



## Machine learning assisted evaluations in structural design and construction

Hao Zheng<sup>a</sup>, Vahid Moosavi<sup>b</sup>, Masoud Akbarzadeh<sup>a,\*</sup>

<sup>a</sup> Polyhedral Structures Laboratory, Weitzman School of Design, University of Pennsylvania, Philadelphia, USA

<sup>b</sup> Chair for Computer Aided Architectural Design, Institute of Technology in Architecture, ETH Zurich, Zurich, Switzerland



### ARTICLE INFO

**Keywords:**

Iterative machine learning  
Neural networks  
Surrogate model  
3D graphic statics  
Subdividing the force diagram

### ABSTRACT

This paper proposes a new design approach based on an iterative machine learning algorithm to speed up the topological design exploration of compression-only shell structures with planar faces, considering both structural performance and construction constraints. In this paper, we show that building neural networks allows one to train a surrogate model to accelerate the structural performance assessment of various possible structural forms without going through a significantly slower process of geometric form-finding. The geometric form-finding methods of 3D graphic statics are used as the primary structural design tool to generate a single-layer, compression-only shell with planar faces. Subdividing the force diagram and its polyhedral cells using various rules results in a variety of topologically different compression-only structures with different load-bearing capacities for the same boundary conditions. The solution space for all possible compression-only forms for a given boundary condition is vast, which makes iterating through all forms to find the ideal solutions practically impossible. After training with an iterative active sampling method, the surrogate model can evaluate the input data, including the subdivision rules, and predict the value of the structural performance and the construction constraints of the planar faces within milliseconds. As a result, one can then evaluate the nonlinear relations among all the subdivision rules and the chosen structural performance measures, and then, visualize the entire solution space. Consequently, multiple solutions with customized thresholds of the evaluation criteria are found that show the strength of this method of form-finding in generating design solutions. Besides, considering the total training time of the neural network model, the proposed framework is still faster than a traditional optimization method, such as the genetic algorithm that can find only the optimum values. This process will result in interactive sampling methods in which the machine learning models assist the designer in choosing and controlling different design strategies by providing real-time feedback on the effects of the selected parameters on the design outputs.

### 1. Introduction

Designing structures by considering the internal force flow in their members can significantly reduce the use of construction materials and the related costs. Geometry-based structural design methods known as Graphic statics (GS), represent a group of techniques that has been used and developed for the past 150 years as an intuitive approach addressing the economics of construction [12]. In 2D/3D Graphic statics the equilibrium of the internal and external forces is represented by two reciprocal diagrams that are topologically and geometrically related [14,15,22,28,36]. The *form* diagram shows the geometry and the boundary conditions of the structure, and the *force* diagram represents the equilibrium of the force in the geometry of the form. The dependency of these two diagrams allows us to generate and control the properties of one diagram from the other. Moreover, a designer can

explicitly control and optimize the magnitude of the internal forces in the structure by designing the geometry of the force diagram.

At the end of the nineteenth century, there was a shift from using graphic-statics based structural design and analysis methods to numerical methods to avoid the lengthy process of constructing the geometric diagrams of forces. The recent advances in computing power, however, have allowed the reemergence of the geometry-based structural design methods particularly in three dimensions after almost a century. The complex geometric diagrams of forces can now be constructed in milliseconds using the current digital computation which allows structural designers and architects to explore an unexplored realm of efficient spatial structural forms in 3D [4,10,11,17,30,31,34,35,39].

The methods of 3D graphic statics allow designers to create 3D funicular solutions for a given boundary conditions by designing and

\* Corresponding author.

E-mail addresses: [zhao@design.upenn.edu](mailto:zhao@design.upenn.edu) (H. Zheng), [svm@arch.ethz.ch](mailto:svm@arch.ethz.ch) (V. Moosavi), [masouda@upenn.edu](mailto:masouda@upenn.edu) (M. Akbarzadeh).

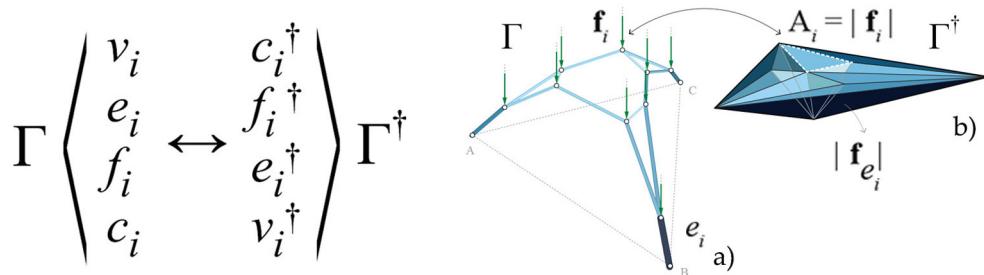


Fig. 1. 2D versus 3D funicular solutions and their corresponding force diagrams [5].

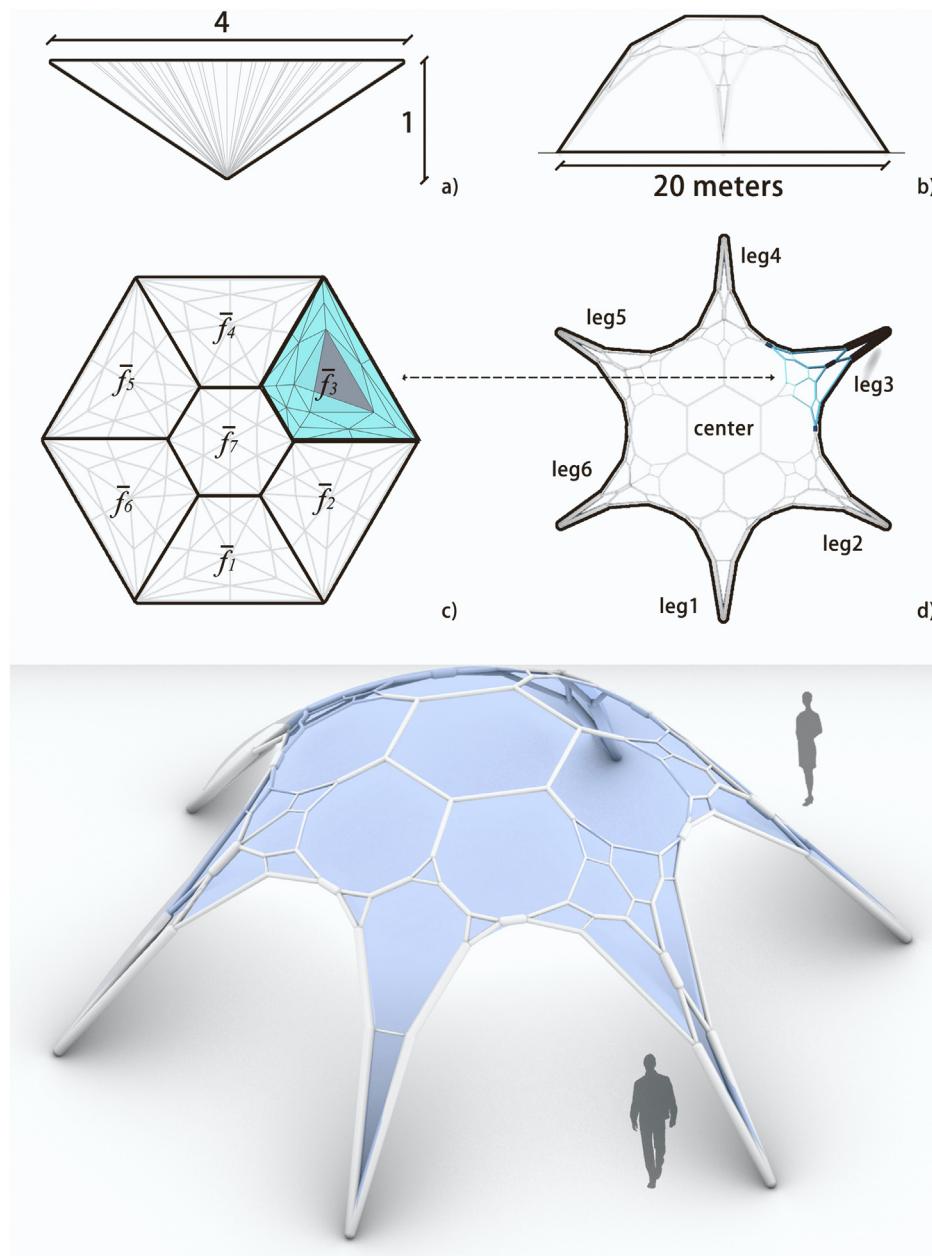


Fig. 2. Design background – a single-layer funicular shell structure with six legs across 20 m.

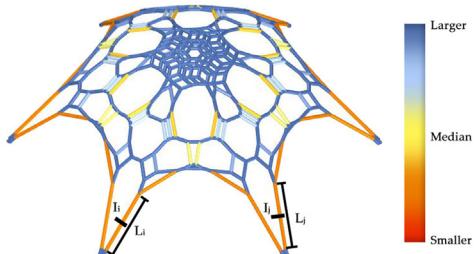
manipulating the force diagrams and the topology of the structural form [2,3,9].

In the form-finding method based on reciprocal polyhedral diagrams of 3D graphic statics (3DGS), the force diagram consists of closed polyhedral cells with planar faces. Each vertex  $v$ , edge  $e$ , face  $f$ , and cell

$c$  of the form diagram  $\Gamma$  corresponds to a cell  $c^\dagger$ , face  $f^\dagger$ , edge  $e^\dagger$  and vertex  $v^\dagger$  of the force diagram  $\Gamma^\dagger$ . The areas of each face  $f^\dagger$  represent the magnitude of a force in the edge  $e$  of the form diagram (Fig. 1). Based on this definition, the external faces of the force polyhedron represent the external loads and reaction forces at the supports of the structural

$$\mathbf{f}_b = \text{Min}\left(\frac{\pi^2 EI_i}{(KL_i)^2}\right) \quad i = (1..n)$$

(E = 200GPa; K = 1)

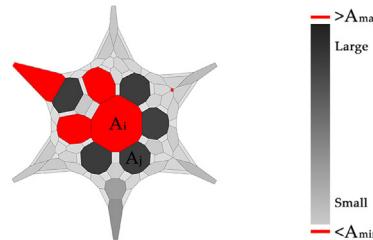


**Fig. 3.** Buckling capacity based on Euler's critical load formula: the first criteria to evaluate the forms. E: modulus of elasticity; I: smallest area moment of inertia; K: column effective length factor; L: length of the member.

$$\mathbf{N} = \mathbf{N}_{(A_i > A_{\max})} + \mathbf{N}_{(A_i < A_{\min})} \quad i = (1..n)$$

$$A_{\max} = 6.25 \text{ m}^2 (2.5 \text{ m} \times 2.5 \text{ m})$$

$$A_{\min} = 0.09 \text{ m}^2 (0.3 \text{ m} \times 0.3 \text{ m})$$



**Fig. 4.** The number of unacceptable faces: the second criterion to evaluate the forms.

form. For instance, the applied load  $f_i$  in the form diagram corresponds to a face  $f_i^\dagger$ . The force  $f_i$  is perpendicular to the face  $f_i^\dagger$  and the area of the face represents the magnitude of the load in the form diagram.

Thus, as long as the force diagram consists of a set of closed polyhedrons, the corresponding form is in equilibrium. A designer can subdivide the force polyhedrons to change the topology of the structural form and thus can explore a variety of different funicular structural forms for the very same boundary condition [19]. The generated structures are always under equilibrium for the given boundary condition. Moreover, the geometry of the compression-only structural forms has planar faces that can be built using flat sheet materials – due to the inherent planarity constraints of the reciprocal polyhedral diagrams. This is a great advantage of using 3DGS over any other structural form finding technique. While having infinite funicular solutions for the same boundary condition might be ideal from design point of view, having particular construction and performative constraints seems quite necessary to choose multiple options from this infinite solution space. We will provide the following design example to elaborate on the necessity of having such control over the results.

### 1.1. Design background

Consider a single-layer, compression-only shell structure covering a circular site with a diameter of 20 m as a design problem (Fig. 2). A 2D hexagonal pattern including polygons  $\bar{f}_{1-7}$  can be used to construct the 3D polyhedral force diagram. The force diagram can be constructed as a set of polyhedrons generated by extruding the polygons  $\bar{f}_{1-7}$  in 2D to a center point downside (Fig. 2c, a). Fig. 2a and c illustrate the elevation and plan of the force diagram, in which the applied loads are balanced by the reaction forces at the supports of the structure from six sides. Multiple geometries for the form diagram can be generated from the force diagram with the constraint from the boundary conditions, where the shell structure contains six legs with one center part (Fig. 2d). The resulting shell covers the area with a diameter of 20 m (Fig. 2b).

Fig. 2e illustrates an example of a shell on which the planar panels are supported by a structure with circular Cross sections. The form can represent a glass shell structure supported by metal frames. One can design various force diagrams by subdividing from the hexagonal faces of the form, and derive a variety of compression-only forms. The PolyFrame [27] plugin for Rhinoceros software [23] can be used as a computational tool to generate form diagrams from a given force

diagram.

### 1.2. Problem statement

Among a variety of forms, the structural performance varies significantly when constructing the structure with steel pipes. Based on the conclusion by [19], the load-bearing capacity of funicular forms could be improved if the force diagram is subdivided. That means the minimum buckling capacity of all members in an ideal form should be larger than that of the members in other forms (Fig. 3). Thus, the weakest element in the form should have the ability to bear more loads.

Other than considering it from the point of the structural performance, constructability is also essential in actual cases. We introduce an additional evaluation criterion which does not accept faces with very large or very small areas – the maximum fabrication area for this problem is 6.25 square meters and the minimum fabrication area is 0.09 square meters (Fig. 4). It would be very expensive for the project to construct such faces, and we need to exclude the structural solutions with such unacceptable faces. Therefore, a better solutions from construction point of view are those with a smaller number of unacceptable faces (ideally 0).

With these two evaluation criteria, any form that is generated by this method can be evaluated. The conventional computational process can be used in loops and evaluation criteria to generate a variety of forms by applying different subdivision rules for the same boundary condition (Fig. 5).

It usually takes around 40 s to generate a form using PolyFrame which uses an iterative algorithm to generating a form diagram from a given force diagram [4]. We can assume that there are only five subdivision rules for the six side parts and seven subdivision rules for the center part of the force diagram where each subdivision rule can take five different numbers of the segments. In that case, the total number of possible solutions will be 546,875 ( $5^6 \times 7 \times 5$ ), resulting in a processing time of 21,875,000 s, that is, 253 days of uninterrupted computing. Running an exhaustive search among this enormous solution space and finding all the ideal forms with maximum buckling capacities and minimum numbers of unacceptable faces seems impossible within a reasonable time limit. Even using genetic algorithm approach to find the solution might take a long time and will result in single optimized solutions.

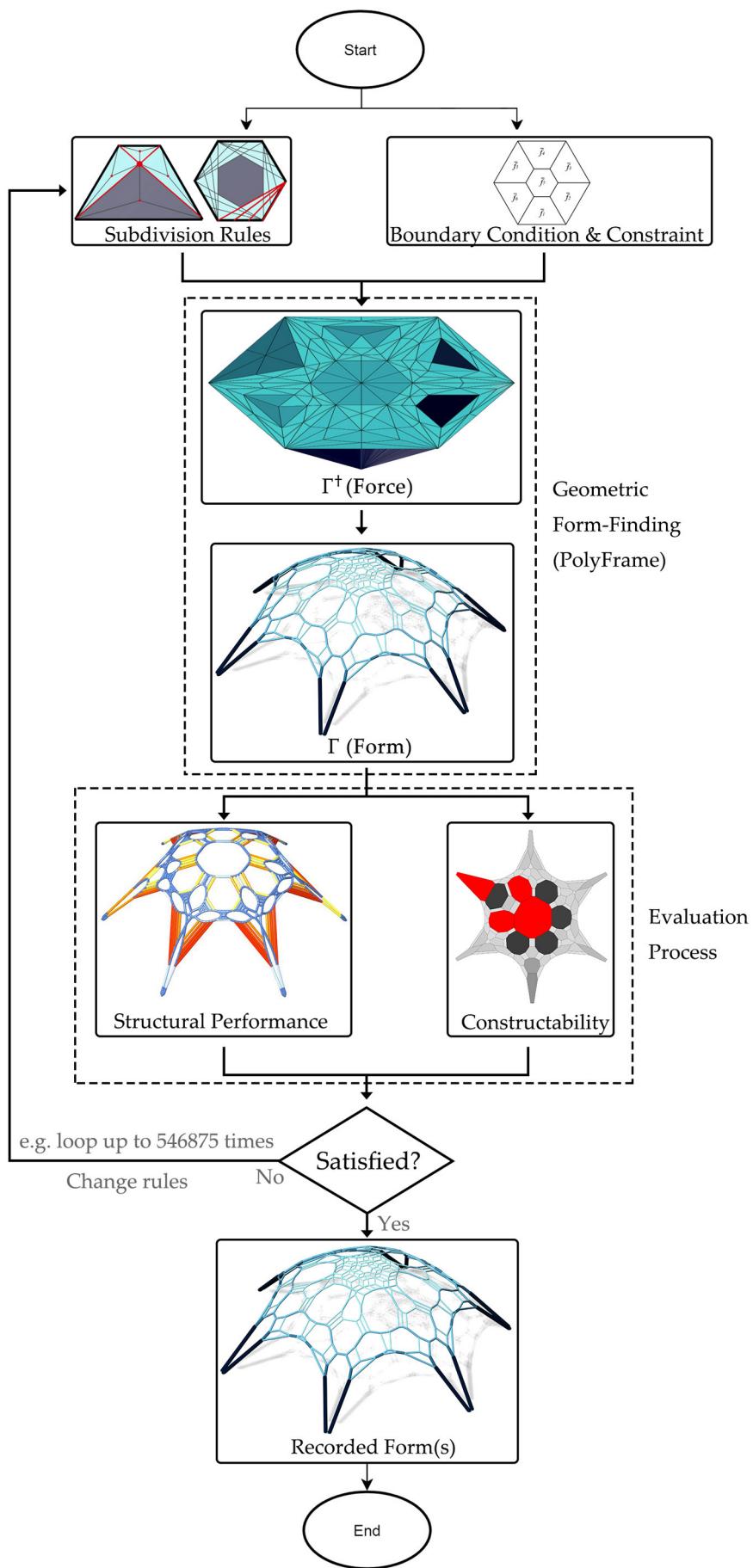
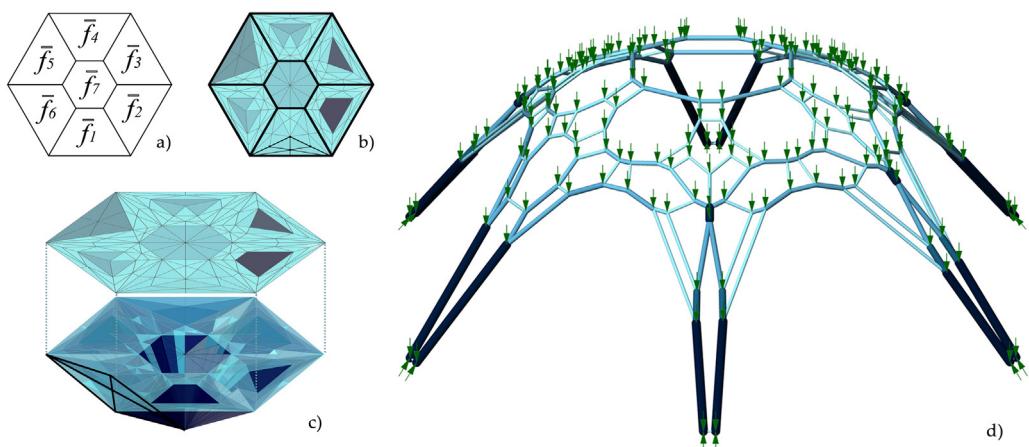
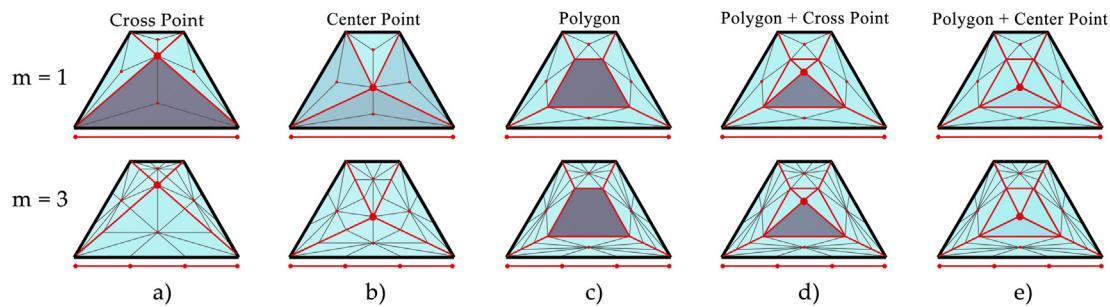


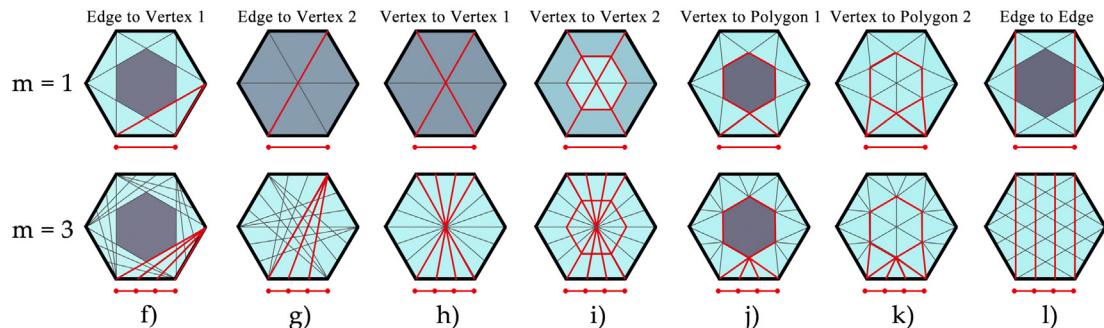
Fig. 5. A conventional computational form finding flowchart.



**Fig. 6.** One example of subdividing the force diagram and its related compression-only shell.



**Fig. 7.** Different subdivision rules for the side units and the center unit.



**Fig. 8.** Different subdivision rules for the center unit.

### 1.3. Machine learning

Recent advances in machine learning techniques offer promising data-driven approaches to ascertain the nonlinear and high-dimensional relations between forces and the structural performance of the generated forms [18,21].

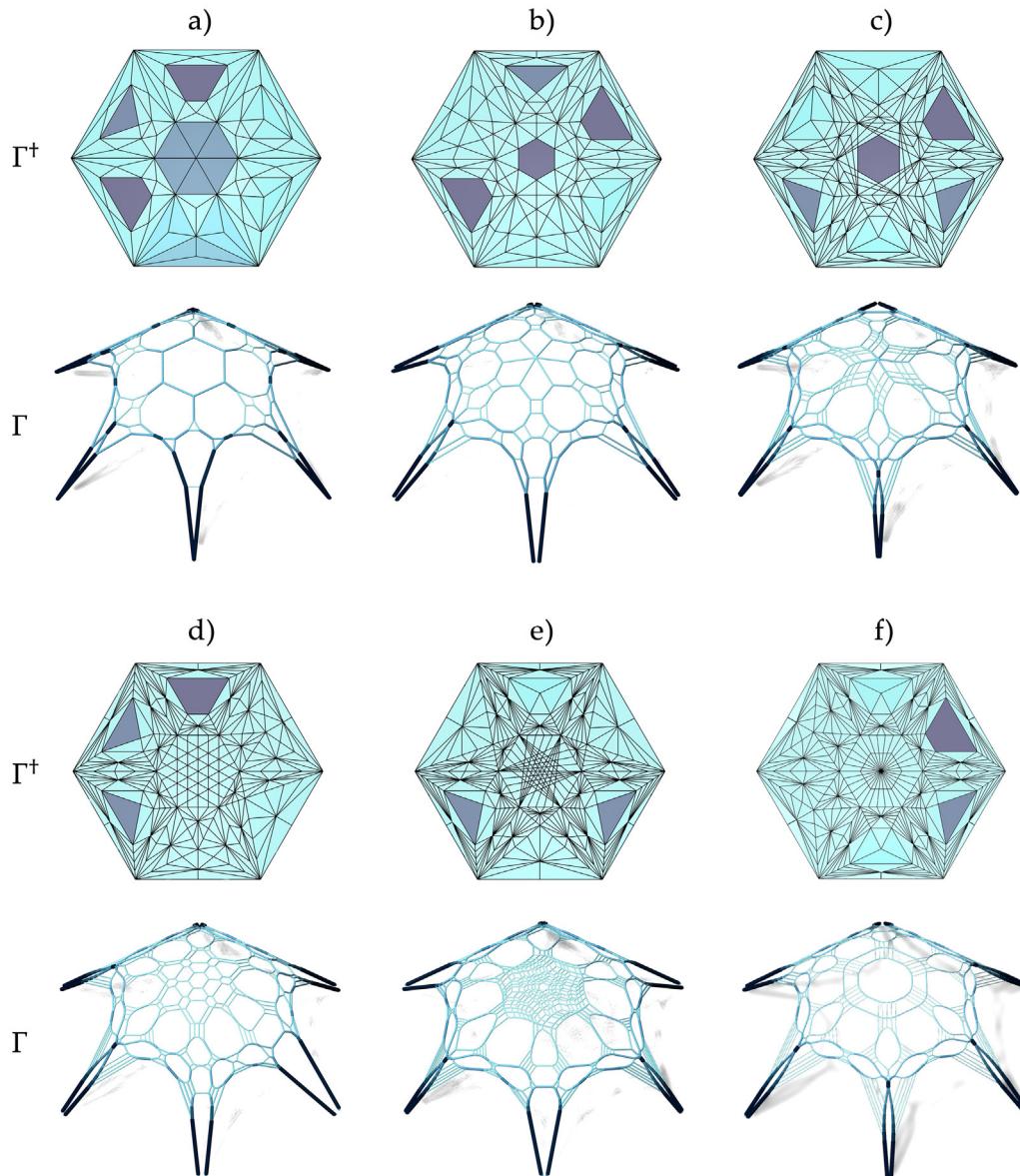
Related research on the application of machine learning in structural design includes applying a general regression framework to the prediction of 3D form data of a bent planar surface [16]; quantifying and optimizing the aesthetics factors in structural design using artificial neural networks [39]; generating structural forms based on the machine learning of the user-provided evaluation data [33]; machine learning the results in the finite element analysis (FEA) to optimize the computation process [6]; and generating small-scale truss structures based on the machine learning of the structural performance of the building units [38]. Previous studies have focused on the generation of target structural solutions without the purpose of exploring the entire solution space.

In addition, machine learning has performed magnificently in the

automation of the construction process. Related research includes capturing real-time images and training a neural network to look for the defects in manufacturing [8]; training neural networks to simulate the process of human-made woodworks [13]; applying image-based neural networks to identify the location of the bamboo nodes and guide the construction of the bamboo structure [37]; training a path-planning machine to generate the moving trajectory of the robotic arm while avoiding obstacles [29]; quantitatively evaluating the safety in the construction process using machine learning [32]; detecting construction workers with motion, shape, and color features using machine learning [26]; and enhancing the decision making in contractor pre-qualification [20]. These topics are related to the fabrication process during construction, but they do not involve examining the constructability from the perspective of the design before construction.

### 1.4. Objectives

This research aims to propose a machine learning assisted method that accelerates the time-consuming form finding process of 3D graphic



**Fig. 9.** Different forms generated from different subdivided force diagrams.

**Table 1**  
5-Fold cross-validation of neural networks with different numbers of layers.

Network structure	Median accuracy for the buckling capacity (%)	Median accuracy for the number of unacceptable faces (%)
2-Layer ANN	93.4	93.6
3-Layer ANN	95.4	95.8
4-Layer ANN	97.0	97.4
<b>5-Layer ANN</b>	<b>98.2</b>	<b>98.8</b>
6-Layer ANN	97.2	97.9
7-Layer ANN	96.4	96.6

statics, thus making it possible to explore a vast solution space without going through the entire geometry generation process. This approach should be able to find all the ideal forms with higher buckling capacities and smaller numbers of unacceptable faces. We propose to address this problem by developing a surrogate model for the form finding process. A surrogate model is a reduced model of a slower computational process (e.g., a physics-based simulation) and is usually developed to learn the nonlinear relationships between the important input and output values of the original model without the need to entirely run the slower

model. Surrogate or meta-models are very well-known concepts in fields such as mechanical system design or aerodynamic systems [7]. We expect that the whole process for one prediction should be completed within milliseconds. Therefore, exploring all possible solutions and filtering the ideal results should take place under a strict time constraint using the surrogate model.

With this surrogate model, the design performance of the forms using different subdivision rules can be analyzed, and the solution space can be visualized. The conventional time-consuming form-finding process can be replaced by the faster machine learning assisted process so that ideal forms can be found among the whole solution space within a reasonable time limit.

## 2. Methodology

The first step in developing a surrogate model for our multi-objective design criteria is to define the variables related to the form finding process.

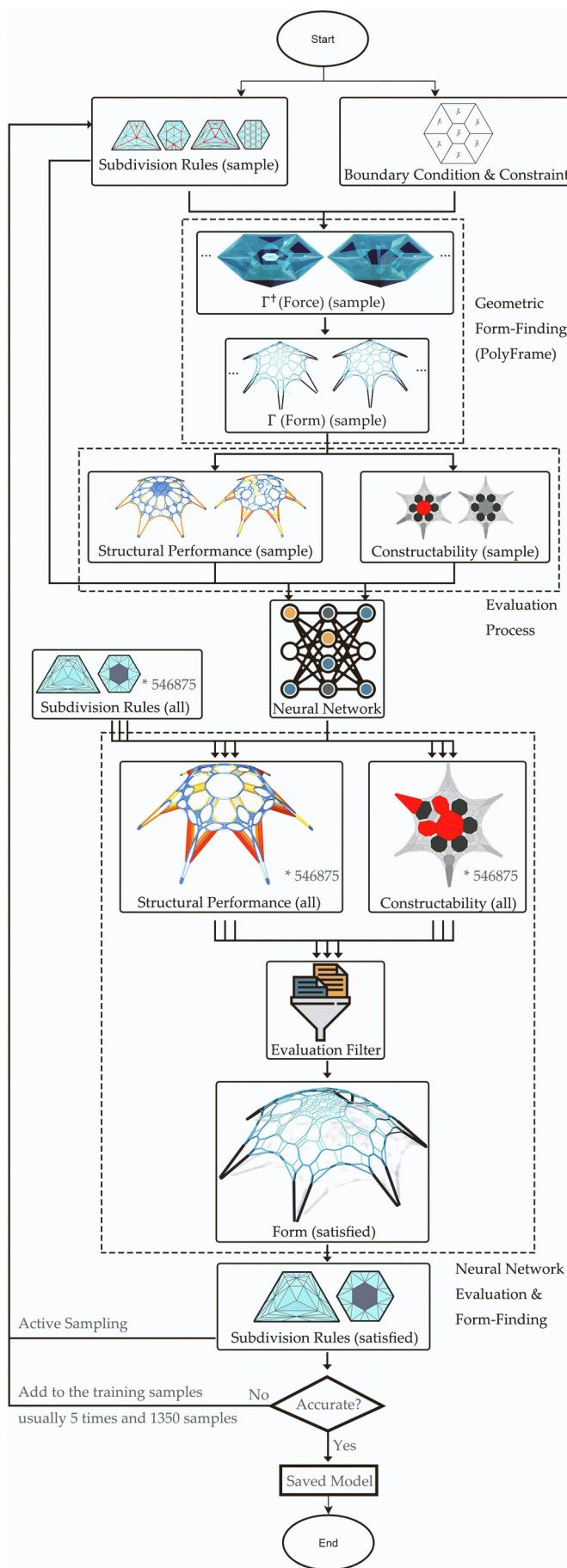


Fig. 10. Machine learning assisted iterative training method.

**Table 2**

Median accuracy of the neural networks in different training loops (buckling capacity (%)/the number of unacceptable faces (%)).

	Random data	Small results	Equal results	Large results
Loop 1	87.4/95.3	85.3/97.9	87.5/96.4	27.1/10.9
Loop 2	94.2/97.6	95.8/98.7	93.9/97.4	42.2/91.5
Loop 3	96.1/97.8	96.1/99.3	95.1/97.9	93.9/97.9
Loop 4	97.0/98.2	96.3/98.6	96.7/98.4	99.7/96.7
Loop 5	97.0/98.6	95.9/99.4	95.1/98.2	98.5/99.3

**Table 3**

The number of instances and required time of the neural networks in different training loops and the genetic algorithm.

	Instances amount	Required time	Best found?	Surrogate?
Loop 1	175	2.03 h	No	Yes
Loop 2	475	5.33 h	No	Yes
Loop 3	775	8.63 h	Yes	Yes
Loop 4	1075	11.97 h	Yes	Yes
Loop 5	1375	15.31 h	Yes	Yes
GA	/	9.56 h	Yes	No

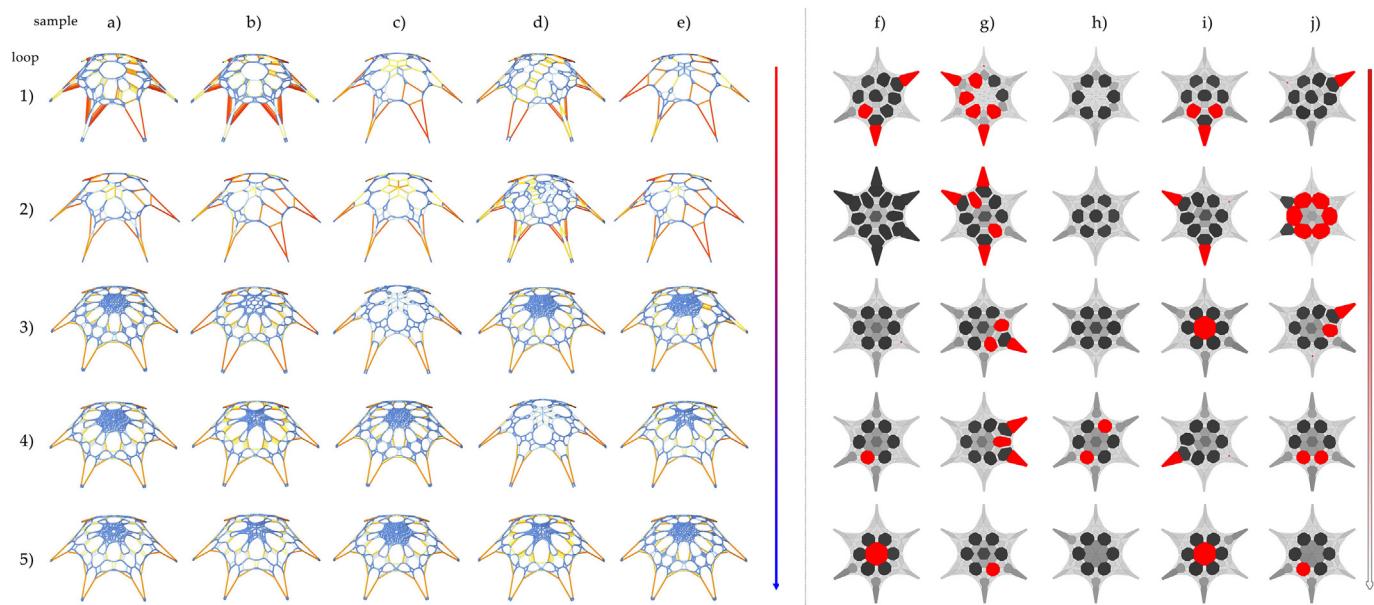
## 2.1. Variables defining the force and form diagrams

In the proposed form-finding process using 3D graphic statics, there are three important inputs: *the boundary conditions* of the force diagram, *the subdivision rules* of the force diagram, and the *geometric constraints* of the form diagram [19].

The first input describes the fixed boundary condition of the force diagram, which also defines the external forces. Generally speaking, the boundary condition can be any closed polyhedron representing various states of external forces. However, in this research, the boundary condition the top face is large hexagon that is divided into seven smaller faces (Fig. 6a). By extruding the vertices of the 2D pattern to a center point similar to Fig. 2, the 3D boundary polyhedron can be constructed (Fig. 6c). This force polyhedron corresponds to a single shell structure for the purpose of this research. The external faces of the force polyhedron correspond to the applied loads and the reaction forces in a shell. Thus, the geometry of the initial force diagram with no subdivision includes seven external loads from the top and six reaction forces on the periphery of the shell. Each internal polyhedral cell represents the equilibrium of a node, and each face is perpendicular to a member in the form. Once subdivided, each segment is divided into multiple polygons that represent applied loads for the geometry of the shell.

The second input, the subdivision rules, describes the strategies by which a given original force diagram is subdivided into several cells, resulting in a force diagram with more cells. The subdivision in this research is based on edges. Each edge in the unit face of the original force diagram is subdivided into several segments; the start and end points of each segment, as well as the center points of each face, will then together become the new subdivided faces in the unit of the force diagram as a 2D pattern (Fig. 6b). Finally, by extruding the edges of the 2D pattern to a center point below the surface, 3D faces are generated (Fig. 6c); the faces together become the force diagram.

For each unit, five types of subdivision rules for the side unit face ( $\bar{f}_1 - \bar{f}_6$ ) (Fig. 7) and seven types of subdivision rules for the center unit face ( $\bar{f}_7$ ) are developed (Fig. 8). Each subdivision rule represents a unique technique to subdivide the unit face, and each edge in the unit face can be subdivided into several segments with different settings of the subdivision count  $m$ . For example, rule a-3 for the side unit face first subdivides the face with the connecting lines between a center point and the four vertices; the center points for the four subdivided faces will then be connected by the four subdivided points (including the start and the end points) in each segment of the faces, since the subdivision count  $m$  is 3 in this case. All segments should be subdivided according



**Fig. 11.** Form-finding results with the largest buckling capacity (left) and smallest number of unacceptable faces (right) in each training loop. The neural network gradually finds better solutions.

**Table 4**  
Parameter table for Fig. 11 left.

	$\bar{f}_1$	$\bar{f}_2$	$\bar{f}_3$	$\bar{f}_4$	$\bar{f}_5$	$\bar{f}_6$	$\bar{f}_7$	$m$	$f_b \times E3(kN)$
1-a	2	4	5	2	5	3	3	4	8.30
1-b	2	4	4	2	5	3	3	4	8.05
1-c	4	4	5	1	2	4	1	1	15.38
1-d	1	5	3	2	5	3	7	2	13.94
1-e	5	4	5	1	5	3	7	1	15.52
2-a	3	3	4	5	1	5	5	1	15.54
2-b	3	2	5	5	1	5	5	1	16.32
2-c	1	3	3	1	1	1	5	1	25.19
2-d	2	2	3	2	1	5	2	2	15.51
2-e	4	3	5	5	1	5	5	1	16.07
3-a	5	5	5	5	5	5	7	5	40.09
3-b	5	5	5	5	5	5	7	4	30.28
3-c	1	1	1	1	1	1	1	2	40.02
3-d	5	5	5	5	5	5	2	4	41.98
3-e	5	5	5	1	5	5	2	4	37.54
4-a	5	5	5	5	5	5	2	4	41.98
4-b	5	5	5	5	5	5	1	4	41.79
4-c	5	5	5	5	5	5	7	5	40.09
4-d	1	1	1	1	1	1	1	2	40.02
4-e	5	5	5	5	5	5	1	3	42.68
5-a	5	5	5	5	5	5	2	3	42.80
5-b	5	5	5	5	5	5	1	3	42.68
5-c	5	5	5	5	5	5	2	4	41.98
5-d	5	5	5	5	5	5	1	4	41.79
5-e	5	5	5	5	5	5	7	5	40.99

to the same  $m$  value to keep the edges touching each other, thus ensuring that the force diagram is correct as a closed polyhedron.

Therefore, in this subdivision system, the variables are the subdivision rules that are applied to the seven faces and the numbers of the segments into which the edges are subdivided. A series of eight numbers, for example 12345432, indicates the subdivision rules for the eight unit faces of the initial force polyhedron ( $\bar{f}_1:1; \bar{f}_2:2; \bar{f}_3:3; \bar{f}_4:4; \bar{f}_5:5; \bar{f}_6:4; \bar{f}_7:3$ ) and the segment counts ( $m:2$ ).

The final input is the boundary constraint for the form which is introduced as geometric constraints of nodes in the construction process of the form based on the geometric algorithm described by Akbarzadeh [1]. In this process, the position of each node is repeatedly updated to minimize the angle between the current member and the normal vector of the corresponding face in the force diagram while keeping the whole

**Table 5**  
Parameter table for Fig. 11 right.

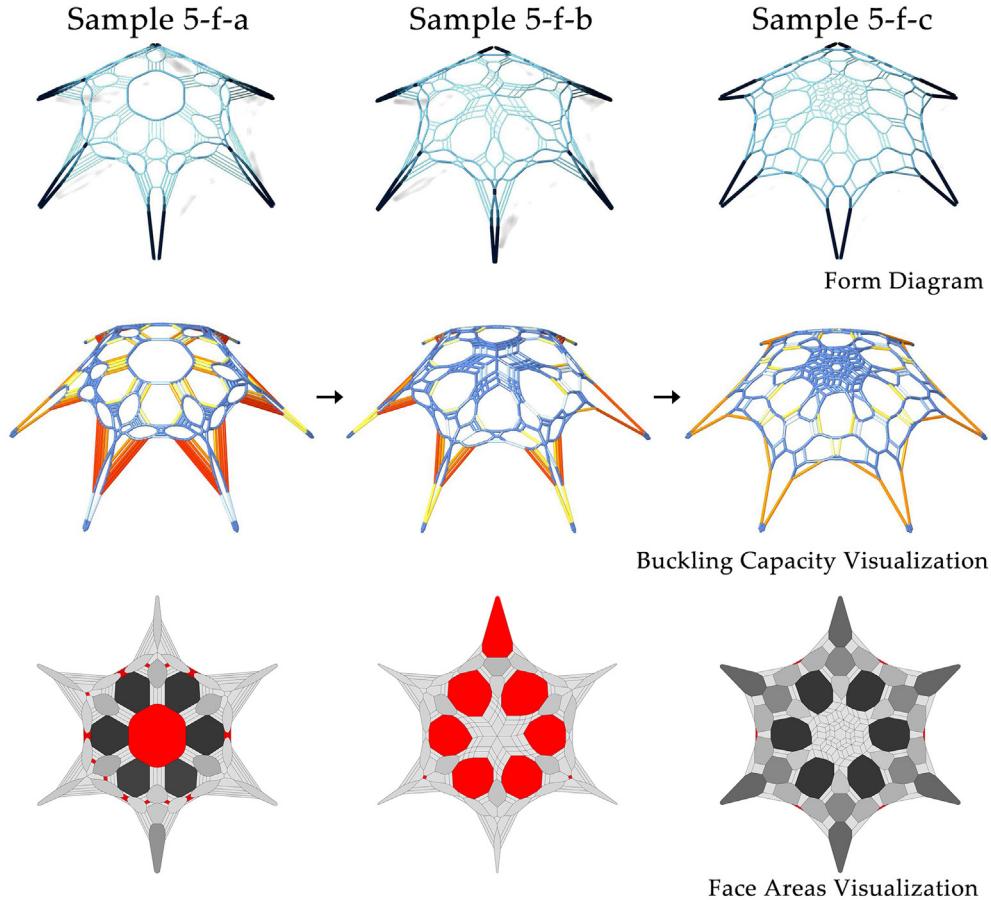
	$\bar{f}_1$	$\bar{f}_2$	$\bar{f}_3$	$\bar{f}_4$	$\bar{f}_5$	$\bar{f}_6$	$\bar{f}_7$	$m$	n
1-f	5	4	5	2	3	1	4	2	3
1-g	5	4	1	3	5	1	7	3	7
1-h	1	4	4	2	4	1	7	3	0
1-i	5	1	3	2	4	1	4	2	3
1-j	1	4	5	2	3	1	4	2	2
2-f	5	5	5	5	5	1	7	2	0
2-g	5	1	3	5	5	2	7	2	5
2-h	4	2	2	4	4	1	4	2	0
2-i	5	2	3	2	5	2	7	2	3
2-j	4	2	4	2	1	1	6	1	6
3-f	1	3	1	2	1	4	6	2	1
3-g	1	5	1	2	1	1	7	2	3
3-h	2	1	1	2	1	1	7	2	0
3-i	1	1	1	2	1	3	3	2	1
3-j	3	1	5	4	2	4	6	2	3
4-f	1	4	4	1	2	1	6	2	1
4-g	4	5	5	3	2	2	6	2	3
4-h	1	4	1	1	2	1	6	2	2
4-i	4	3	4	1	3	5	6	2	2
4-j	1	1	4	1	2	1	6	2	2
5-f	1	2	2	4	4	1	3	2	1
5-g	1	1	2	4	2	4	7	2	1
5-h	1	4	2	5	1	2	5	2	0
5-i	1	2	1	4	4	1	3	2	1
5-j	1	2	2	3	4	1	5	2	1

geometry inside the boundary constraint of a circle with a diameter of 20 m (Fig. 6d). This geometric operation ensures that the final form precisely matches its corresponding force diagram and touches the ideal boundary of the design space.

Thus, based on the fixed force boundary condition and the form boundary constraint as well as the different subdivision rules for each edge as variables, a series of funicular, topologically different, single-shell structural forms can be generated (Fig. 9). PolyFrame, a plug-in tool in Rhino, is used for the geometric computations [27].

## 2.2. Neural networks

The goal of this research is to evaluate the funicular forms generated from their corresponding force diagram based on their Euler buckling



**Fig. 12.** Form-finding results: examples of small (sample 5-f-a)/middle (sample 5-f-b)/large (sample 5-f-c) buckling capacity (figures in the middle).

performance and the size of their planar panels. Thus, a clear transformation of the form into understandable digital data for the computer is needed. To improve comparability with other forms, the structural data should represent the features of the form and be unique.

Initially, the subdivision rules can be set as eight input neurons with values of 1, 2, 3, 4, 5, 6, or 7 to indicate the subdivision rules for each face and how many segments into which each edge should be subdivided. However, since the values should be discontinuous as categorical variables, the one-hot encoding should be done to the input neurons. Thus, five or seven input neurons with binary values of either 0 or 1 are used to replace the single neuron of 1 to 5 (side subdivision and the segment count) or 1 to 7 (center subdivision). For example, a set of five neurons with the values of 0, 0, 1, 0, and 0 means the face is subdivided using the third rule. In this situation, only one neuron can have a value of 1, and the other neurons should all be 0. Thus, 42 ( $5 + 5 + 5 + 5 + 5 + 7 + 5$ ) neurons in total are used as the input layer structure for the neural network.

For the output data structure, a simple method is to directly use the values of the buckling capacity and the number of unacceptable faces as the two output neurons. However, we processed the normalized data collected as a real number between 0 and 1. This standardization ensures that the magnitude of all the data, including the input and the output, remains in the same range. Thus, the activation function, which describes the formula to map between the input data and the output data, is set using the Sigmoid function (Eq. (1)). In this formula,  $\hat{y}$  represents the predicted value,  $x$  represents the input value,  $w$ , and  $b$  are the weight and bias parameters, which the network will determine during training. In addition, the loss function is a mean squared error (MSE) function (Eq. (2)) by default.

$$\hat{y} = \text{Sigmoid}\left(w * x + b\right) = \frac{1}{1 + e^{-(w*x+b)}} \quad (1)$$

$$\text{Loss}(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (2)$$

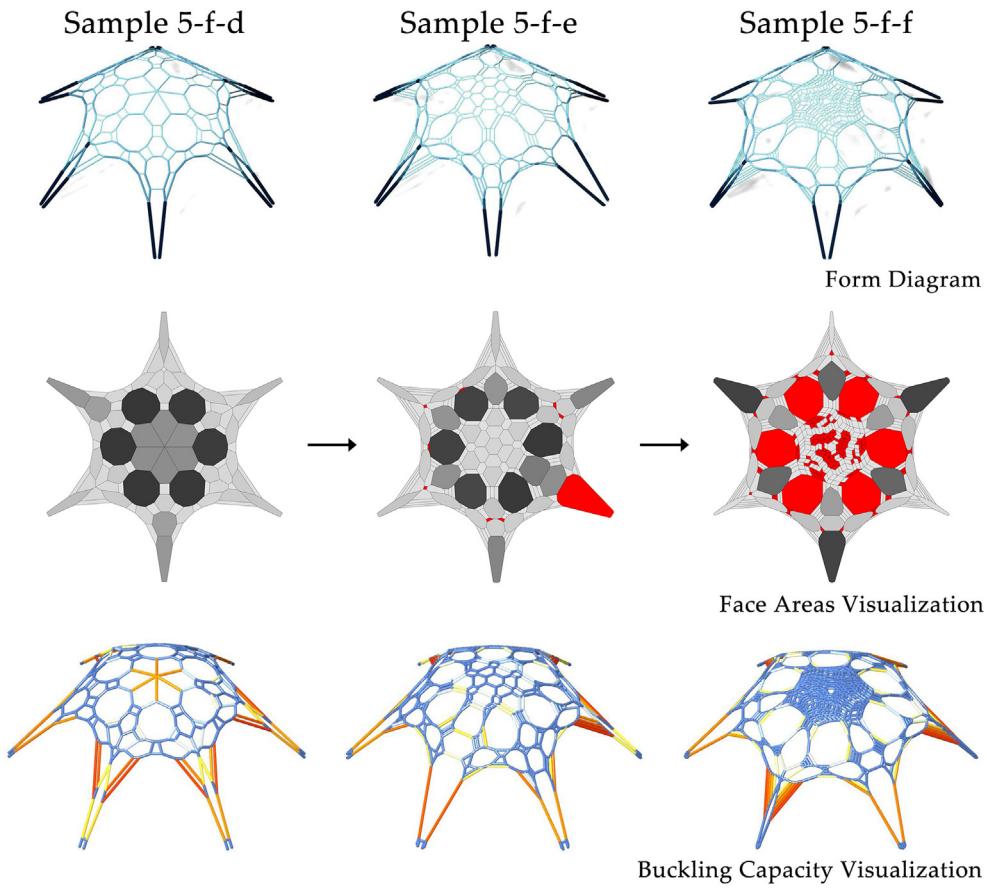
Finally, the number of hidden layers in the neural network should be defined. This depends on the complexity of the problem. A larger network with more hidden layers does not guarantee better performance. Thus, a 5-fold cross-validation test with 1500 randomly generated instances, which is commonly used to test the performance of a machine learning algorithm, is implemented with Tensorflow and given to the neural networks with different numbers of layers. To simplify the loss function to be better understood, the accuracy function (Eq. (3)) is used to evaluate the accuracy of the outputs.

$$\text{Accuracy}(y, \hat{y}) = 1 - |\hat{y} - y| \quad (3)$$

To test the effect of layer sizes in the neural network on the overall performance of the surrogate model, we performed the validation test. Table 1 presents the median accuracy of different neural networks in the 5-fold cross-validation test, using a dataset of 1500 randomly generated instances. It can be seen that the artificial neural network with five layers has the highest accuracy compared to the other neural networks. Thus, the neural network with five layers is chosen as the final setting.

### 2.3. Iterative training and targeted sampling

With the data structure, the activation and loss functions, and the neural network settings described above, after training, the neural network model should predict the two values within milliseconds, given



**Fig. 13.** Form-finding results: examples of small (sample 5-f-d)/middle (sample 5-f-e)/large (sample 5-f-f) number of unacceptable faces (figures in the middle).

**Table 6**  
Parameter table for Figs. 12 and 13.

	$\bar{f}_1$	$\bar{f}_2$	$\bar{f}_3$	$\bar{f}_4$	$\bar{f}_5$	$\bar{f}_6$	$\bar{f}_7$	$m$	$f_b \times E_3(\text{kN})$	$n$
5-f-a	1	4	4	2	3	2	3	5	6.71	28
5-f-b	4	3	2	5	4	3	1	3	12.93	9
5-f-c	5	5	5	5	5	5	2	3	42.80	6
5-f-d	1	4	2	3	1	2	5	2	12.71	0
5-f-e	1	5	1	2	3	3	7	4	10.45	9
5-f-f	5	2	5	4	5	3	2	5	12.71	171

a specific subdivision rule.

Thus, to test its accuracy in different domains to see whether the trained model can correctly predict the buckling capacity and the number of unacceptable faces among different ranges of the solution space, all possible subdivision rules were fed to the trained neural network, and the predicted values were then sorted. According to the sorted values, 300 samples with the smallest/equally distributed/largest buckling capacity/number of unacceptable faces were extracted and sent back to the geometric process to verify its accuracy.

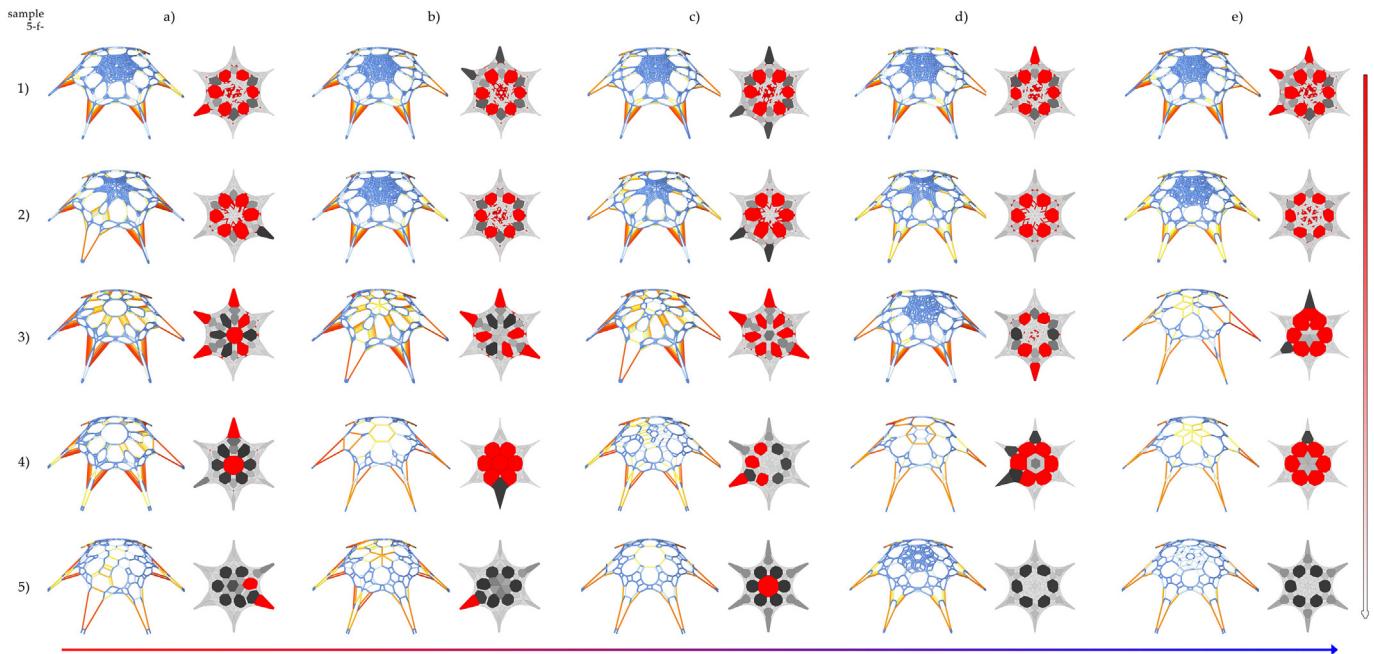
However, according to the geometric results, the accuracy of the three sets of 100 subdivision rules are very unsatisfying: They are only 79.6%, 71.6%, and 20.4% for the buckling capacity and 98.3%, 59.2%, and 4.2% for the number of unacceptable faces, much smaller than the expected accuracy. Thus, the current neural network model performs well in a random domain but poorly in the domain of extreme values. This phenomenon is caused by missing key instances in the training data. Since all instances were generated randomly, the model over-fit specific values according to the training set. Thus, it lost the ability to predict extreme values that were not included in the training set. We can conclude that targeted sampling (active sampling) is needed rather

than random sampling.

To solve this problem, an iterative training method is proposed, adding the ground truth values of the selected subdivision rules into the training dataset in each training loop (Fig. 10). First, the initial training data containing 175 instances are used to train the neural network model in the first loop. The initial subdivision rules are generated by changing one subdivision rule in the ruleset while keeping the other seven rules the same. For example, rule “A = {1,1,1,1,1,1,2}” and rule “B = {2,2,2,2,2,2,2}” are contained in the first training set. The reason behind setting up the initial input values in this way is that it generates training samples that have equally separated subdivision rules in the solution space; thus, the output values should have a better chance of also being equally distributed. The learning of these instances should help the neural network to obtain an overview of knowledge of the whole solution space.

Based on the trained model in the first training loop, all possible series of subdivision rules are predicted using the surrogate model, and the output values are sorted. Next, 150 rules with the smallest/equally distributed/largest buckling capacity values and 150 rules with the smallest/equally distributed/largest number of unacceptable faces are selected as the instances waiting for the geometric verification. The reason behind selecting these 300 rules is that the 200 smallest and largest values boost the information in the domain of the extreme value area, and the 100 equally distributed values add necessary information for the overall prediction. Thus, after the feedback from the geometric operation has been received, the ground truth values and the selected subdivision rules are added to the training dataset, and a new neural network model in the next loop will be trained.

After the iterative method had been executed five times, the accuracy of the neural network model reached 97% and 98.6% in the randomly generated validation set. The accuracy of the smallest/equally



**Fig. 14.** Form finding: matrix of results with different ranges of the buckling capacity and the number of unacceptable faces.

**Table 7**  
Parameter table for Fig. 14.

	$\bar{f}_1$	$\bar{f}_2$	$\bar{f}_3$	$\bar{f}_4$	$\bar{f}_5$	$\bar{f}_6$	$\bar{f}_7$	$m$	$f_b \times E3(kN)$	n
1-a	4	4	2	1	4	5	2	5	11.89	133
1-b	4	4	3	5	5	3	2	5	12.38	146
1-c	5	3	4	5	3	5	2	5	12.46	163
1-d	3	3	2	5	3	1	2	5	12.59	131
1-e	4	4	3	5	5	5	2	5	12.78	149
2-a	2	5	4	4	2	1	1	5	10.70	99
2-b	4	4	2	1	4	5	2	5	11.70	114
2-c	5	3	3	1	2	5	1	5	11.90	108
2-d	1	1	1	1	1	1	1	5	22.97	84
2-e	1	1	1	1	1	1	2	5	23.55	90
3-a	4	4	3	5	5	5	3	5	7.20	24
3-b	4	5	2	5	5	4	5	5	7.34	38
3-c	4	5	2	5	5	3	6	5	7.44	37
3-d	5	3	4	4	1	2	2	4	11.35	30
3-e	2	2	4	5	3	1	1	1	14.68	9
4-a	3	2	3	5	3	1	3	4	8.96	9
4-b	5	3	3	4	3	3	2	1	13.99	10
4-c	3	2	2	1	1	5	2	2	14.97	2
4-d	3	2	2	1	1	5	4	1	15.05	7
4-e	3	2	2	1	2	3	7	1	26.87	6
5-a	4	5	1	4	4	4	7	2	12.55	2
5-b	4	2	1	2	4	5	5	2	12.94	1
5-c	1	1	1	1	1	1	3	2	33.39	1
5-d	1	1	1	1	1	1	2	3	36.39	0
5-e	1	1	1	1	1	1	2	2	38.75	0

distributed/largest results is 95.9%/99.2%, 95.1%/98.2%, and 98.5%/99.3%, significantly higher than in the previous model (i.e., with random sampling). The whole process, including geometric operations and neural networks training and testing, took around 15.31 h, which falls within an acceptable time limit. Using the trained model, the prediction of the two values from a single set of the given subdivision rules takes less than 1 millisecond to complete, while an exhaustive search for all solutions would take around 6000 h, and the genetic algorithm for one single solution would take around 10 h.

### 3. Results

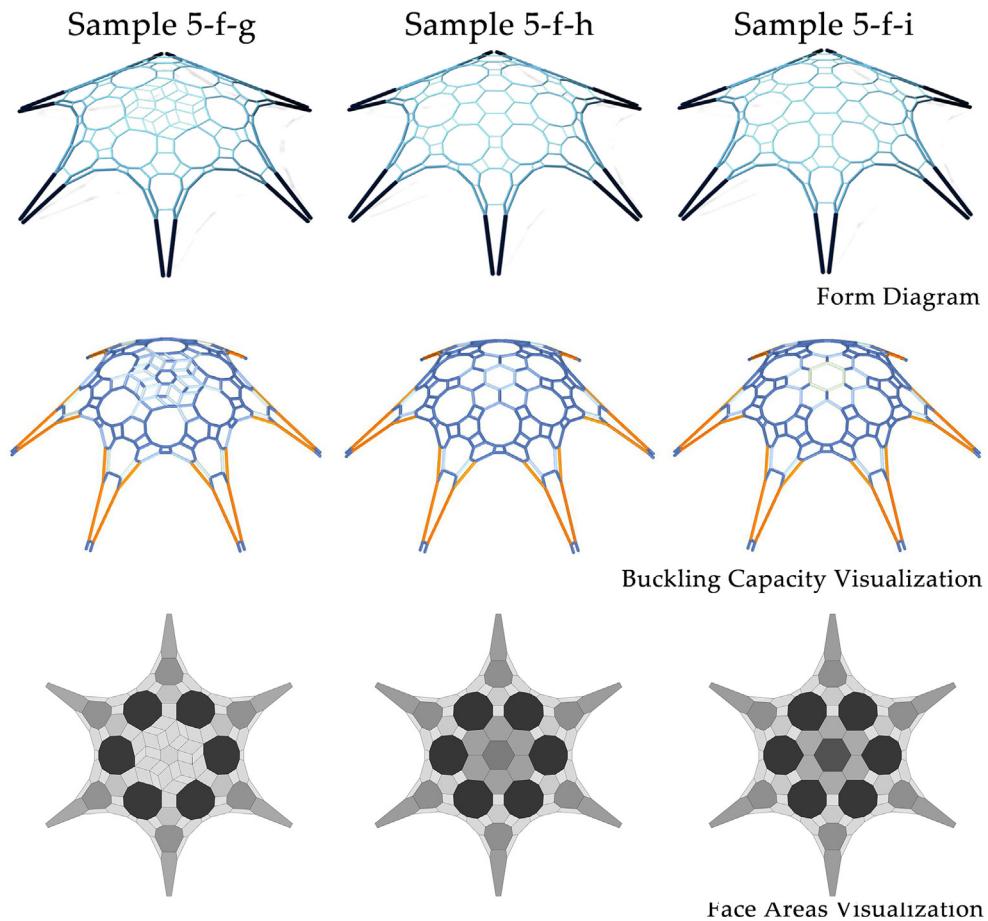
#### 3.1. Model training

Based on the neural network settings and the iterative training method described above, neural network models in different loops were built and trained. Table 2 presents the testing accuracy in each training loop. The sample solutions were iteratively updated and added to the training dataset in each loop. The randomly generated 300 instances were used for the random testing, while the 300 verified instances were generated as the testing data for the results in the domains.

It is clear that in the first training loop, the accuracy was lower than expected. The initial data of the 175 instances were too small for the model to learn; this could have easily caused the over-fitting problem. As the program looped, the accuracy in loop 2 increased. In loops 3, 4, and 5, the looping did not help much in increasing the accuracy of predicting randomly selected instances, but the prediction of extreme values was improved. The models in loop 5 reached the ideal level of accuracy for both the randomly testing data and the selected extreme data.

Moreover, Table 3 presents the required time for each training loop, including generating the training instances, and the required time of the optimization program using the genetic algorithm. In loops 3, 4, and 5, the best solutions with the largest buckling capacity and the smallest number of unacceptable faces are found in the predicted 100 extreme values. The required time for loop 3 is 8.63 h, while the required time for a genetic algorithm to find those two solutions is 9.56 h. The neural network is faster than the genetic algorithm in reaching the same level of effect, and it is a surrogate model that can predict all values, while the genetic algorithm only finds the single best values.

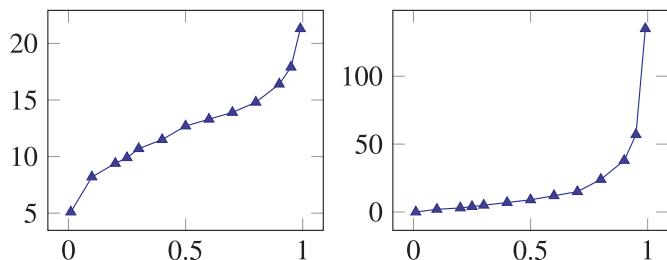
Also, Fig. 11 depicts the forms of the geometries found with extreme values in each loop. As the training loop proceeds, the predicted maximum buckling capacity value keeps increasing, while the predicted minimum number of unacceptable faces value keeps decreasing. For example, the buckling capacity values for samples 1-a, 2-a, 3-a, 4-a, and 5-a are 8.3 E3(kN), 15.54 E3(kN), 40.09 E3(kN), 41.98 E3(kN), and 42.8 E3(kN), respectively. Furthermore, the number of unacceptable faces values for samples 1-g, 2-g, 3-g, 4-g, and 5-g are 7, 5, 3, 3, and 1, respectively. That means the neural network gradually finds better



**Fig. 15.** Form finding: larger buckling capacity + smaller number of unacceptable faces (ideally 0).

**Table 8**  
Parameter table for Fig. 15.

	$\bar{f}_1$	$\bar{f}_2$	$\bar{f}_3$	$\bar{f}_4$	$\bar{f}_5$	$\bar{f}_6$	$\bar{f}_7$	$m$	$f_b \times E3(kN)$	n
5-f-g	1	1	1	1	1	1	2	2	38.75	0
5-f-h	1	1	1	1	1	1	6	2	35.48	0
5-f-i	1	1	1	1	1	1	7	2	35.45	0



**Fig. 16.** Data distribution of the buckling capacity (left) and the number of unacceptable faces (right).

solutions with a larger buckling capacity and a smaller number of unacceptable faces (Table 4).

In loop 3, the global maximum buckling capacity solution “A = {5,5,5,5,5,2,3}” first appears in the top 50 samples, and in loop 5, it becomes the best prediction as sample 5-a. For the number of unacceptable faces, even in loop 1, the model finds at least one solution with 0 unacceptable faces, for example, sample 1-h, and the overall tendency of the predicted values runs toward 0 as the training proceeds.

That means the training was successful, and the final model in loop 5 can be used as a surrogate model for the further form-finding process. Therefore, we can conclude that the trained model has very high accuracy in the whole solution space. This surrogate model is trustworthy for the predictions (Table 5).

### 3.2. Form finding

Next, using the trained model, the form-finding process can proceed. In this process, given all the combinations of the subdivision rules, the trained surrogate model can quickly estimate the buckling capacity and the number of unacceptable faces, almost in real time.

First, to test the ability of the trained neural network in the form-finding process, examples with a small/middle/large buckling capacity and number of unacceptable faces are found by the neural network model. Fig. 12 presents three examples for buckling capacity, while Fig. 13 presents three examples for the number of unacceptable faces. The trained network successfully predicts the two output values and finds the cases with a specific range of buckling capacity and number of unacceptable faces.

For example, according to Table 6, the buckling capacity for sample 5-f-a is 6.71 E3(kN), much smaller than the buckling capacity of 42.8 for sample 5-f-c E3(kN). Thus, if the user wants to find a solution with low buckling capacity, sample 5-f-a will be recommended. In addition, the number of unacceptable faces for sample 5-f-f is 171, much larger than that for sample 5-f-d of 0. Similarly, if the user wants a form that contains a large number of unacceptable faces for comparison, sample 5-f-f will be recommended.

Fig. 14 presents the resulting matrix of the form-finding process, where the vertical axis presents the examples with different ranges of

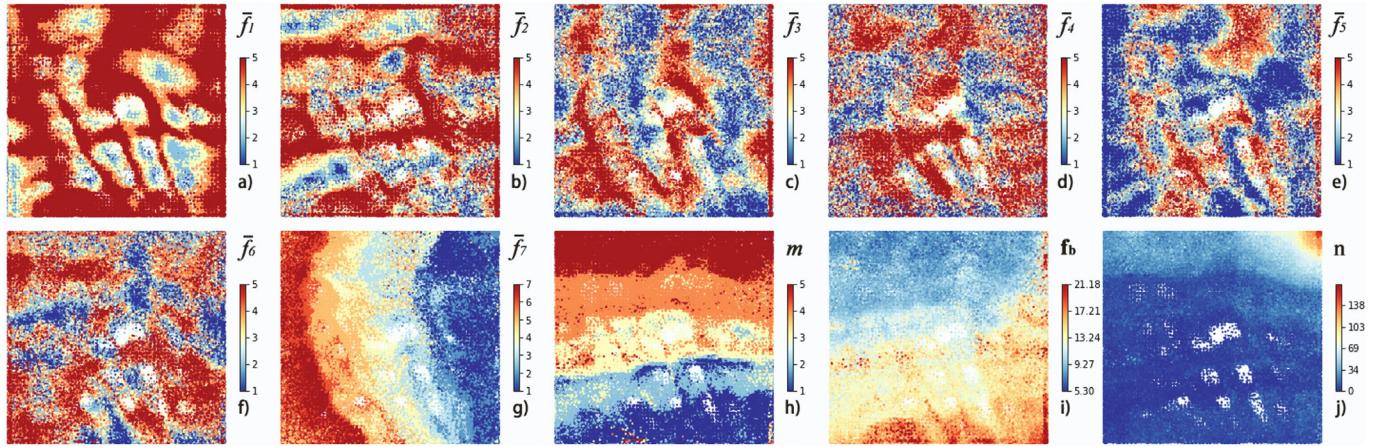


Fig. 17. The nonlinear effects of different design variables (i.e., subdivision rules) on the final structural performance measures, using self-organizing maps.

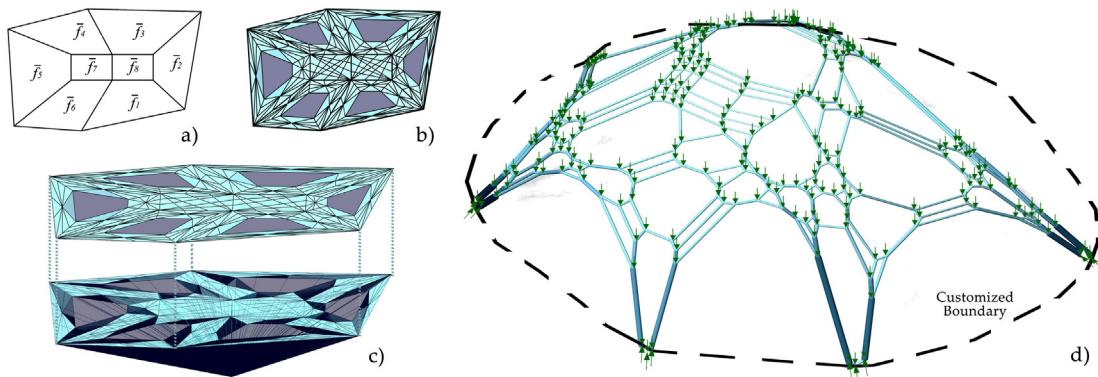


Fig. 18. Asymmetric condition: user-defined original force units and form constraint.

the buckling capacity and the horizontal axis presents the examples with different ranges of the number of unacceptable faces. For example, sample 1-a shows a form with a small buckling capacity and a large number of unacceptable faces, which can be regarded as one of the worst results, while sample 5-e shows one of the best results, a form with a large buckling capacity and a small number of unacceptable faces. This resulting matrix demonstrates that the form-finding method meets the requirement for the user to be able to find the solutions with any specific range of buckling capacity and number of unacceptable faces (Table 7).

However, in our design case, ideally, the number of unacceptable faces should be 0 so that the form can be built in reality. Therefore, based on the output values, we set the maximum acceptable number of unacceptable faces as 0 while sorting all filtered results based on the buckling capacity. Fig. 15 presents the top three results the neural network finds. Their buckling capacity values are more significant than those of other forms, and they are constructive in reality without any unacceptable faces (Table 8).

Therefore, based on the returning results, the neural network successfully finds multiple solutions within the expected time limit. By changing the thresholds of the two output values, designers can be presented with different recommended solutions and can thus obtain the ideal forms with a specific buckling capacity and number of unacceptable faces.

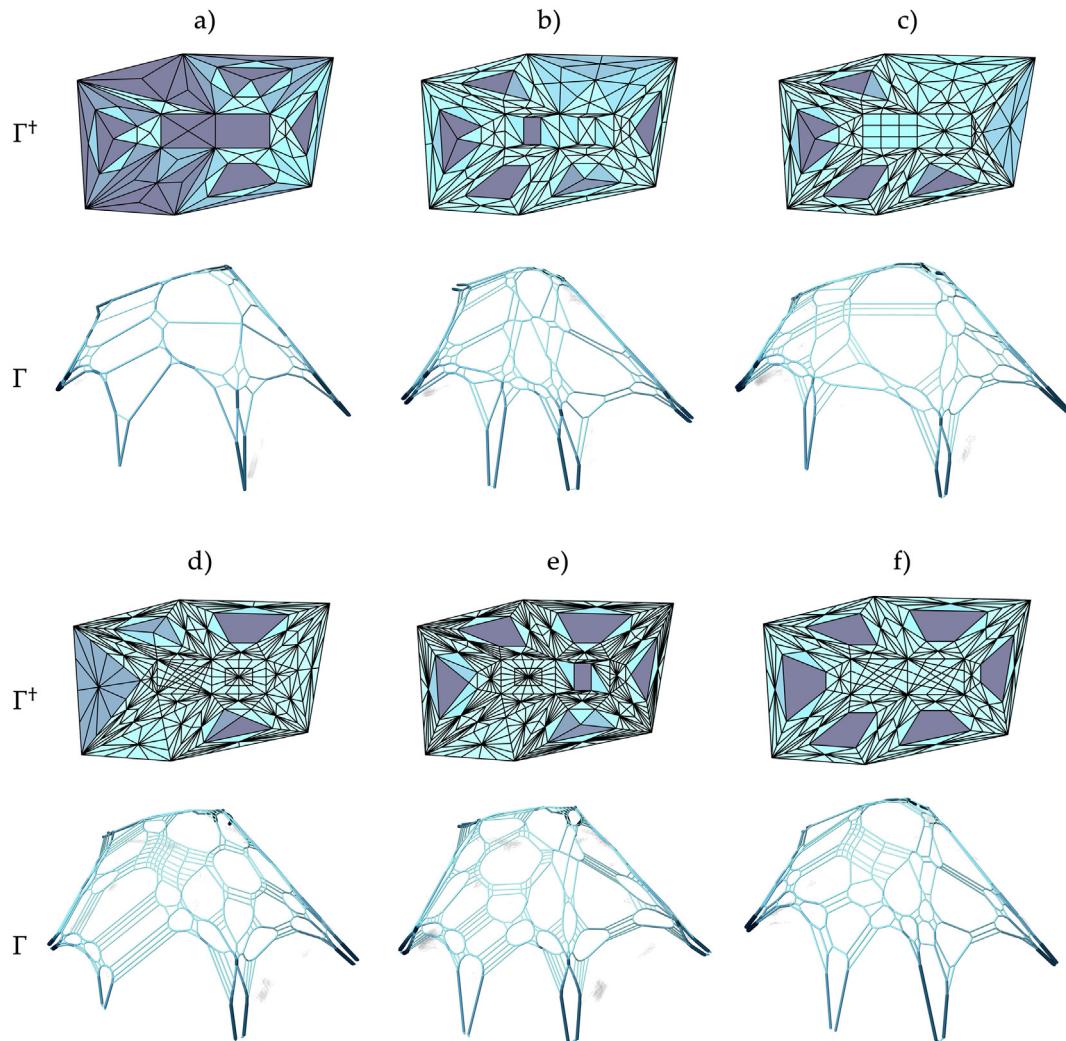
### 3.3. Data analysis

With the trained model, the entire solution space can be visualized in a matter of milliseconds. Fig. 16 illustrates the data distribution of the buckling capacity and the number of unacceptable faces. It can be

inferred that among all the solutions, the buckling capacity has a distribution close to a normal distribution, as the cumulative distribution is closer to a 45-degree line, while most of the solutions have several unacceptable faces larger than 0. Since the number of unacceptable faces is caused by both faces that are too large and faces that are too small, subdividing the force diagrams and generating more members in the forms does not guarantee a smaller number of unacceptable faces, because it subdivides the large faces while producing small faces. Thus, a further subdivision is not always suitable for all cases.

To obtain a closer look into the solution space, if we are now interested in analyzing the relationships between the subdivision rules on each face and the final two performance measures, we have a 10-dimensional space (six for subdivision rules of the side units, one for subdivision rules of the center units, one for the subdivision count, one for the buckling capacity, and one for the number of unacceptable faces). The dataset in the 10-dimensional space is then transformed to a 2-dimensional space and plotted in Fig. 17 using self-organizing maps (implemented in Python [24]), a powerful nonlinear manifold learning and dimensionality reduction method that can visualize a high-dimensional space via a lower-dimensional space (usually two dimensions) [25].

According to the result, first, the inverse relation between buckling capacity and the “rule in initial subdivision” (Fig. 17i and h) can be found. That means subdividing the initial force diagram with more segments in the majority of state-space decreases. However, some instances still have a larger buckling capacity with a larger subdivision count, for example, the upper right corner in Fig. 17i. Secondly, the importance of “rule in initial face 7” (Fig. 17g) in comparison to the side units (Fig. 17a to f) can be observed. The distribution pattern of “rule in initial face 1 to 6” is more random than the pattern of “rule in



**Fig. 19.** Asymmetric condition: example forms.

initial face 7,” which indicates that the subdivision rule for the center unit plays a more important role in deciding the structure and performance and the constructability. Furthermore, the interplay between some of the side units indicates exciting relations. For example, “rules in initial face 1, 3, and 5” (Fig. 17a, c, and e) are organized with negative local correlations (i.e., with opposite colors) of one another, while “rule in initial face 2, 4, and 6” (Fig. 17b, d, and f) have locally negative co-relations. This phenomenon demonstrates that the results are usually similar when different subdivision rules are applied to the crossing faces (for example, 1, 3, and 5 or 2, 4, and 6).

#### 3.4. User-defined force boundary and form constraint

In addition to the fixed force boundary and form constraint described above, the neural network can learn and predict the geometric process and results with customized force boundaries and form constraints. The same procedure can be applied to the cases to build another surrogate model with different variables of the subdivision rules.

Fig. 18 introduces an example of the form generation under a user-defined asymmetric force boundary and form constraint. The initial force pattern contains six side units and two center units (Fig. 18a), which is different from the previous case. Moreover, each face can be geometrically different but follow the same subdivision rules of the five side subdivision rules and the seven center subdivision rules. The form constraint (Fig. 18d) can be any closed planar curve that the legs of the

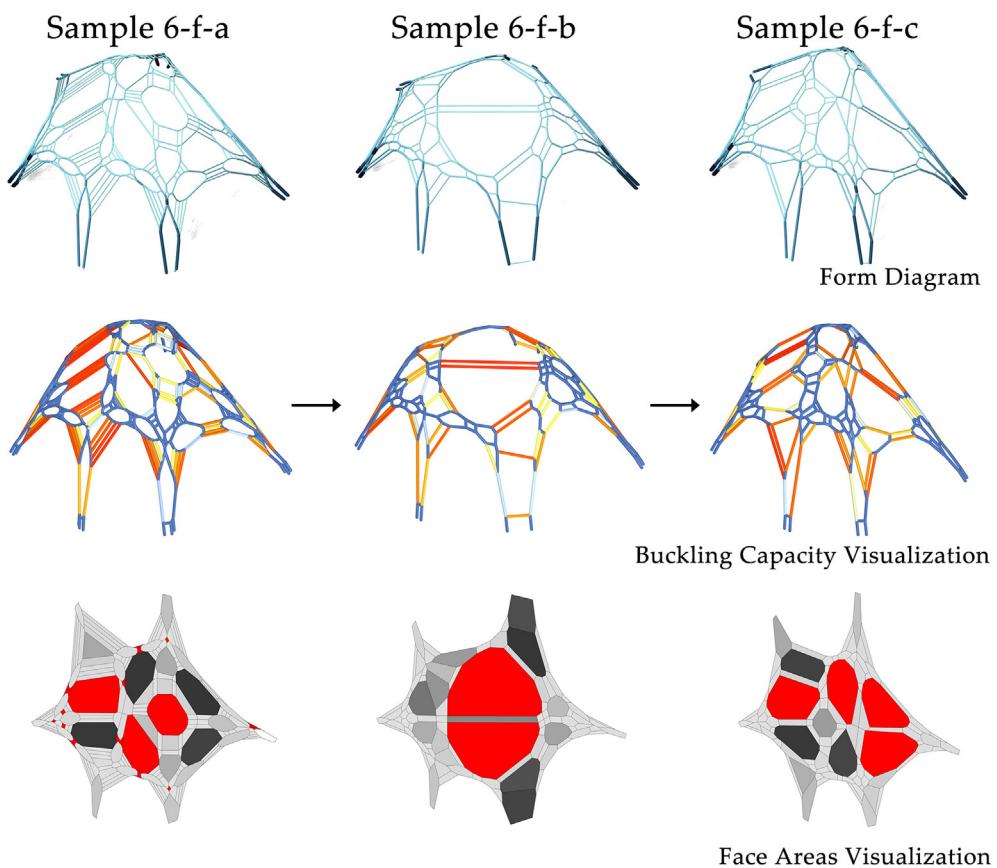
form are forced to touch during the generation process.

With all the settings above, the program will automatically load these prerequisites and the subdivision rules and generate the forms. Fig. 19 presents examples of the force diagrams and the corresponding forms. Since the force boundary and form constraint are asymmetric and different from in the previous case, the structural performance and the constructability are also very different.

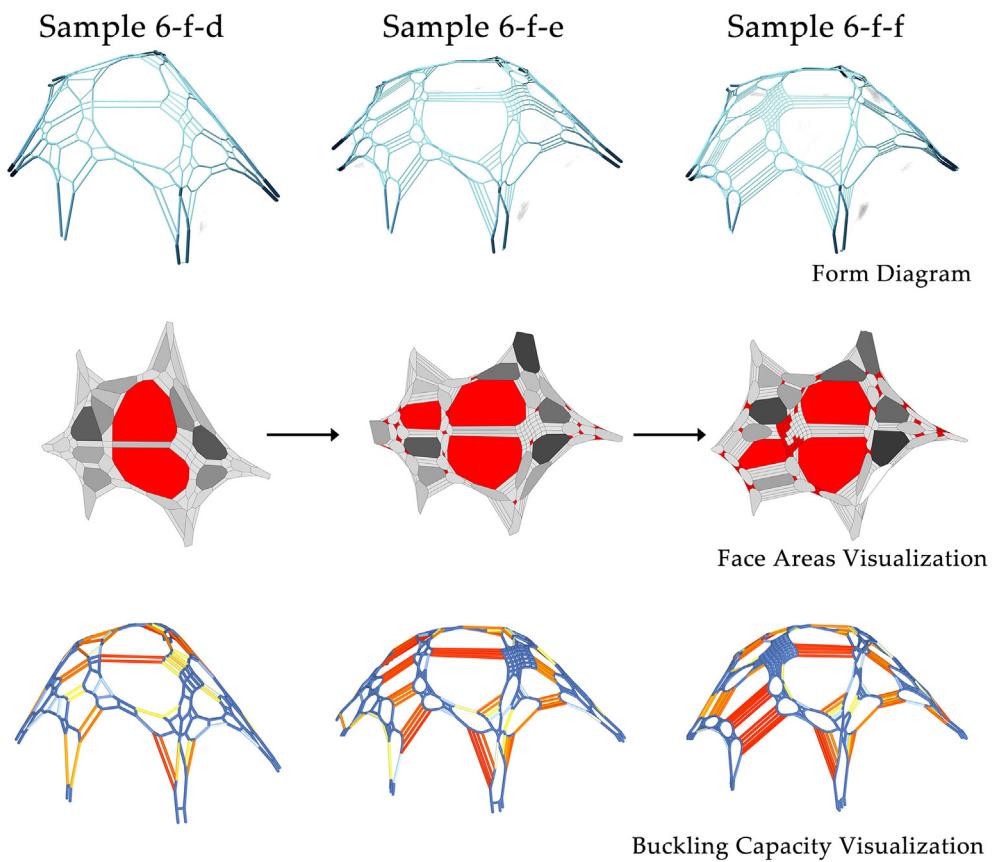
When the same process of iteratively training another neural network model and applying the final trained model in the form-finding step is followed, forms with a specific range of buckling capacity and number of unacceptable faces can also be found. Fig. 20 presents the examples with small/middle/large buckling capacity, and Fig. 21 presents the examples with small/middle/large number of unacceptable faces. Sample 6-f-a has a smaller buckling capacity than sample 6-f-c, while sample 6-f-d has a smaller number of unacceptable faces than sample 6-f-f. This result further demonstrates the ability of the neural network to predict and find forms with any range of values (Table 9).

Moreover, with the setting of a threshold for the number of unacceptable faces, forms with larger buckling capacity are shown in Fig. 22. However, in this case, the minimum number of unacceptable faces is two, which means all solutions should have at least two faces that exceed the range of constructive areas. Thus, with a threshold value of two, samples 6-f-g, 6-f-h, and 6-f-i are found to have a larger buckling capacity (Table 10).

Through the analysis of the solution space, in this case, a



**Fig. 20.** Asymmetric condition: form-finding results: examples of small (sample 6-f-a)/middle (sample 6-f-b)/large (sample 6-f-c) buckling capacity.



**Fig. 21.** Asymmetric condition: form-finding results: examples of small (sample 6-f-d)/middle (sample 6-f-e)/large (sample 6-f-f) number of unacceptable faces.

**Table 9**  
Parameter table for Figs. 20 and 21.

	$\bar{f}_1$	$\bar{f}_2$	$\bar{f}_3$	$\bar{f}_4$	$\bar{f}_5$	$\bar{f}_6$	$\bar{f}_7$	$\bar{f}_8$	m	$f_b \times E3(\text{kN})$	n
6-f-a	3	3	2	3	4	3	5	4	4	3.67	28
6-f-b	2	1	2	1	2	4	7	2	2	5.09	2
6-f-c	2	4	1	2	5	2	4	5	2	10.05	4
6-f-d	1	2	1	4	2	1	7	1	2	9.21	2
6-f-e	1	4	1	5	3	1	6	2	4	5.38	32
6-f-f	5	2	4	5	1	5	2	5	5	8.51	93

subdivision segment count of two would still produce a better result that is more constructive and structurally stronger. Therefore, this machine learning assisted evaluation method is adaptive for a variety of situations in the form finding with single-shell structures.

#### 4. Conclusion

3D graphic statics provides a robust geometry-based structural design and analysis method to generate funicular forms based on the corresponding force diagrams. However, the large solution space among a variety of sets of subdivision rules requires a tremendous amount of computing power to find the form with the best structural performance.

Meanwhile, machine learning methods (e.g., neural networks) provide an alternative approach to building a surrogate model to predict the structural performance and the constructability of a given set of subdivision rules. In this work, we demonstrate how, by using an iterative process, a neural network model can be trained to accurately predict the nonlinear relations between different subdivision rules (as

**Table 10**  
Parameter table for Fig. 22.

	$\bar{f}_1$	$\bar{f}_2$	$\bar{f}_3$	$\bar{f}_4$	$\bar{f}_5$	$\bar{f}_6$	$\bar{f}_7$	$\bar{f}_8$	m	$f_b \times E3(\text{kN})$	n
6-f-g	1	1	1	1	1	1	7	7	2	5.62	2
6-f-h	2	2	2	2	2	2	7	7	2	8.35	2
6-f-i	1	2	1	4	2	1	7	1	2	9.21	2

the input variables to the 3D graphic statics procedure) and two structural and construction-related performance measures of the geometric forms (the buckling capacity and the number of unacceptable faces). After the surrogate model has been trained and given all the possible subdivision rules, the two performance measures can be estimated within seconds; this renders form finding in the whole solution space practical. Analytical plots can be drawn using the predicted values, visualizing and analyzing the solution space.

Unlike in a traditional optimization process, such as using the genetic algorithm, which only finds a set of (local) optimum solutions, in the machine learning assisted evaluation and form-finding methodology, the trained surrogate learns the relationship between the input and the outputs, which enables the designer to interactively search the entire solution space; thus, it provides more flexible methods for multi-objective assessments of the solution space. By changing the thresholds of the desired performance measure values, designers can easily be presented with different recommended solutions and can thus obtain the ideal solutions within specific ranges of the output values.

Therefore, in the future, the tendency toward design cooperation between the human and the machine will become more evident. The machine will assist the design process not only in simple repetitive work but also in creative work by learning the design examples from the

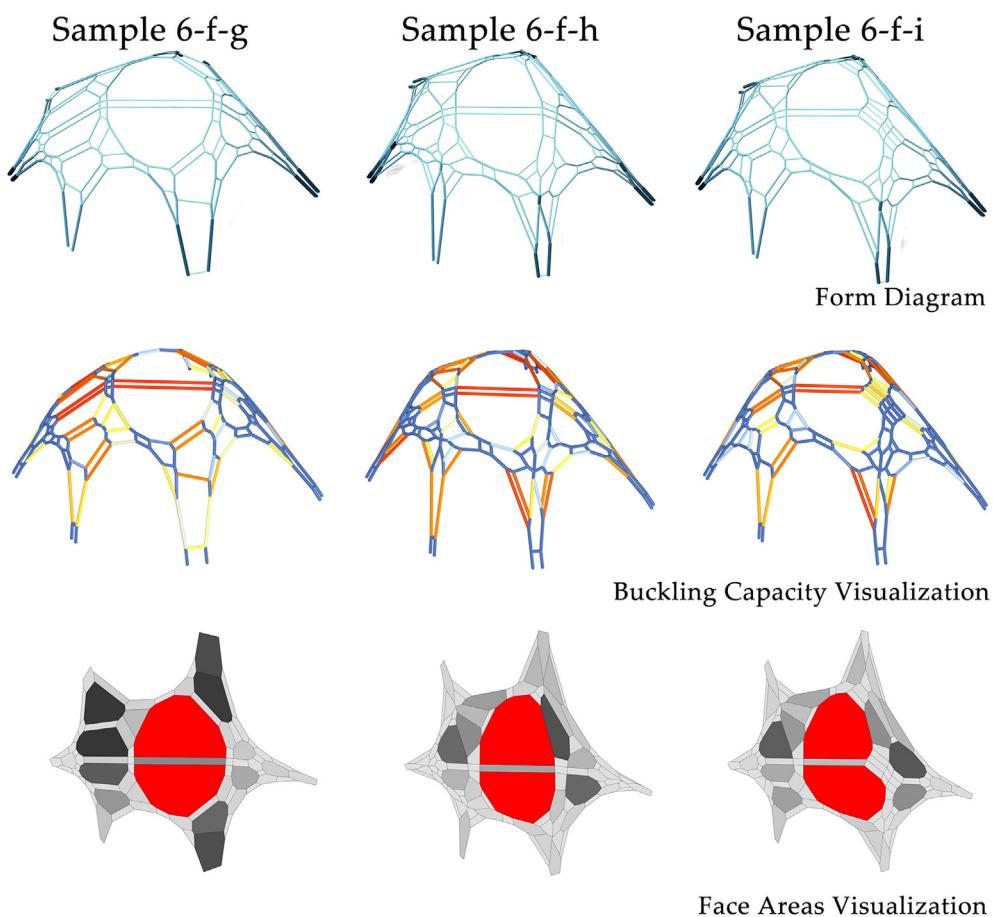


Fig. 22. Asymmetric condition: form finding: larger buckling capacity + smaller number of unacceptable faces (smallest 2).

human. The next step of this research is to extend the usage of machine learning to provide a more general framework with different setups, rules, boundaries, constraints, and topology, thus providing a real-time feedback system to advise designers in their choice of design strategies.

## Declaration of competing interest

There is no conflict of interest.

## Acknowledgement

The implementation environments of this research include: PolyFrame for running 3DGS algorithm in Rhino [27]; Tensorflow for building and training the neural networks; and SOMPY [24] for running the analysis of Self Organizing Map.

## References

- [1] M. Akbarzadeh, 3D Graphic Statics Using Polyhedral Reciprocal Diagrams (PhD thesis), ETH Zürich, Zürich, Switzerland, 2016.
- [2] M. Akbarzadeh, T. Van Mele, P. Block, Compression-only form finding through finite subdivision of the external force polygon, in: J.B. Obrebski, R. Tarczewski (Eds.), Proceedings of the IASS-SLTE Symposium 2014, Brasilia, Brazil, 2014.
- [3] M. Akbarzadeh, T. Van Mele, P. Block, Spatial compression-only form finding through subdivision of external force polyhedron, Proceedings of the International Association for Shell and Spatial Structures (IASS) Symposium, Amsterdam, August 2015.
- [4] M. Akbarzadeh, T. Van Mele, P. Block, On the equilibrium of funicular polyhedral frames and convex polyhedral force diagrams, Comput. Aided Des. 63 (2015) 118–128, <https://doi.org/10.1016/j.cad.2015.01.006>.
- [5] M. Akbarzadeh, T. Van Mele, P. Block, Three-dimensional graphic statics: initial explorations with polyhedral form and force diagrams, International Journal of Space Structures 31 (2016) 217–226.
- [6] Zeynep Aksöz, Clemens Preisinger, An interactive structural optimization of space frame structures using machine learning, Proceedings of the Design Modelling Symposium 2019, 2019, pp. 18–31 (Berlin, Germany).
- [7] Charles Audet, J. Denni, Douglas Moore, Andrew Booker, Paul Frank, A surrogate-model-based method for constrained optimization, 8th Symposium on Multidisciplinary Analysis and Optimization, 2000, p. 4891.
- [8] Joshua Bard, Ardavan Bidgoli, Wei Wei Chi, Image classification for robotic plastering with convolutional neural network, Proceedings of Robotic Fabrication in Architecture 2018, 2018, pp. 3–15.
- [9] L.L. Beghini, J. Carrion, A. Beghini, A. Masurek, W.F. Baker, Structural optimization using graphic statics, Struct. Multidiscip. Optim. 49 (3) (2013) 351–366.
- [10] P. Block, J. Ochsendorf, Thrust network analysis: a new methodology for three-dimensional equilibrium, Journal of the International Association for Shell and Spatial Structures 48 (3) (2007) 167–173.
- [11] M. Bolhassani, M. Akbarzadeh, M. Mahnia, R. Taherian, On structural behavior of the first funicular polyhedral frame designed by 3d graphic statics, Journal of Building Engineering (2017), <https://doi.org/10.1016/j.jstruc.2018.02.002> (Under review).
- [12] R.H. Bow, Economics of Construction in Relation to Framed Structures, Spon, London, 1873.
- [13] Giulio Brugnaro, Sean Hanna, Adaptive robotic carving, Robotic Fabrication in Architecture, Art and Design, Springer, 2018, pp. 336–348.
- [14] L. Cremona, Graphical Statics: Two Treatises on the Graphical Calculus and Reciprocal Figures in Graphical Statics, Clarendon Press, Oxford, 1890.
- [15] K. Culmann, Die Graphische Statik, Verlag Meyer & Zeller, Zürich, 1864.
- [16] Renaud Danhaive, Caitlin Mueller, Structural metamodelling of shells, Proceedings of IASS Annual Symposia, 2018.25 International Association for Shell and Spatial Structures (IASS), 2018, pp. 1–4.
- [17] C. Fivet, D. Zastavni, Constraint-based graphic statics: new paradigms of computer-aided structural equilibrium design, Journal of the International Association of Shell and Spatial Structures 54 (4) (2013) 271–280.
- [18] Lukas Fuhrmann, Vahid Moosavi, Patrick Ole Ohlbrock, Pierluigi Dacunto, Data-driven design: exploring new structural forms using machine learning and graphic statics, Proceedings of IASS Annual Symposia, International Association for Shell and Spatial Structures (IASS), 2018, pp. 1–8.
- [19] A.T. Ghomi, M. Bolhassani, Nejur, M. Akbarzadeh, The effect of subdivision of force diagrams on the local buckling, load-path and material use of founded forms, Proceedings of IASS Symposium 2018, MIT, Boston, USA, 2018, pp. 1–8.
- [20] K.C. Lam, C.Y. Yu, A multiple kernel learning-based decision support model for contractor pre-qualification, Autom. Constr. 20 (5) (2011) 531–536, <https://doi.org/10.1016/j.autcon.2010.11.019> (ISSN 0926-5805), <http://www.sciencedirect.com/science/article/pii/S0926580510001986>.
- [21] Andrew Liew, R. Avelino, Vahid Moosavi, Tom Van Mele, Philippe Block, Optimising the load path of compression-only thrust networks through independent sets, Struct. Multidiscip. Optim. 60 (1) (2019) 231–244.
- [22] J.C. Maxwell, On reciprocal figures and diagrams of forces, Philosophical Magazine and Journal Series 4 (27) (1864) 250–261.
- [23] Robert Mcneel and Associates, Grasshopper: graphical algorithm editor, Plug in for Rhinoceros, 2015 <http://www.grasshopper3d.com>.
- [24] V. Moosavi, S. Packmann, I. Vallés, Sompy: a python library for self organizing map (som), [Online]. Available GitHub, <https://github.com/sevamoo/SOMPY>, (2014).
- [25] Vahid Moosavi, Computing With Contextual Numbers (arXiv preprint arXiv:1408.0889), (2014).
- [26] Man-Woo Park, Ioannis Brilakis, Construction worker detection in video frames for initializing vision trackers, Autom. Constr. 28 (2012) 15–25, <https://doi.org/10.1016/j.autcon.2012.06.001> (ISSN 0926-5805), <http://www.sciencedirect.com/science/article/pii/S0926580512001136>.
- [27] Polyhedral Structures Laboratory, Polyframe, <https://www.food4rhino.com/app/polyframe>, (2018).
- [28] M. Rankine, Principle of the equilibrium of polyhedral frames, Philos. Mag. 27 (180) (1864) 92.
- [29] Gabriella Rossi, Paul Nicholas, Haptic learning: towards neural-network-based adaptive cobot path-planning for unstructured spaces, Proceedings of ECAADE SIGRADI 2019, 2019, pp. 201–210.
- [30] M.J. Schrems, T. Kotnik, Statically motivated form-finding based on extended graphical statics (eggs), in: R. Stouffs, P. Janssen, S. Roudavski, B. Tuner (Eds.), Open Systems: Proceedings of the 18th International Conference on Computer-aided Architectural Design Research in Asia (CAADRIA 2013), 2013, pp. 843–852.
- [31] C. Theodoropoulos, Shaping Structures: Statics, MIT Press Five Cambridge Center, Cambridge, MA 02142 USA, 2000.
- [32] Antoine J.-P. Tixier, Matthew R. Hallowell, Balaji Rajagopalan, Dean Bowman, Application of machine learning to construction injury prediction, Autom. Constr. 69 (2016) 102–114.
- [33] Matt Turlock, Kyle Steinfeld, Necessary tension: a dual-evaluation generative design method for tension net structures, Proceedings of the Design Modelling Symposium 2019, 2019, pp. 250–262.
- [34] T. Van Mele, P. Block, Algebraic graph statics, Comput. Aided Des. 53 (2014) 104–116, <https://doi.org/10.1016/j.cad.2014.04.004>.
- [35] T. Van Mele, M. Rippmann, L. Lachauer, P. Block, Geometry-based understanding of structures, Journal of the International Association of Shell and Spatial Structures 53 (2) (2012).
- [36] W.S. Wolfe, Graphical Analysis: A Text Book on Graphic Statics, McGraw-Hill Book Company, Inc., 1921.
- [37] Ngai Hang Wu, Marina Dimopoulos, Han Hsun Hsieh, Christos Chatzakis, Rawbot: a digital system for fabrication of bamboo structures through the discrete digitization of bamboo, Proceedings of ECAADE SIGRADI 2019, 2019, pp. 161–170.
- [38] Ozan Yetkin, Arzu Gönenç Sorguç, Design space exploration of initial structural design alternatives via artificial neural networks, Proceedings of ECAADE SIGRADI 2019, 2019, pp. 55–60.
- [39] Hao Zheng, Form finding and evaluating through machine learning: the prediction of personal design preference in polyhedral structures, The International Conference on Computational Design and Robotic Fabrication, Springer, 2019, pp. 169–178.