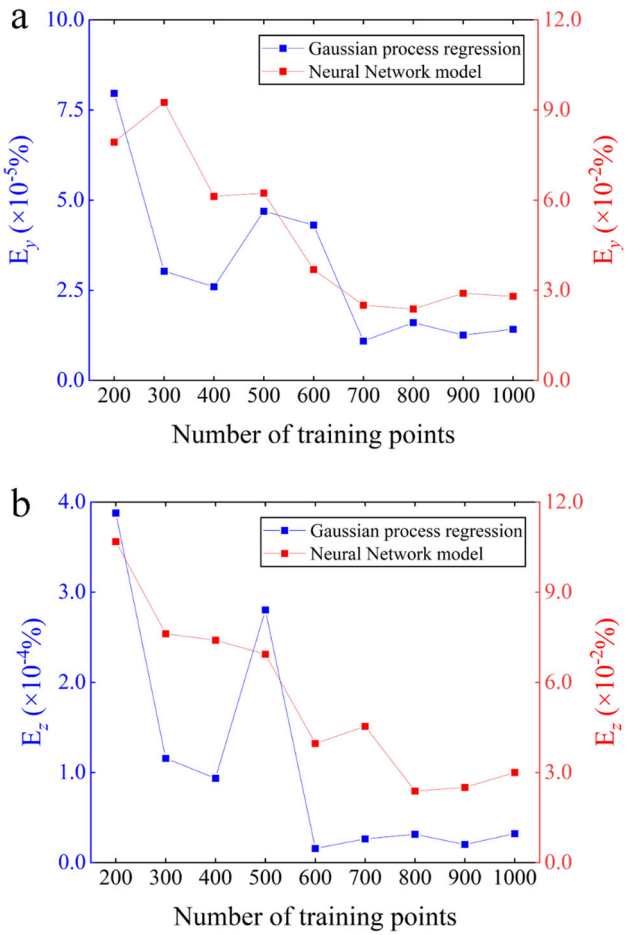

Fig. 10 Error of displacements predicted by Gaussian Process Regression and Neural Network: **a** displacement-*y*; **b** displacement-*z*

small training dataset. Nevertheless, the accuracy of NN is still enough high for an application in engineering. However, to train 1,000 samples costs 147.3 s by NN, while GPR costs 630.3 s. When the training samples increase to 100,000, NN costs about 1,000 s, while GPR costs several hours. In addition, the inference time to predict 100,000 new points by NN is only about 3.0 ms which is faster than GPR by three orders of magnitude. The prediction by NN is real-time, and thus it is adopted for the optimization design in the next section.

3.3 Uncertainty propagation

The trained surrogate model can be used to quantify the uncertainty of response (uncertainty propagation) with Monte Carlo simulation. Monte Carlo simulation for uncertainty propagation calculates the desired quantity by randomly drawing from the specified uncertainty distributions of input variables. A_14 and A_23 are set to be 10%, and assumed to suffer from a random mechanical error ϕ, i.e.,

$$\text{A\_14} = \text{A\_23} = 0.1 - \phi. \tag{9}$$

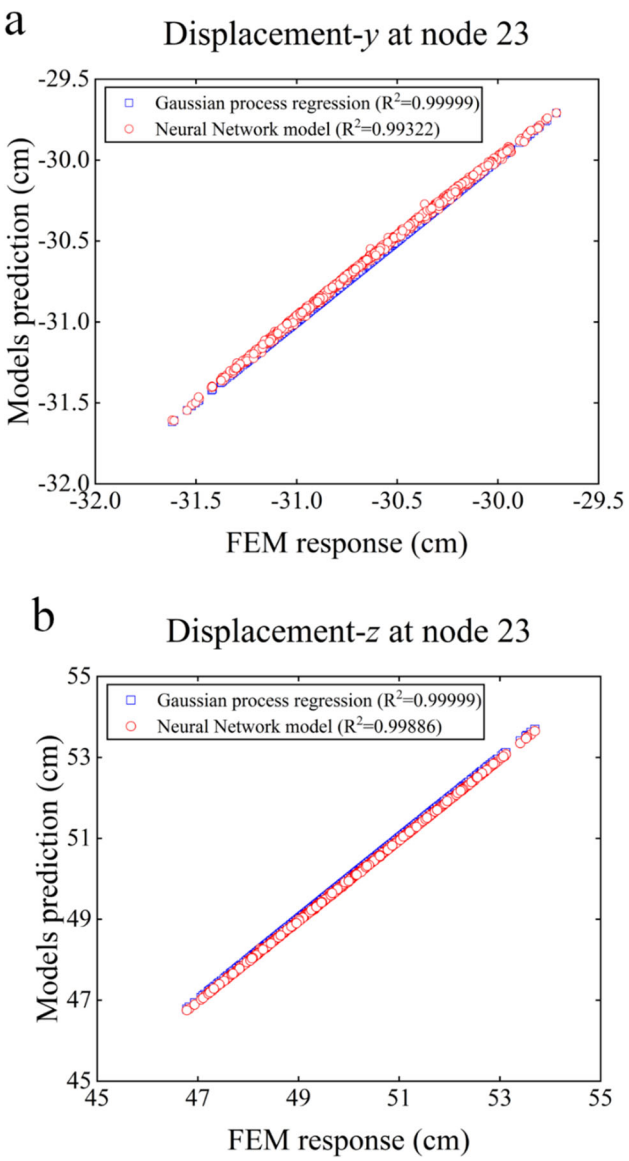

Fig. 11 Comparison of results obtained by surrogate models and FEM: **a** displacement-*y*; **b** displacement-*z*

ϕ obeys two random distributions: (a) a Sobol distribution within [0,1%,]; (b) an exponential distribution with an expectation of 0.001. In the two cases, the probability distribution functions (PDF) of actuation A_14 (A_23) can be plotted in Fig. 12a and b, respectively. QoI is the displacement along direction-*z* at node 23. The trained NN surrogate model in previous section is employed to complete the prediction of QoI. The histogram and its related kernel density estimation are shown in Fig. 12c and d. The results are accumulated by using 10,000 new samples for input variable distribution. QoI presents a normal distribution (Fig. 12c) when the actuation obeys a Sobol distribution (Fig. 12a); QoI presents an approximately exponential distribution (Fig. 12d) when the actuation obeys an exponential distribution (Fig. 12b). The

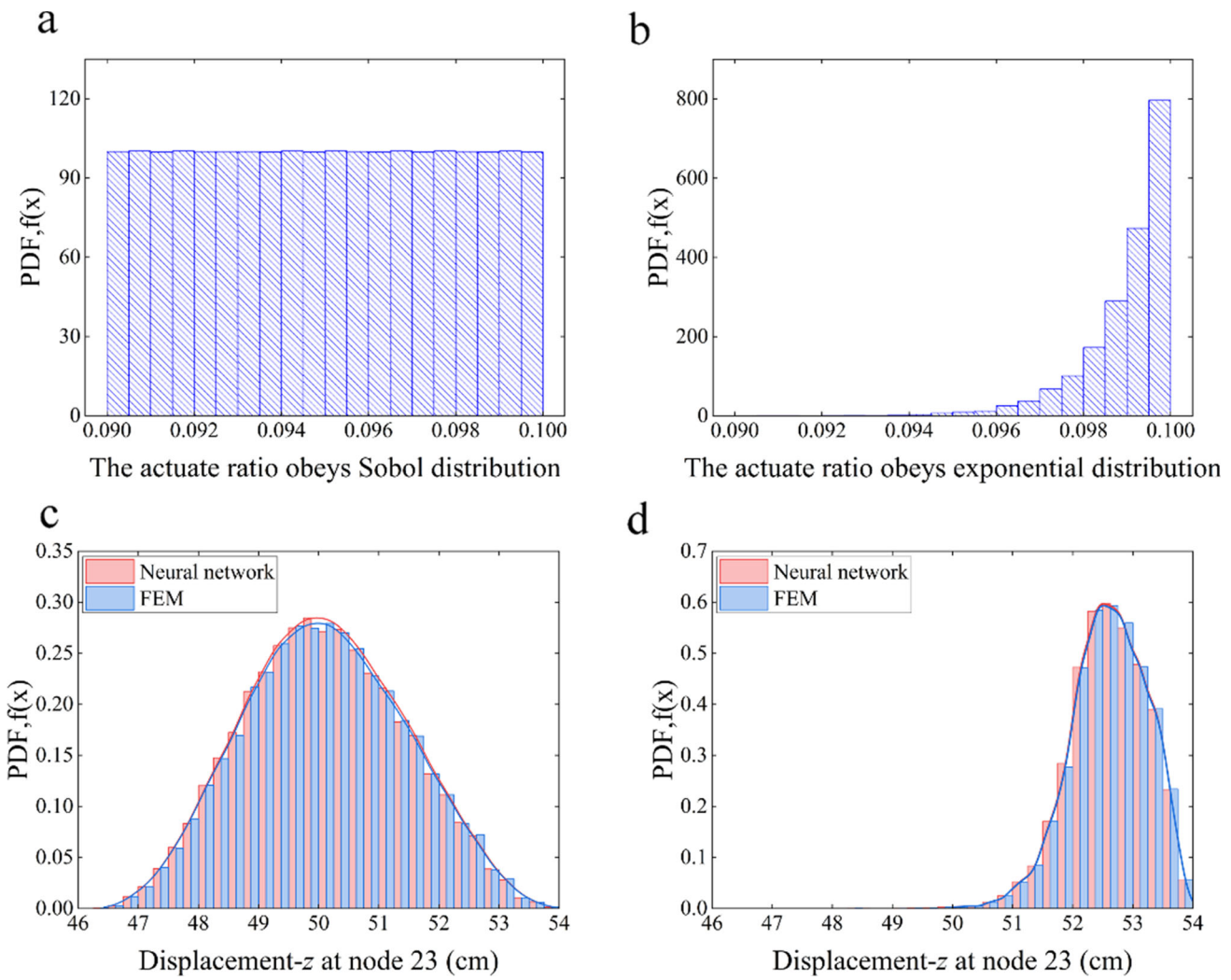

Fig. 12 Uncertainty propagation in the clustered tensegrity beam predicted by the Neural Network surrogate model: **a** PDF of actuation obeying a Sobol distribution; **b** PDF of actuation obeying an exponential distribution; **c** PDF of displacement-z obtained by case (**a**); **d** PDF of displacement-z obtained by case (**b**)

Table 2 Statistics of displacement-z under different actuation distributions

Statistical values (cm)	Sobol distribution		Exponential distribution	
	FEM	Neural network	FEM	Neural network
Mean	52.60	52.60	50.10	50.11
Std	0.64	0.64	1.34	1.33
Min	48.48	48.50	46.42	46.41
Max	54.03	54.01	53.88	53.88
25%	52.20	52.19	49.15	49.15
50%	52.63	52.63	50.08	50.08
75%	53.08	53.08	51.06	51.06

reference solution is obtained by MCs with the same samples by FEM. Additionally, the statistical results are listed in Table 2, which shows the FEM reference solution and Neural Network prediction are in good agreement. The reference MCs need additional 10,000 FEM simulations, which costs about 2695 s, while the NN surrogate model takes only 3.0 ms.

4 Bayesian optimization for the design of actuation

4.1 Model design

A global sensitivity analysis on the four actuations (A_1, A_2, A_3, and A_4, each acting on the related clustered cable independently) is carried out to determine the key variables before the optimization design. Interested readers are referred to Appendix A for the implementation of global sensitivity analysis. Here, we just present the result. Figure 13 plots first order and total sensitivity indices calculated by 10,000 FEM simulations whose DoEs are obtained by Sobol sequence for four actuations within the bound [0,30%]. It indicates that actuations to the two clustered cables (A_1 and A_4) on the top surfaces have much stronger effects on the displacement-z than the two bottom ones (A_2 and A_3). Therefore, actuations to the two top clustered cables are separated, while the two bottom ones are bounded together and denoted by A_23, in consideration that the computational cost increases exponentially with the dimension of design variables.

Subsequently, 100,000 simulations by FEM sampled with Sobol sequence in the 3-dimensional (A_1, A_23, and A_4 within the bound [0,30%]) input variable space are carried out. The converged results of the simulation are divided into groups: (a) an upward deflection and (b) an irregular configuration. The classification of a huge dataset is achieved by the HDBCAN algorithm that is introduced in Appendix B. After the classification, 98,365 data with an upward deflection are used for training to construct a surrogate model by the Neural Network. The training process costs about 1000 s. In the following text, the surrogate model will be used for the optimization design and is denoted by a function,

$$\mathbf{q}=(u,\ v,\ w) = H_{nn}(\mathrm{A\_}1,\ \mathrm{A\_}23,\ \mathrm{A\_}4). \tag{10}$$

4.2 Bayesian optimization

A myriad of optimization methods is available in the literature. Typically, they can be classified into derivative-based or derivative-free optimization methods. Derivative-based methods usually have a faster convergence rate [38, 39]. Derivative-free methods are mainly reserved for applications where direct access to the derivatives of DoIs do not exist or

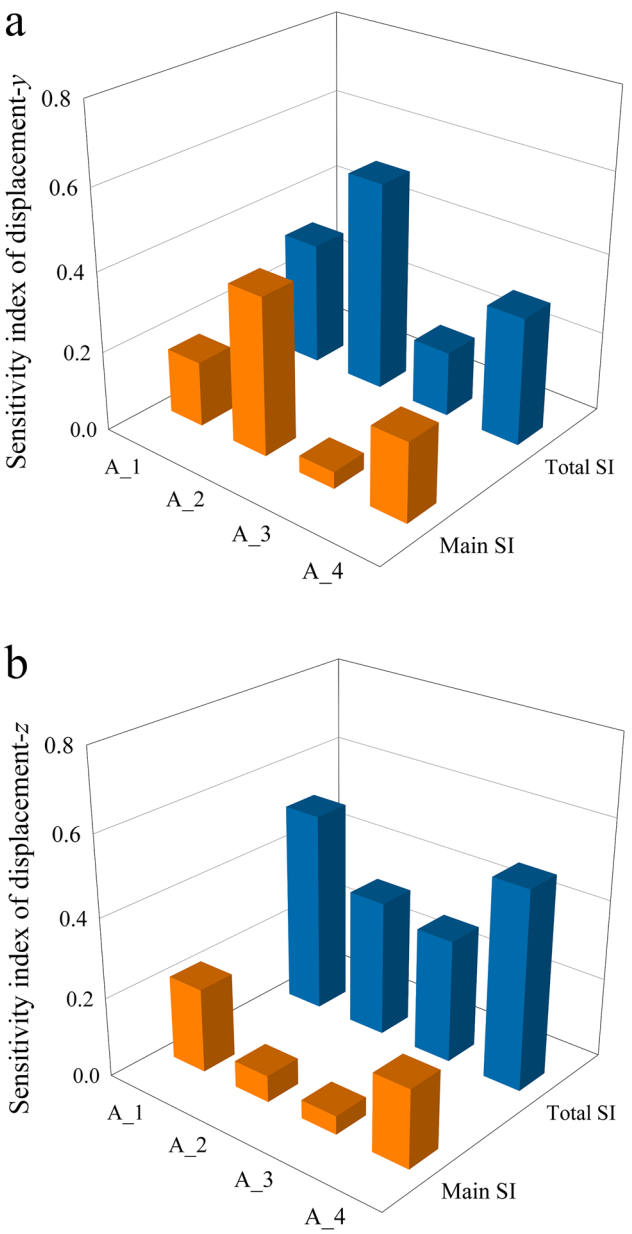

Fig. 13 Sensitivity indices of displacements (Actuation: 4-dimensional): **a** displacement-y; **b** displacement-z

is too time-consuming to compute [40]. Here, we adopt a derivative-free Bayesian optimization method that attempts to find the global optimum within a minimum number of iterations [41]. It incorporates prior belief about f and updates the prior with samples drawn from f to get a posterior that better approximates f. Bayesian optimization also uses an acquisition function that directs sampling to areas where an improvement over the current best observation is likely.

In Bayesian optimization, a prior is incorporated with the objective function. The posterior is cheap to evaluate and is used to propose points in the search space where sampling is likely to yield an improvement. Proposing sampling points in

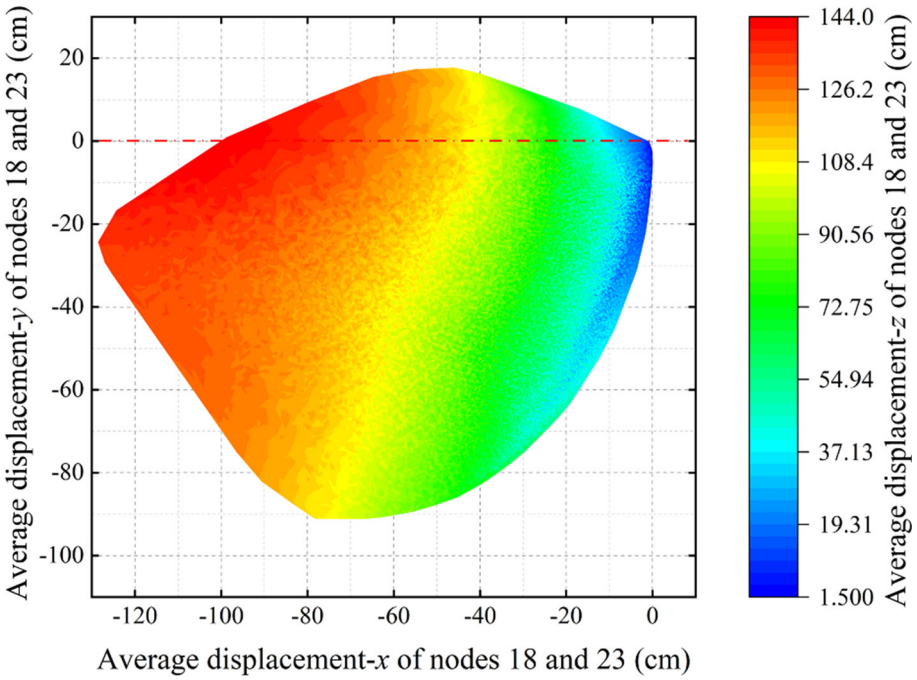

Fig. 14 The range of displacement-z (A_1, A_23 and A_4 within the bound [0, 30%])

the search space is done by acquisition functions. There are many different choices of the acquisition functions, such as the maximum probability of improvement (MPI), expected improvement (EI) and upper confidence bound (UCB) [42]. Here, we use UCB which is most widely used. It is defined as

$$u_{UCB}(x;\,\xi) = \mu(x) + \xi\sigma(x). \tag{11}$$

ξ is a positive trade-off parameter; $\mu(x)$ and $\sigma(x)$ are the mean and standard deviation of the GPR posterior prediction. The first summation term in Eq. (11) is an exploitation term and the second summation term is an exploration term. Exploitation means sampling where the surrogate model predicts a high objective, while exploration means sampling at locations where the prediction uncertainty is high. Exploitation and exploration both correspond to high acquisition function values and the goal is to maximize the acquisition function to determine the next sampling point. The parameter ξ determines the amount of exploration during optimization. The recommended default value of $\xi = 0.01$ is used here.

4.3 Numerical results

The optimization objective is to find an actuation strategy, which would achieve a maximum deflection-z while keeping a minimal deflection-y of the clustered tensegrity beam. The optimization is formulated as follows,

$$\begin{aligned} max.\quad & c\,(\text{A\_ 1, A\_23, A\_ 4}) = \frac{|w|}{|v|+|w|} \\ s.t.\quad & \mathbf{q} = (u,\,v,\,w) = H_{nn}\,(\text{A\_ 1, A\_23, A\_ 4}) \\ & \text{A\_ 1, A\_23, A\_ 4} \;\in [0,\,0.3] \end{aligned} \tag{12}$$

In the equations above, w denotes the average of displacement-z at nodes 18 and 23 (at the free end), and v is the average of displacement-y. The first constraint condition represents the equilibrium, expressed by the trained surrogate model. The second indicates that the actuations are limited to the bound [0, 0.3].

The contour plot interpolated from the training dataset is shown in Fig. 14. It can be found that the maximum average displacement-z at nodes 18 and 23 is about 144 cm, as the displacement-y vanishes. The question is what actuations should be applied? Both the derivative-based optimization(SQP[43] and BFGS[37]) and Bayesian optimization converge to the same result. The optimized displacement-z at the free end is 146.85 cm, while the displacement-y is 1.00 cm. The design variables are A_ 1 = A_23 = 0.3 and A_4 = 0.16.

Comparison of the clustered tensegrity beams before and after optimization is shown in Fig. 15. Compared with Fig. 15c and d, it can be observed that a deflection along direction-y is almost eliminated. Four identical beams can be assembled to form a flexible capturing mechanism, as shown in Fig. 16. The optimized one has a smaller clearance, comparing Fig. 16c and d. After the optimization the opening space formed by the 'fingertips' are greatly reduced. In the final design, the opening gap has been limited to 18.22 cm, and its area has been compressed down to 332 cm^2 from

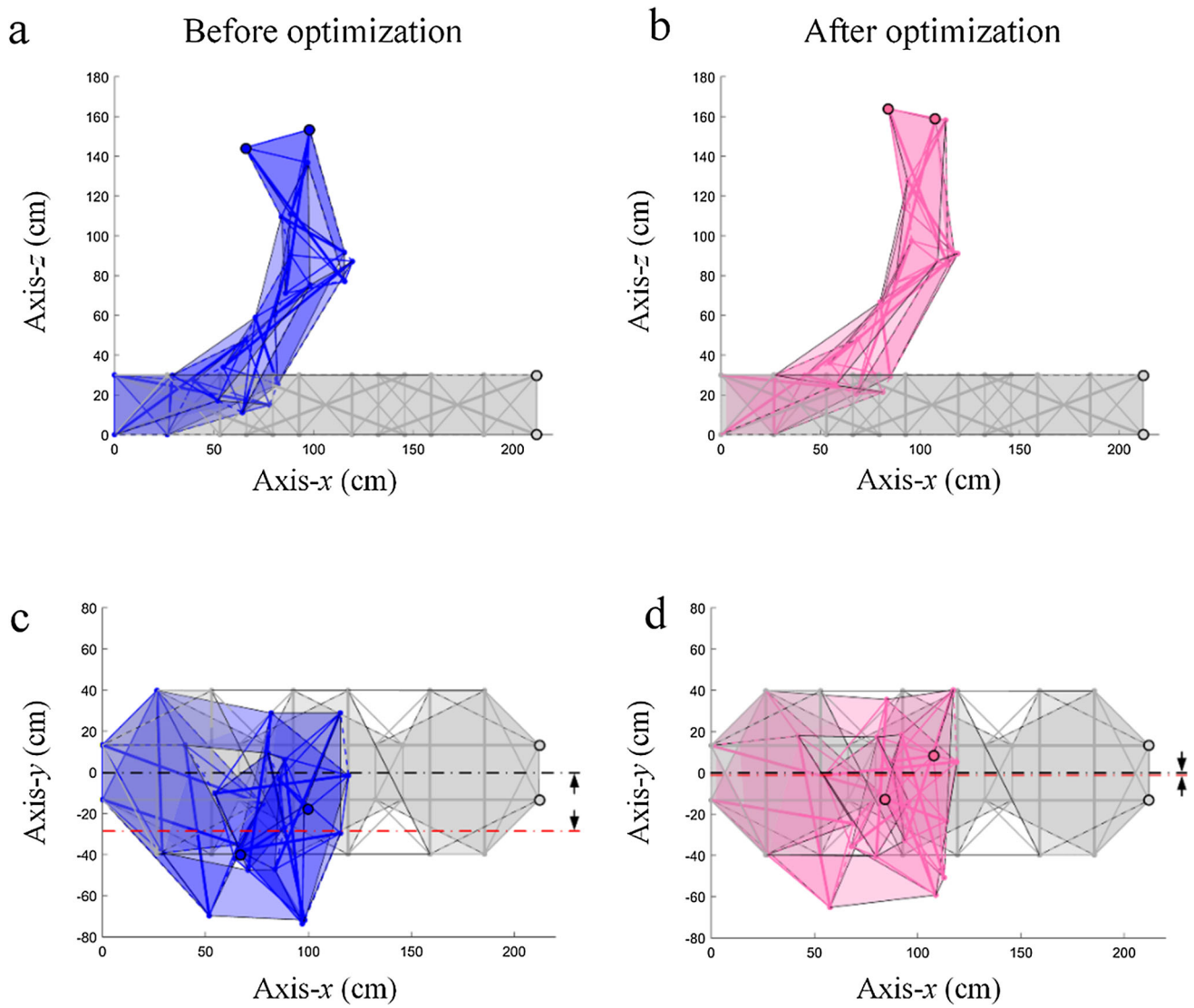

Fig. 15 Comparison of actuated deformations before and after optimization: **a** and **c** A side view; **b** and **d** A top view

2910 cm^2. The optimization result is significant for capturing a smaller object.

Convergence curves of both derivative-base optimization(SQP and BFGS) and Bayesian optimization are plotted in Fig. 17. Bayesian optimization took about 2 min, while, thanks to the efficient neural network model, the derivative-based optimization finished in less than 30 s. The convergence behaviors are quite different. 200 initial value points via the Latin Hypercube sampling are employed to start the multi-start SQP algorithm [43], which takes over 2,000 function evaluations to get the global optimum. On the other hand, Bayesian optimization adopts the upper confidence bound as the optimization criteria. Unlike the random search in SQP, Bayesian optimization efficiently reaches the global optimum via less than 40 function evaluations. In addition, the differences in sampling strategies are clear in their intermediate solution distributions. We sample 200 intermediate solutions in the optimization process and draw the histogram for both methods. As shown in Fig. 18, SQP shows an almost 'uniform' distribution for all local minimums, which is corresponding to the uniform distribution of initial value points. Similar behavior is observed in the BFGS results. In Bayesian optimization, although the exploration strategy built in the acquisition function tends to explore a wider solution space, it shows a clear concentration of sampling around the optimal value of $z = 146.85$ cm.

It needs to be pointed out that even though Bayesian optimization takes fewer function evaluations, but the overall exploration algorithm takes longer time to finish. Benefiting from the efficiency of the neural network model, the derivative-based method, such as SQP and BFGS converged

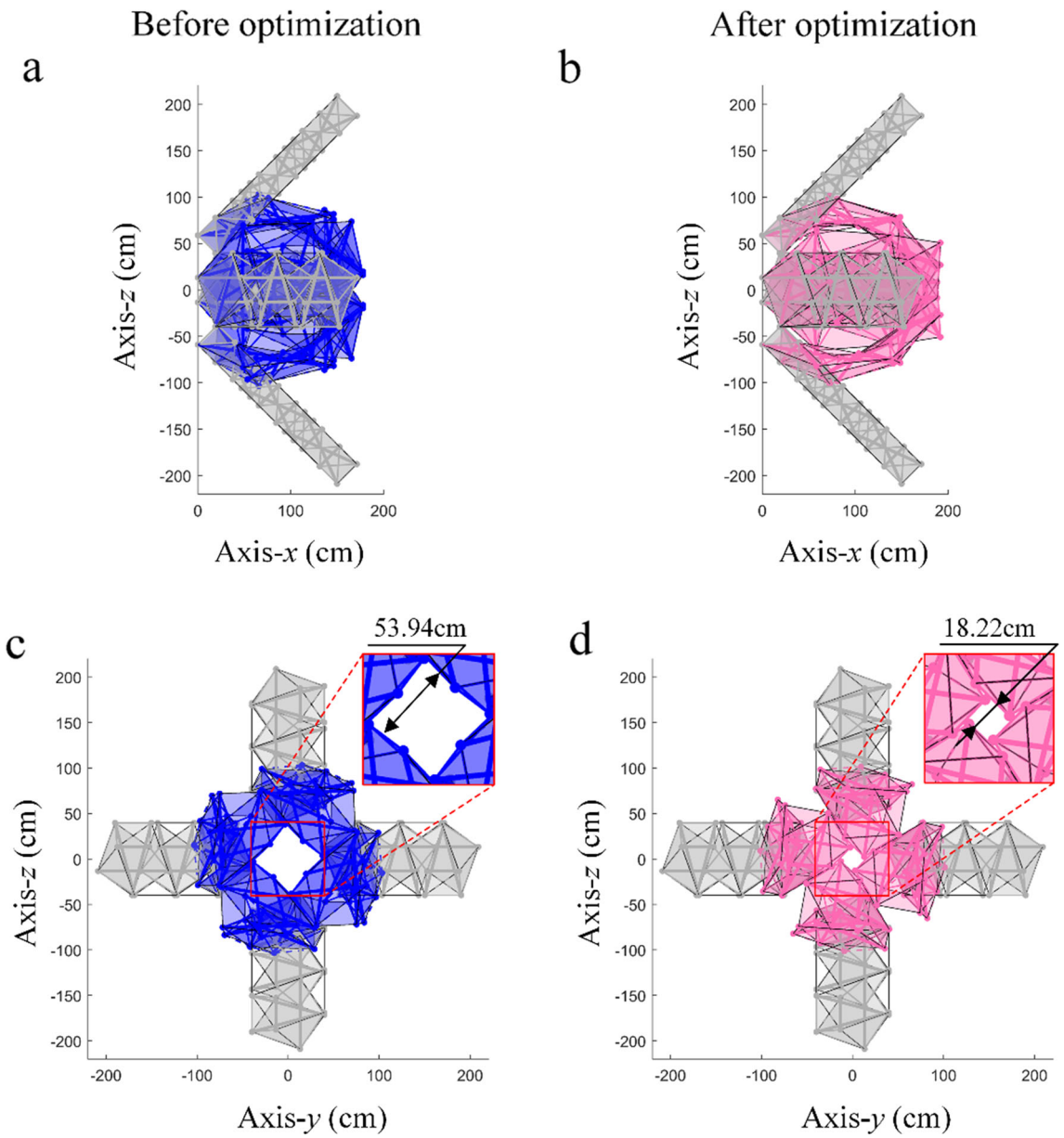

Fig. 16 Capture mechanism consisting of four tensegrity beams before and after optimization: **a** and **c** A side view; **b** and **d** A top view

within 30 s. When processing speed is an important factor, the neural network surrogate model may be a more favored choice.

5 Concluding remarks

This paper presents a data-driven computational approach that overcomes the difficulties encountered by the conventional FEA and promotes the design of actuation to clustered tensegrity structures. Uncertainty propagation of clustered actuation and temperature stress on structural response is investigated in detail by the Monte Carlo method based on an efficient surrogate model. Importantly, a surrogate model-based optimization is carried out by using the Bayesian optimization method that can get the optimized solution efficiently. The result of optimization shows that the actuated deformation can be controlled accurately by adjusting the clustered actuations, and thus the data-driven computational approach could find an application to design of tensegrity-based flexible manipulators. The prediction and optimization based on the surrogate model are both in real time.

Implementation of the proposed data-driven computational approach utilize many statistical tools. Firstly, Sobol sequence is employed to produce input database of DoEs. Secondly, a clustering algorithm is adopted for data selection

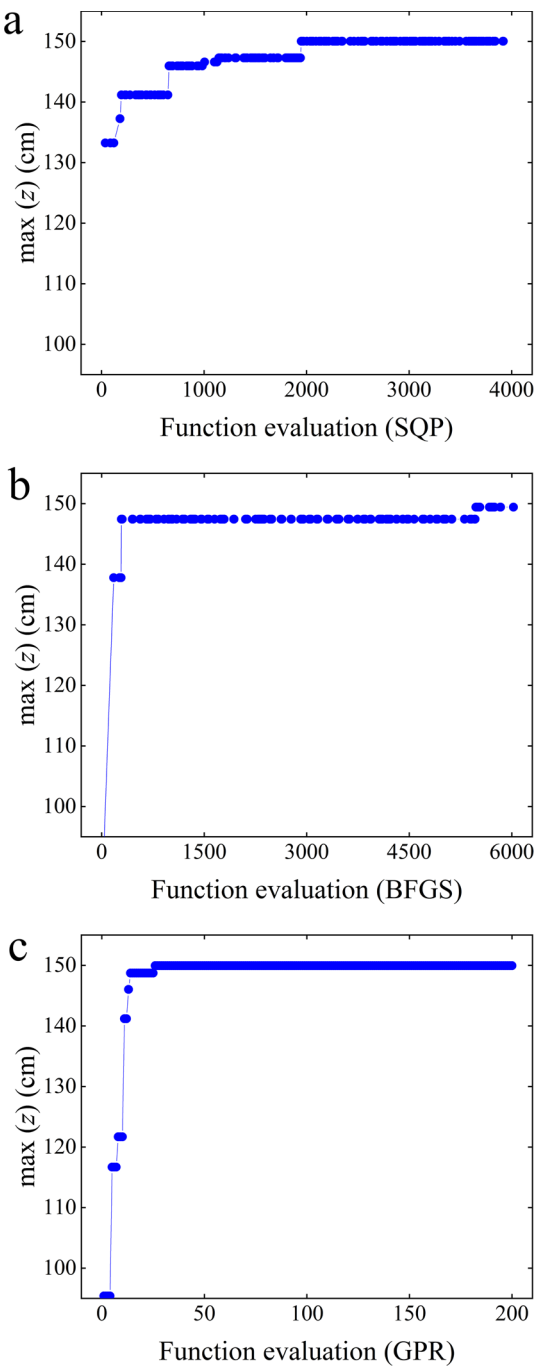

Fig. 17 Convergence curves: **a** SQP; **b** BFGS; **c** Bayesian optimization

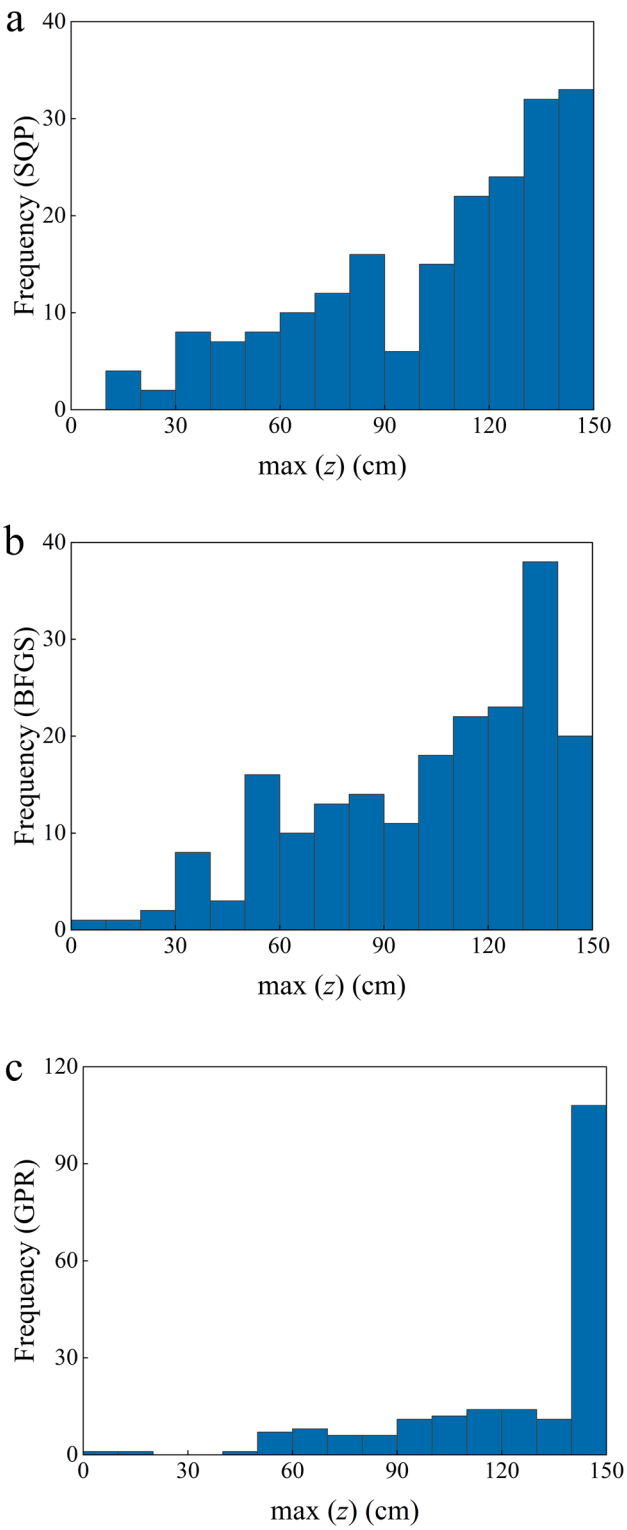

Fig. 18 Histogram of iterative solutions: **a** SQP; **b** BFGS; **c** Bayesian optimization

to distinguish the desired upward-deflected configuration and irregular deformed shape. Thirdly, global sensitivity analysis is used to reduce the dimensions of design variables. Some improvement is worthwhile to be considered in the future, such as.

- The input dataset can be produced with real probabilistic or non-probabilistic models from experiments rather than the Sobol sequence.

- The prestress or physical properties of cables and struts could be considered as the design variables. More complex objective functions or constraints should be embedded into the optimization designs of tensegrity-based manipulators, such as the stability of structures.
- Experiments are necessary for real-time model validation based on the way of Bayesian inference.

Acknowledgements The supports of National Natural Science Foundation of China (Grant No. 12272068) and Center of Space Exploration, Ministry of Education, China (Grant No. SKTC202107)are gratefully acknowledged.

Appendix A

Global sensitivity analysis

Global Sensitivity Analysis (GSA) is employed to quantify the uncertainty of QoIs with respect to input variables, their individual contributions, or the contribution of their interactions. GSA is significant for reorganizing DoEs in the data-driven computational framework and for reducing dimension in optimization design. There are many methods for calculating such index [44–46]. In this example, Sobol's method, a variance-based method is adopted [47]. It helps to get not just the individual parameter's sensitivities but also gives a way to quantify the affect and sensitivity from the interaction between parameters. The main idea for the computation of sensitivity indices is to decompose the function into summands of increasing dimensionality, namely,

$$y = f_0 + \sum_i f_i(x_i) + \sum_i \sum_{i<j} f_{ij}(x_i, x_j)... + f_{1,2...}(x_i, x_i, ...x_k), \tag{13}$$

where k is the number of independent variables. The sensitivity of each input is often represented by a numeric value, called the sensitivity index. The first-order index measuring contribution to the output variance by a single input variable alone can be calculated by

$$S_i = \frac{\mathrm{V}_{x_i}(\mathrm{E}_{x_{\sim i}}(y|x_i)}{\mathrm{V}(y)}, \tag{14}$$

$$\mathrm{V}(y) = \mathrm{V}_{x_i}(\mathrm{E}_{x_{\sim i}}(y|x_i)) + \mathrm{E}_{x_i}(\mathrm{V}_{x_{\sim i}}(y|x_i)). \tag{15}$$

In Eqs. (15) and (16), $\mathrm{V}_{x_i}(\cdot)$ and $\mathrm{E}_{x_i}(\cdot)$ denote variance or mean of argument $(\cdot)$ taken over x_i, respectively, while $\mathrm{V}_{x_{\sim i}}(\cdot)$ and $\mathrm{E}_{x_{\sim i}}(\cdot)$ denote the same quantities taken over all factors except x_i.

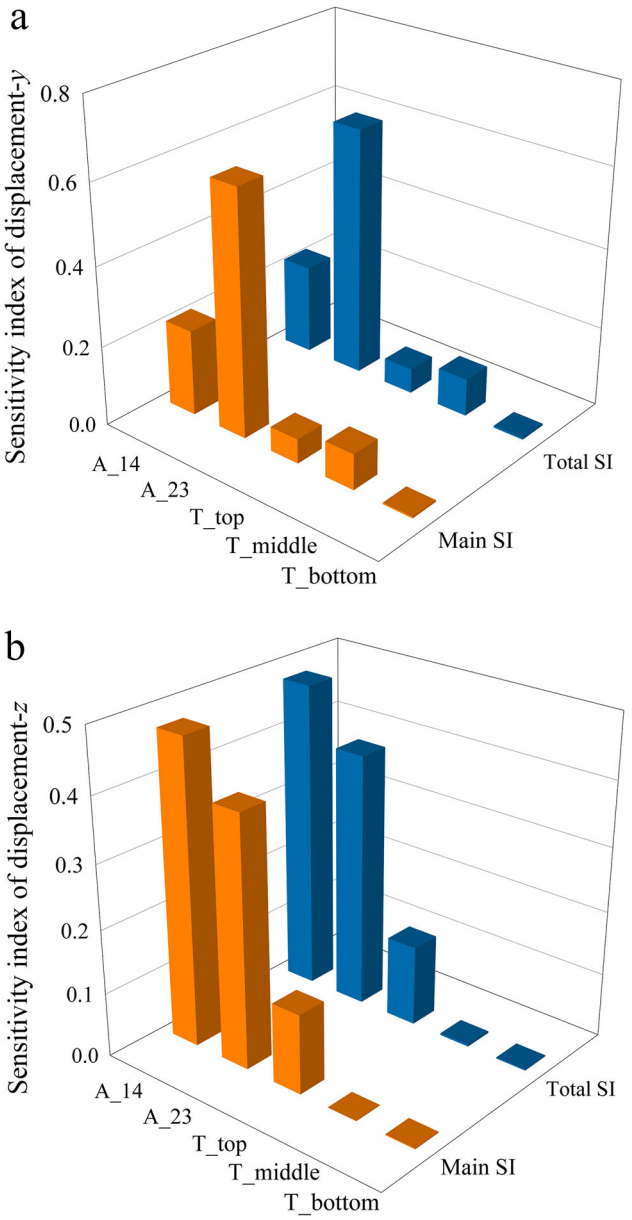

Fig. 19 Sensitivity indices of displacements (Actuation: 2-dimensional, Temperature: 3-dimensional): **a** displacement-y; **b** displacement-z

Additionally, the total-order index measures the contribution to the output variance caused by a model input, including both its first-order effects (the input varying alone) and all higher-order interactions. The total-order index is given by

$$S_{Ti} = \frac{\mathrm{E}_{x_{\sim i}}(\mathrm{V}_{x_i}(y|\mathbf{x}_{\sim \mathbf{i}})}{\mathrm{V}(y)}. \tag{16}$$

$\mathbf{X}_{\sim \mathbf{i}}$ is a $N \times (d_{in} - 1)$ matrix of all factors but x_i, where N is the sample size.

The calculated Sensitivity indices with 100,000 DoEs are plotted in Fig. 19. As observed, the sensitivity of T_bottom

(temperature subjected to cables on the bottom surface) to displacements at node 23 is negligible. Of all the input variables, the sensitivity is much larger for actuation than for temperature. One input variable could have different sensitivities to different QoIs. For example, A_14 (actuation to clustered cables on the top surface) exhibits a high sensitivity to displacement-z, but a relatively low sensitivity to displacement-y. A_23 shows the highest sensitivity to displacement-y. Qualitative analysis is helpful for optimization design to reduce the dimension of variables.

Appendix B

HDBSCAN algorithm

After putting all 100,000 DoEs through FEA, the resulting QoIs are drawn as a scatter plot in Fig. 20a. Due to unconvergence and the collapse of the structure, not all of the 100,000 DoEs can be used for training. HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) is an unsupervised clustering algorithm used for outlier detection in data selection [48]. It extends DBSCAN by converting it into a hierarchical clustering algorithm and then using a technique to extract a flat clustering based on the stability of clusters [49]. It is quite different from the classical K-mean clustering [50], which relies on strong assumptions about the data: The total number of clusters K is prescribed by the user without any insight from the data, and the data is assumed scattering around these K cluster centroids following a Gaussian distribution. Following the literature [51, 52], five steps of the HDBSCAN algorithm are explained as follows,

- Step 1: Estimate the density of distributed data.

Estimate the density of distributed data with mutual reachability distance $d_{\text{mreach - }k}(a, b) = \max\{\text{core}_k(a), \text{core}_k(b), d(a, b)\}$, where $\text{core}_k(a)$ means the distance between a and its k-th nearest neighbor, $d(a, b)$ is the distance between a and b. Here the mutual reachability distance is an estimation of the Probability Density Function (PDF) of the data, $\lambda = \frac{1}{d_{mreach-k}}$. A smaller value in d implies a larger PDF λ.

- Step 2: Build a minimum spanning tree for the data.

Consider the data as a weighted graph with the data points as vertices and an edge between any two points with a weight equal to the mutual reachability distance of those points. In such a minimum spanning tree, no lower weight edge could connect the components.

- Step 3: Build the cluster hierarchy.

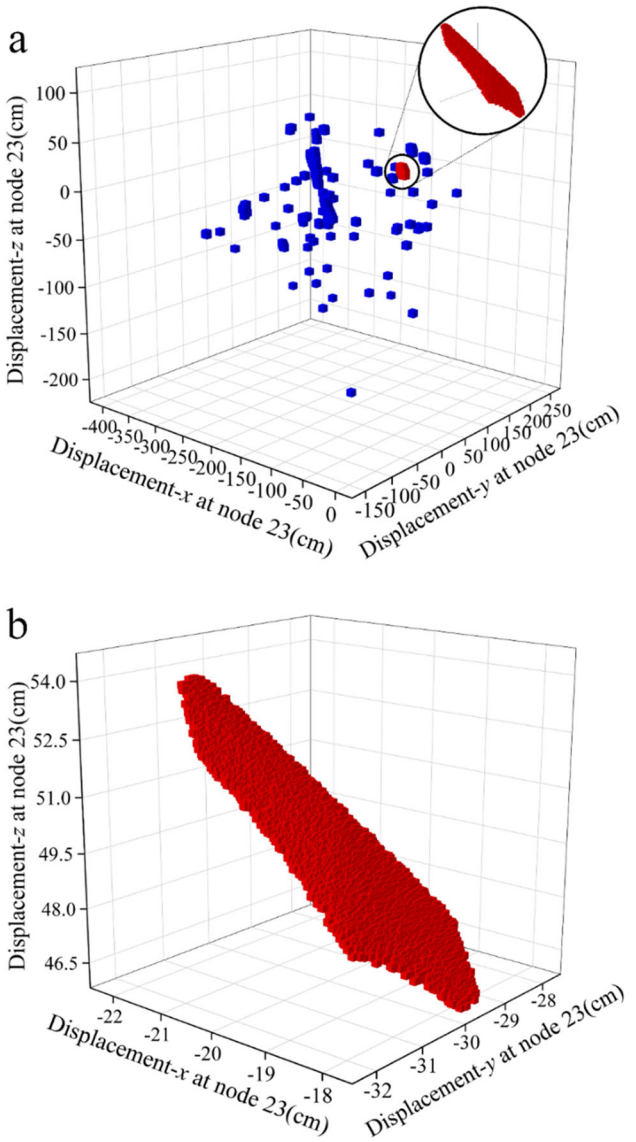

Fig. 20 Outlier detection by a clustering algorithm, HDBSCAN: **a** Total data; **b** The selected data

Given the minimal spanning tree, sort the edges of the tree by distance in increasing order and then iterate through, creating a new merged cluster for each edge.

- Step 4: Condense the cluster tree.

The first step in cluster extraction is condensing down the large and complicated cluster hierarchy into a smaller tree with a little more data attached to each node. The minimum cluster size is taken as a user prescribe parameter to the algorithms.

- Step 5: Extract the clusters.

Now, for each cluster compute the stability as $\sum_{p \in \text{cluster}} (\lambda_p - \lambda_{\text{birth}})$, where λ_{birth} is the value when the cluster split off and became its own cluster, λ_p is the value at which that point 'fell out of the cluster'. Declare all leaf nodes to be selected clusters. Work up through the tree in the reverse topological sort order. If the sum of the stabilities of the child clusters is greater than the stability of the cluster, then we set the cluster stability to be the sum of the child stabilities. Otherwise, we declare the cluster to be a selected cluster and unselect all its descendants. Once we reach the root node, we call the current set of selected clusters our flat clustering, and the algorithm is finished.

The HDBSCAN algorithm makes a few implicit assumptions about the data and asks the user to give the minimum cluster size as the only control parameter. In this study, we used the Python implementation provided by McInnes [52]. Here, failure cases are regarded as noise, and the minimal cluster size is set to be 40. The results are separated into four groups, and the largest group which contains all the viable solutions to the DoEs has a total of 98,365 samples which take up over 98.4% of the data, as shown in Fig. 20b. The converged results from the first two groups are selected to create the training database for the data-driven studies. It is worthwhile to mention that even though no convergence constraints are given, all the selected DoEs meet the convergence tolerance.

References

1. Rhode-Barbarigos L, Bel Hadj Ali N, Motro R, Smith I (2012) Design aspects of a deployable tensegrity-hollow-rope footbridge. Int J Space Struct. 27:81–96
2. G. Tibert, Deployable tensegrity structures for space applications, (Ph.D. thesis), Cambridge, Royal Institute of Technology, (2002).
3. Wang N, Naruse K, Stamenovic D, Fredberg JJ, Mijailovich SM, Toric-Norrelykke IM, Polte T, Mannix R, Ingber DE (2001) Mechanical behavior in living cells consistent with the tensegrity model. Proc Natl Acad Sci U S A 98:7765–7770
4. Xu G-K, Li B, Feng X-Q, Gao H (2016) A tensegrity model of cell reorientation on cyclically stretched substrates. Biophys J 111:1478–1486
5. Fraternali F, Carpentieri G, Amendola A (2015) On the mechanical modeling of the extreme softening/stiffening response of axially loaded tensegrity prisms. J Mech Phys Solids 74:136–157
6. Luo A, Liu H (2017) Analysis for feasibility of the method for bars driving the ball tensegrity robot. J Mech Robot 9:051010
7. Kim K, Agogino AK, Agogino AM (2020) Rolling locomotion of cable-driven soft spherical tensegrity robots. Soft Robot 7:346–361
8. Surovik D, Wang K, Vespignani M, Bruce J, Bekris KE (2021) Adaptive tensegrity locomotion: Controlling a compliant icosahedron with symmetry-reduced reinforcement learning. Int J Robot Res 40:375–396
9. J. Kimber, Z. Ji, A. Petridou, T. Sipple, K. Barhydt, J. Boggs, L. Dosiek, J. Rieffel, Low-cost wireless modular soft tensegrity robots, In 2019 2nd IEEE International Conference on Soft Robotics (RoboSoft), 2019, pp. 88–93.
10. Rieffel J, Mouret J-B (2018) Adaptive and resilient soft tensegrity robots. Soft Robot 5:318–329
11. Moored KW, Bart-Smith H (2007) The analysis of tensegrity structures for the design of a morphing wing. J Appl Mech-Trans Asme 74:668–676
12. Moored KW, Bart-Smith H (2009) Investigation of clustered actuation in tensegrity structures. Int J Solids Struct 46:3272–3281
13. Ali NBH, Smith IFC (2010) Dynamic behavior and vibration control of a tensegrity structure. Int J Solids Struct 47:1285–1296
14. Ali NBH, Rhode-Barbarigos L, Smith IFC (2011) Analysis of clustered tensegrity structures using a modified dynamic relaxation algorithm. Int J Solids Struct 48:637–647
15. Tran HC, Lee J (2011) Geometric and material nonlinear analysis of tensegrity structures. Acta Mech Sin 27:938–949
16. Zhang L, Gao Q, Liu Y, Zhang HW (2016) An efficient finite element formulation for nonlinear analysis of clustered tensegrity. Eng Comput 33(1):252–273
17. Kan Z, Peng H, Chen B, Zhong W (2018) Nonlinear dynamic and deployment analysis of clustered tensegrity structures using a positional formulation FEM. Compos Struct 187:241–258
18. Kirchdoerfer AT, Ortiz M (2016) Data-driven computational mechanics. Comput Methods Appl Mech Eng 304:81–101
19. Oishi A, Yagawa G (2017) Computational mechanics enhanced by deep learning. Comput Methods Appl Mech Eng 327:327–351
20. Prudencio EE, Bauman PT, Faghihi D, Ravi-Chandar K, Oden JT (2015) A computational framework for dynamic data-driven material damage control, based on Bayesian inference and model selection. Int J Numer Methods Eng 102:379–403
21. Lima EABF, Oden JT, Wohlmuth B, Shahmoradi A, Hormuth DA II, Yankeelov TE, Scarabosio L, Horger T (2017) Selection and validation of predictive models of radiation effects on tumor growth based on noninvasive imaging data. Comput Methods Appl Mech Eng 327:277–305
22. Jha PK, Cao L, Oden JT (2020) Bayesian-based predictions of COVID-19 evolution in Texas using multispecies mixture-theoretic continuum models. Comput Mech 66:1055–1068
23. Bessa MA, Bostanabad R, Liu Z, Hu A, Apley DW, Brinson C, Chen W, Liu WK (2017) A framework for data-driven analysis of materials under uncertainty: countering the curse of dimensionality. Comput Methods Appl Mech Eng 320:633–667
24. Bessa MA, Pellegrino S (2018) Design of ultra-thin shell structures in the stochastic post-buckling range using Bayesian machine learning and optimization. Int J Solids Struct 139:174–188
25. Bessa MA, Glowacki P, Houlder M (2019) Bayesian machine learning in metamaterial design: fragile becomes supercompressible. Adv Mater 31(48):1904845
26. Li X, Ning S, Liu Z, Yan Z, Luo C, Zhuang Z (2020) Designing phononic crystal with anticipated band gap through a deep learning-based data-driven method. Comput Methods Appl Mech Eng 361:112737
27. Qiu Z, Ma L, Wang X (2009) Non-probabilistic interval analysis method for dynamic response analysis of nonlinear systems with uncertainty. J Sound Vib 319:531–540
28. Wang C, Qiu Z (2015) Hybrid uncertain analysis for steady-state heat conduction with random and interval parameters. Int J Heat Mass Transf 80:319–328
29. Luo Y, Xing J, Kang Z, Zhan J, Li M (2020) Uncertainty of membrane wrinkling behaviors considering initial thickness imperfections. Int J Solids Struct 191:264–277
30. Simpson TW, Lin D (2001) Sampling strategies for computer experiments: design and analysis. Int J Reliab Appl 2:209–240
31. Mckay MD, Beckmanw RJ, Conover J (1979) A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics. 21:239–245

32. Sobol IM (1969) The distribution of points in a cube and the approximate evaluation of integrals. USSR Comput Math Math Phys 7:784–802
33. Rasmussen CE, Williams CKI (2005) Gaussian processes for machine learning. The MIT Press
34. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521:436–444
35. Goodfellow I, Bengio Y, Courville A (2016) Deep Learning, Illustrated. The MIT Press
36. D. Kingma, J. Ba, Adam: A method for stochastic optimization, (2014). http://arxiv.org/abs/1412.6980.
37. Byrd RH, Lu P, Nocedal J, Zhu C (1995) A limited memory algorithm for bound constrained optimization. Siam J Sci Comput 16:1190–1208
38. Fletcher R (1981) Practical methods of optimization: constrained optimization. Wiley
39. Hock W, Schittkowski K (1983) A comparative performance evaluation of 27 nonlinear programming codes. Computing 30:335–358
40. Zitzler E, Thiele L (1999) Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. IEEE Trans Evol Comput 3:257–271
41. J. Snoek, H. Larochelle, R.P. Adams, Practical Bayesian optimization of machine learning algorithms, in: Procceeding 25th International Conference Neural Information Processing Systems - Vol. 2, Curran Associates Inc., Red Hook, NY, USA, 2012: pp. 2951–2959.
42. J.T. Wilson, F. Hutter, M.P. Deisenroth, Maximizing acquisition functions for Bayesian optimization, in: S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. CesaBianchi, R. Garnett (Eds.), Adv. Neural Inf. Process. Syst. 31 Nips 2018, Neural Information Processing Systems (nips), La Jolla, 2018.
43. B. Peng, F. Yang, C. Yan, X. Zeng, D. Zhou, Efficient multiple starting point optimization for automated analog circuit optimization via recycling simulation data, in: Proceeding 2016 Des. Autom. Test European Conference Exhibition Date, IEEE, New York, 2016, pp. 1417–1422.
44. Saltelli A (2002) Making best use of model evaluations to compute sensitivity indices. Comput Phys Commun 145:280–297
45. Saltelli A, Annoni P, Azzini I, Campolongo F, Ratto M, Tarantola S (2010) Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index. Comput Phys Commun 181:259–270
46. Sobol IM (2001) Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. Math Comput Simul 55:271–280
47. Herman J, Usher W (2017) SALib: an open-source python library for sensitivity analysis. J Open Source Softw 2:97
48. R. Campello, D. Moulavi, J. Sander, Density-based clustering based on hierarchical density estimates, In: 2013, pp. 160–172.
49. M. Ester, H.-P. Kriegel, J. Sander, X. Xu, A Density-based algorithm for discovering clusters in large spatial databases with noise, In: 1996, pp. 226–231.
50. Hartigan JA, Wong MA (1979) Algorithm AS 136: a k-means clustering algorithm. J R Stat Soc Ser C Appl Stat 28:100–108
51. R.J.G.B. Campello, D. Moulavi, J. Sander, Density-based clustering based on hierarchical density estimates, in: Adv. Knowl. Discov. Data Min., Springer, Berlin, Heidelberg, 2013, pp. 160–172.
52. McInnes L, Healy J, Astels S (2017) hdbscan: Hierarchical density-based clustering. J Open Source Softw 2:205