Proceedings of the IASS 2024 Symposium
Redefining the Art of Structural Design
August 26-30, 2024, Zurich, Switzerland
Philippe Block, Giulia Boller, Catherine DeWolf,
Jacqueline Pauli, Walter Kaufmann (eds.)

Constrained shape optimization of grid shells based on deep learning

Andrea FAVILLI[a,b], Francesco LACCONE[*], Paolo CIGNONI[a], Luigi MALOMO[a], Daniela GIORGI[a]

[*] Institute of Information Science and Technologies (ISTI), National Research Council of Italy (CNR)
via G. Moruzzi 1, Pisa
francesco.laccone@isti.cnr.it

[a] Institute of Information Science and Technologies (ISTI), National Research Council of Italy (CNR)
[b] University of Pisa

Abstract

Designing grid shells requires finding a happy medium between aesthetics and engineering quality: architects and structural engineers join efforts to define geometries and grid topologies that achieve structural efficiency. In sculptural architecture, the artistic intent prevails, and produces freeform shapes with possibly large openings to create spectacular effects. This calls for shape optimization methods to mitigate inefficiency caused by bending forces. However, if modifications are not bounded, optimization may either alter the surface aesthetics or violate design constraints. This paper implements a shape optimization method that improves the performance of triangular grid shells while ensuring small shape changes. A graph neural network learns to update the nodal coordinates of the grid shell and reduce both strain-energy, as a measure of structural efficiency, and the total weight of the structure, as a measure of sustainability. Our case studies include regular shapes among the baseline structures of the FreeGrid benchmark, as well as non-conventional geometries.

Keywords: FreeGrid benchmark, conceptual design, form finding, shape optimization, deep learning, gridshell, sustainability, steel structures, automatic differentiation

1. Introduction

Shells and grid shells are subject to unique contextual, functional, constructional, and aesthetic constraints, with the correlation between geometric shape and structural parameters varying on a case-by-case basis. In architectural practice, free-form configurations are customary, as they open up to many different creative design possibilities. However, free-form grid shells often exhibit complex structural behavior, whereas the desiderata would be predominant membrane characteristics, low bending forces, and robustness [1]. In the ideal case, the optimal form emerges naturally by applying loading and boundary conditions in form-finding methodologies [2, 3], whereas in the general case, the shape is not naturally efficient and requires several design checks and improvements based on heuristics, experience insights, or automated optimization methods. Objective functions typically include structural parameters, which are iteratively adjusted by editing design variables within predefined bounds [4].

If the target shape is fixed, the structural response of the final structure can be improved via grid design [5], topology optimization [6], cross section refinement [7, 8], and reinforcement [9, 10]. Conversely, *shape optimization* offers a set of methods and tools for maximizing the performance of a structure modifying the input shape.

The literature offers a large variety of shape optimization methods, differing in the types of variables, objective functions, optimization methods, etc. In [11], the variables are finite element mesh nodes, which are updated through deformation gradient. The solution is improved through surface regularization and distortion control. The authors of [12] employ a NURBS surface formulation, using control points and thickness as variables. The shape modifications result from finite element analysis and gradient evaluations, performed using Automatic Differentiation (AD). In [13], structural stiffness is maximized in a grid shell by iteratively updating nodal coordinates according to sensitivity information. Since the procedure is initialized with a displacement perturbation, which can lead to a jagged surface, a filtering scheme is adopted to normalize the non-smooth gradient fields. Other shape optimization methods are formulated as multi-criteria optimization. In [14], curves and surfaces can have discontinuities in tangent vectors and curvatures (creases). A multi-objective optimization problem is solved by the constraint approach to generate a trade-off design between smoothness and mechanical compliance.

In this paper, we propose an alternative implementation of the method in [15], which uses geometric deep learning to modify a grid shell shape. In the original work, a graph neural network is fed with geometric features of the grid shell and performs structural analysis at each learning step; the variables are the node coordinates, and the output is a node displacement field that improves the input shape. In this paper, the graph-neural-network model in [15] is expanded to solve a multi-target task. The original model is driven only by the static analysis of grid shells to solve both form finding and shape optimization problems. In the presented method we introduce an additional term in the objective function (i.e., the loss) that encourages the reduction of global beam length. This augmentation aligns with sustainable principles of saving material, according to the FreeGrid benchmark.

The test dataset is composed of two regular shapes from the FreeGrid benchmark [16, 17], a parabolic dome and a hyperbolic paraboloid with different meshing, and three non conventional free-form shapes. In Section 2. we introduce the formulation of the updated problem and the learning model. The results are presented in Section 3. and discussed in Section 4..

2. Geometric Deep Learning for Shape Optimization

The input grid shell is represented as triangular mesh $\mathcal{M} = (\mathcal{V}, \mathcal{E}, \mathcal{F})$, in which $\mathcal{V}$ are the vertices, identified with the structural nodes; $\mathcal{E}$ are the edges, identified with the beams; and $\mathcal{F}$ are the mesh faces. In the following, the term displacement will denote the shape optimization outcome, while deformation is intended as the byproduct of loads.

The learning-based shape optimization acts on the vertices by imposing translations, i.e., displacement, while keeping the same number of beams and node connectivity. In other words, the neural network model predicts an optimal translation vector $\delta_{\mathbf{v}} \in \mathbb{R}^3$ for each vertex $\mathbf{v} \in \mathcal{V}$, so that the original mesh topology is retained. The degrees of freedom that are fixed due to external constraints are not affected. Thus, null displacements and rotations are applied on a given subset of constrained vertices $\overline{\mathcal{V}} \subseteq \mathcal{V}$.

The neural network model takes as input an augmented encoding of the original structure. Different pieces of information are combined into an *input feature vector* $\mathbf{x}_{\mathbf{v}} \in \mathbb{R}^{12}$, for each vertex $\mathbf{v} \in \mathcal{V}$. The vector includes as *features*: vertex coordinates, vertex normals, principal curvatures of the underlying freeform surface, and four distance and centrality measures of vertices with respect to the shape boundary $\partial\mathcal{V}$. Then, three modules arranged in a deep sequence of layers produce intermediate feature transformations until the final layer yields the prediction $\delta_{\mathbf{v}}$ (Figure 1). A critical network module is made of *Graph Attention Layers*, which work on nearest-neighbor graphs in feature spaces: at each layer, the vertex features are updated as weighted averages of neighboring features in the graph [15].

The network weights are determined by minimizing the *loss*, which is the target objective function based on the strain energy and on the total amount of steel used. The loss minimization follows a gradient

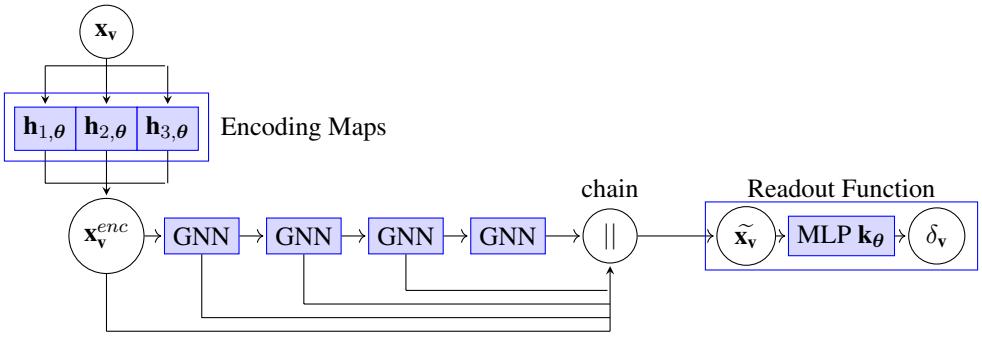


Figure 1: The adoped learning model is made of three modules: *Encoding Maps*, *Graph Attention Layers*, and a *Readout Function*. The feature vector $\mathbf{x_v}$ is encoded in $\mathbf{x}_\mathbf{v}^{enc}$: three Encoding Maps expand each semantic cluster of the feature vector (coordinates, curvatures, and geodesic measures) to 256 components in the range $[-1, 1]$. $\mathbf{x}_\mathbf{v}^{enc}$ is subsequently fed to four Graph Attention layers (GNN) to get a deep vector $\widetilde{\mathbf{x}_\mathbf{v}}$ after layer output concatenation. Finally, a multilayer perceptron plays the role of the Readout Map to produce vertex displacements $\delta_\mathbf{v}$ from the deep vectors.

descent method: starting from randomly initialized network weights, the weights are moved along the direction of maximum loss decrease (i.e., the opposite of the gradient) in an iterative procedure. At each step, the decrease direction vector is scaled by the *learning rate* parameter.

The loss to be minimized is the sum of three terms:

$$\mathcal{L}(\mathcal{M}) = \mathcal{L}^* + \zeta\mathcal{B} + \eta\mathcal{S} \tag{1}$$

The main loss component concerns the structural performance and exploits a simplified linear static analysis of the grid shell adopting a two-node Euler-Bernoulli beam formulation with linearly-interpolated endpoints force values only. This formulation enables simplified and rapid computations with reasonable accuracy for a grid shell that is uniquely loaded at the vertices. The loss function $\mathcal{L}^*$ is defined as the mean strain energy over all beams:

$$\mathcal{L}^* = \frac{1}{|\mathcal{E}|}\sum_{e\in\mathcal{E}} E_e \tag{2}$$

where E_e is the strain energy of the beam e. To avoid distortions on the boundary fairness, the boundary constraints are solved in a soft weighted fashion imposing:

$$\mathcal{B} = \frac{1}{|\overline{\partial V}|}\sum_{\mathbf{v}\in\overline{V}} ||\delta_\mathbf{v}|| \tag{3}$$

where $\mathcal{B}$ is a penalty term ensuring that the displacement for the fixed degrees of freedom remain small during the training, up to the final iteration, in which they are forced to zero. The weight ζ sets the penalty at a fixed percentage of the structural loss $\mathcal{L}_0^*$ at the first step 0, which is set at 30% in the current implementation.

The third term of the loss $\mathcal{S}$ is inspired by the FreeGrid benchmark [16, 17], and accounts for the sustainability. In FreeGrid, the goal is to obtain more sustainable solutions by reducing the steel weight. In particular, the loss term considers the length of the beams, the steel grade and cross section type.

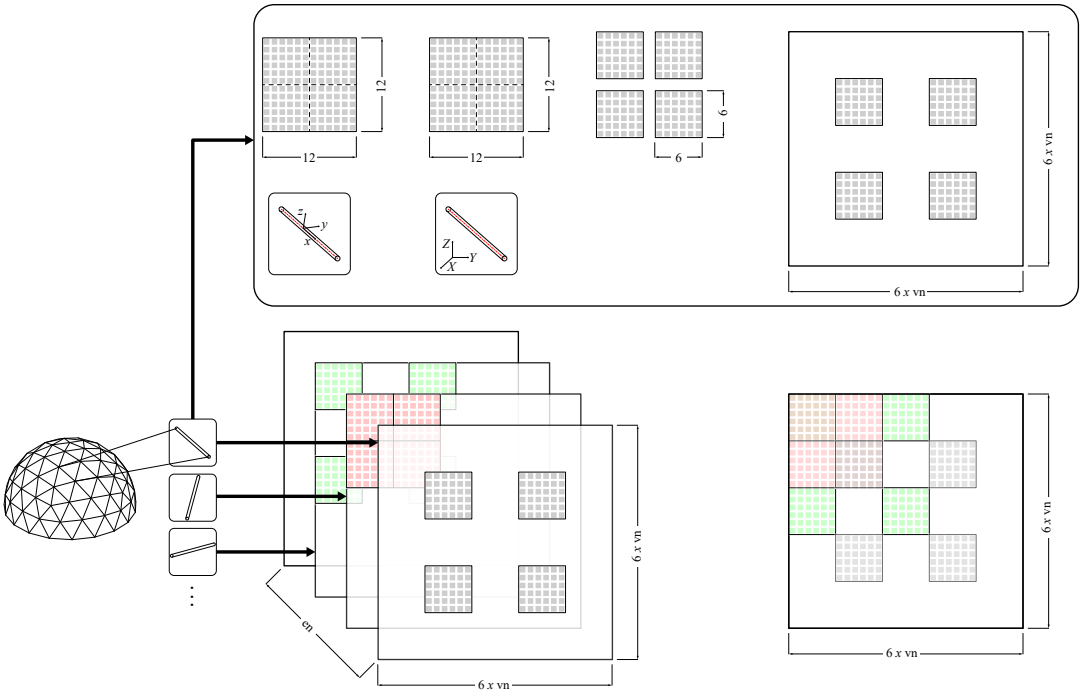


Figure 2: Assembly of the stiffness matrix starting from a 3D array. For each beam, the local stiffness matrix is transformed in the global reference system, then the four 6x6 blocks, corresponding to the 6 degrees of freedom at the 2 beam endpoints, are arranged in a $6|V|$x$6|V|$ matrix according to the endpoints' indices. Lastly, the global stiffness matrix is assembled as the sum of all beam matrices along the beam dimension.

Neglecting the terms that are constant during learning, the sustainability term reads as:

$$\mathcal{S} = \sum_{e \in \mathcal{E}} l_e = L_{tot} \tag{4}$$

where l_e is the length of a single beam e. This term is weighted by a factor η, namely a fixed percentage of the structural loss $\mathcal{L}_0^*$ at step 0. We set η to 100% in the current implementation, since the structural and sustainability partial metrics in FreeGrid equally contribute in the bulk performance metric.

One of the main novelties of the approach is the introduction of a differentiable Euler-Bernoulli linear static approach that integrates with learning models. The loss function in Eq. (1) can be made differentiable with respect to vertex coordinates, so that it can be used to determine the neural network weights through gradient descent. The method relies on Automatic Differentiation (AD) to get punctual evaluations of gradients, thus avoiding the need to express analytically the partial derivatives of $\mathcal{L}$. For computational efficiency on GPUs, the assembly of the stiffness matrix is vectorized, as shown in Fig. 2. Sparse matrices, which gather the individual beam stiffness matrices, are assembled node-wise in parallel. Subsequently, the non-zero values are summed all at once.

This shape optimization task is framed as *single-instance learning*. For each input mesh, the network weights are reset and a new loss minimization procedure is started. Apart from the mesh, the input parameters are the beam characteristics, the external load, and the learning rate. Formally, the task consists of finding an optimal mesh $\mathcal{M}^* = T_{\boldsymbol{\theta}^*}(\mathcal{M})$ such that:

$$\boldsymbol{\theta}^* \in \text{argmin}_{\boldsymbol{\theta}}\, \mathcal{L}(T_{\boldsymbol{\theta}}(\mathcal{M})) \tag{5}$$

where the neural model $T_{\boldsymbol{\theta}}$ is a function of the weights $\boldsymbol{\theta}$.

A scale-dependent stopping criterion is adopted for the iterative procedure, which is based on both absolute and relative loss variation. The relative component is satisfied when the difference between the maximum and the minimum loss is $0.1 \cdot \mathcal{L}_0$, where $\mathcal{L}_0$ is the initial loss. The absolute component is satisfied when the loss variation in the last 50 iterations is less than 0.005. An alternative version has been proposed in [18] implementing a graphical user interface that allows the user to enter the execution flow. The user can change the optimization parameters on the fly and visualize the shape modification from different perspectives. Moreover, the user can replay or restart the procedure until he/she founds a balance between performance metrics and amount of shape alteration.

3. Case studies and optimization setup

We evaluate the performance of the proposed method against the benchmark FreeGrid, which is meant to test and compare different approaches to the design and optimization of steel gridshells [16, 17]. The benchmark aims at improving three baseline design problems: a barrel vault, a parabolic dome, and a hyperbolic paraboloid, having a free-edge and subjected to uniform and piecewise load conditions. The improvement is measured in relative terms according to a holistic bulk metric, which considers structural behavior, buildability, and sustainability. Specifically, the structural performance metric accounts for both ultimate and serviceability behavior, by calculating the critical Load Factor and maximum vertical displacement; the buildability performance metric evaluates face planarity, uniformity of structural joints and members; the sustainability performance metric accounts for the carbon the structure embodies. The benchmark admits all techniques for shape, topology and size modifications, as long as the design domain fits some given constraints related to geometry, structure and material.

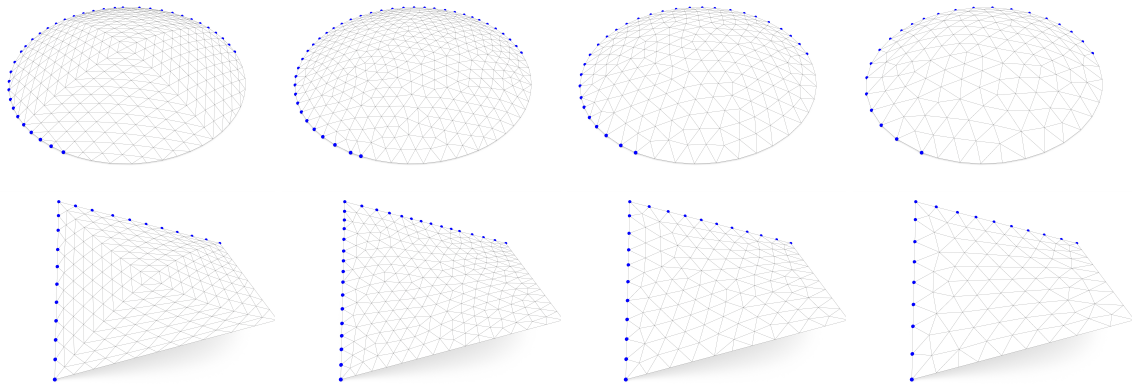
Fig. 3 and Fig. 4 show our five case studies. The first two cases are the FreeGrid parabolic dome and hyperbolic paraboloid, after suitable triangular remeshing (Fig. 3). The free edge causes structural inefficiency, therefore a shape optimization step is required. We analyse four different meshes for each case study, obtained by bracing the quads of the starting baseline geometries, and performing isotropic remeshing with different target average lengths. Tab. 1 reports the statistics for each mesh configuration, with letter b labelling bracing and i-x isotropic remeshing with target length x, respectively. Consider that grid shells with uniform cross section and same total beam (or edge) length are deemed comparable (i.e., the $b-$ and $i-1.5$).

In all cases the beams are made of common steel S355 with Young's modulus E = 2.1e+ 5 MPa, Poisson's ratio $\nu = 0.3$, yield strength $f_y = 355MPa$, and density $\rho = 7850kg/m^3$. The cross section are tubes of 101.6mm diameter and 10mm thickness as in [17]. The present method for shape optimization performs under SLS uniform loading of $1.2\ kN/m^2$ and for the same boundary conditions adopted in the benchmark, i.e. with hinged nodes on the boundary apart from the free edge.

Additionally, we test our method on other three unconventional geometries, shown in Fig. 4. The aim is to showcase shape optimization in presence of more complex shape features, boundary status and irregular structural behavior. The mesh statistics are reported in the last three rows of Tab. 1.

4. Results and discussion

The results are presented in Figs. 5-7, where two columns are reported for each case study: on the left, the input shape mesh; on the right, the optimized model. Color maps represent the beam strain energy, the deformations due to load (u), and the displacements (δ, i.e., increments for node coordinates). The last column of Tab. 1 includes the sustainability performance after the optimization (Eq. 4). Remarkably, in the present method the learning rate is the only parameter controlling the shape change at each iteration.



(b) (i-1.5) (i-2.0) (i-2.5)

Figure 3: Case studies from the FreeGrid benchmark: (top) parabolic dome; (bottom) hyperbolic paraboloid. Different meshings: (b) braced quad; (i-1.5) isotropic with target length 1.5m; (i-2.0) isotropic, 2m; (i-2.5) isotropic, 2.5 m. Blue spheres indicate hinge supports.

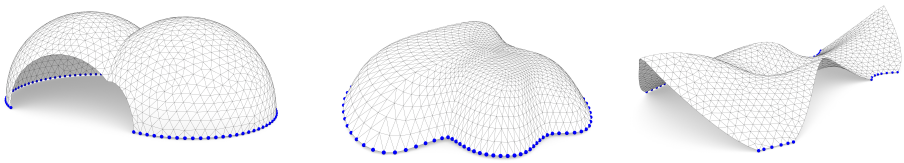


Figure 4: Case studies from non conventional shapes. From left: 2Spheres, Botanic, Wave. Blue spheres indicate hinge supports.

The learning rate controls convergence speed and closeness of the output geometry to the original shape. We set the learning rate to 0.1 for the parabolic dome and to 0.01 for the hyperbolic paraboloid, and keep it constant for the different meshing configurations, which have highly-variable stiffness, number of nodes, and energy distribution.

For the parabolic dome (Fig. 5), deformation and strain energy localization are qualitatively uniform across the different cases. Notably, the mesh (b) exhibits higher efficiency owing to its more rationale beam allocation. The use of geometric features as input to the learning model ensures smoothness preservation in the displaced shapes. Furthermore, all optimized solutions are characterized by the elevation of the free edge, namely its vertices have a non-zero vertical component. As the number of beams decreases, the projected area of the grid shell contracts, resulting in a reduction of the cantilevered portion. The stiffness augmentation, associated with a higher number of beams (given the cross section constant), enhances the dome's capacity to withstand openings, and reduces the need for significant modifications to its shape. Conversely, in case (i-2.5), the impact of shape modifications on the cantilever part is more pronounced. It is noteworthy the unique pattern observed in the case (b), characterized by radial lines where mesh diagonal orientations switch, denoting discontinuities. These features are intentionally preserved by the learning model, and become sharp lines. With more iterations and higher learning rate, this grid shell would fold along these lines. In all cases, the total length of beams increases due to the raised elevation of nodes (last row in Fig. 5 and last column of Tab. 1). This shape modification is sustained by the strain energy minimization (loss component $\mathcal{L}^*$, Eq. 2) which hampers the sustainability (loss component $\mathcal{S}$, Eq. 4).

Table 1: Mesh statistics of the various case studies employed: Name, Size (mxmxm), number of vertices $|\mathcal{V}|$ (i.e., structural nodes), faces $|\mathcal{F}|$ (triangular panels), edges $|\mathcal{E}|$ (beams), average edge length avg(l_e) in meters, total length of the edges $L_{tot} = \sum l_e$ in meters at the starting condition and $L_{tot,opt}$ after the optimization (see Sec. 4.).

Name	Size	Mesh type	$\|\mathcal{V}\|$	$\|\mathcal{F}\|$	$\|\mathcal{E}\|$	avg(l_e)	L_{tot}	$L_{tot,opt}$
parabolic dome	30.00x30.00x3.75	(b)	349	640	988	1.73	1711.86	1726.22
		(i-1.5)	399	737	1135	1.54	1744.88	1769.36
		(i-2.0)	232	418	649	2.04	1324.27	1346.90
		(i-2.5)	156	274	429	2.52	1082.09	1089.53
hyperbolic paraboloid	30.00x30.00x7.50	(b)	221	400	620	1.78	1105.19	1102.17
		(i-1.5)	261	463	723	1.54	1113.57	1109.80
		(i-2.0)	146	250	395	2.10	829.87	825.22
		(i-2.5)	102	168	269	2.56	688.62	663.43
2Spheres	34.23x30.56x11.38	-	1373	2617	3989	1.04	4135.99	4148.81
Botanic	30.80x28.57x7.69	-	1121	2152	3272	0.97	3181.73	3259.04
Wave	28.26x49.03x9.91	-	1130	2110	3239	1.03	3325.53	3311.40

For the hyperbolic paraboloid (Fig. 6), unlike the previous scenario, the design space is constrained since its anticlastic shape is more efficient. To enhance the stiffness of the cantilevered section along the free edge and mitigate strain energy at the transition from fixed to free edge, the most viable approach involves reducing the span of the cantilever. This approach is also driven by the sustainability term in the loss (see the length reduction in the last column of Tab. 1). Considering also the lower number of beams and nodes with respect to the dome cases, a smaller learning rate is adopted to avoid the shape folding on the fixed part. Since the learning model and the static solver include and update the dead load of the beams, comparing different configurations is not easy. Only the meshes (b) and (i-1.5) are genuinely comparable for similar starting conditions, and hence the resulting stiffness and the magnitude of the displacement is similar. Conversely, case (b) exhibits a folding tendency akin to the dome case, prompting a rise in shape, particularly at the center. In isotropic cases (i), displacements occur more smoothly and locally. The case (i-2.5) starts from a higher strain energy, so the learning rate proportionally makes shape changes more intense, and displacements larger.

Concerning free-form shapes (Fig. 7), different learning rates are adopted: 0.8 for the shapes *2Spheres* and *Botanic*, 0.5 for *Wave*. The 2Spheres model results as a fusion of two hemispheres intersecting at a crease line with an opening, causing stress concentration and deformable parts, respectively. The displacement mobilize smoothly all nodes, aiming to maintain uniform curvature while transitioning towards a funicular configuration. The Botanic model is a nearly funicular design. The displacements tend to increase the curvature in the flatter areas and more uniformly distribute the energy among the beams. The Wave model has a shape developing membrane behavior in the central part, with cantilevering boundaries and large openings, where maximal deformations occur. Despite experiencing significant displacement, reaching up to 3.78 m, the overall shape remains relatively unaltered, with the smoothness and distinctive geometric features, such as peaks, valleys, and the shape of openings, preserved. Moreover, this latter model shows an improvement in terms of sustainability (last column of Tab. 1).

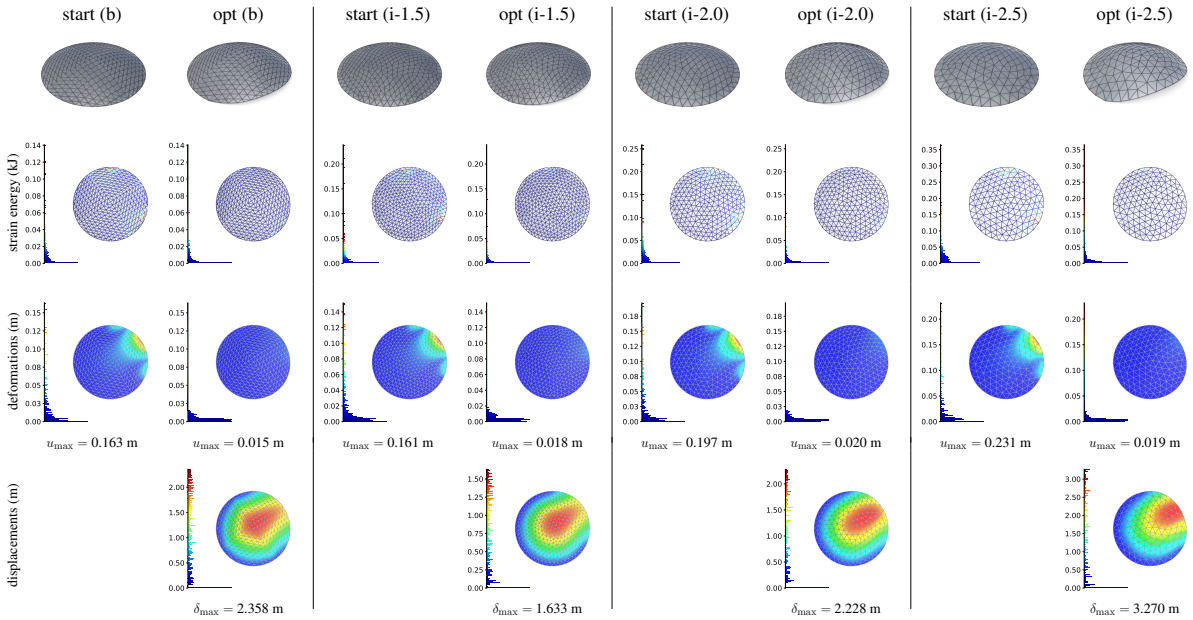


Figure 5: Results on the parabolic dome.

5. Conclusions and future perspectives

This contribution implements a geometric deep learning-based approach to shape optimization of grid shells that targets the minimization of a loss summing the strain energy of the structure and the sustainability impact of the steel beams used. The proposed method is deployed on common shapes characterized by a free edge, as well as on freeform shapes. Results indicate good performance in achieving multi-target objectives, effectively balancing the requirements for enhanced static efficiency and reduced steel consumption. Notably, the method is able to preserve relevant input geometric attributes, such as meshing and curvature, which is advantageous in freeform applications. The inherent flexibility of the core methodology suggests several future research lines, including expanding the loss formulation to encompass additional metrics, such as those related to buildability, in addition to statics and sustainability.

Acknowledgments

This work was supported by the NextGenerationEU programme under the funding schemes PNRR-PE-AI scheme (M4C2, investment 1.3, line on AI) FAIR "Future Artificial Intelligence Research", grant id PE00000013. Data and source code are available on GitHub at `https://github.com/cnr-isti-vclab/GeomDL4GridShell`

References

[1] E. Ramm, "Shape finding of concrete shell roofs," *Journal of the International Association for Shell and Spatial Structures*, vol. 45, no. 1, pp. 29–39, 2004, ISSN: 1028-365X.

[2] D. Veenendaal and P. Block, "An overview and comparison of structural form finding methods for general networks," *International Journal of Solids and Structures*, vol. 49, no. 26, pp. 3741–3753, 2012. DOI: 10.1016/j.ijsolstr.2012.08.008.

[3] S. Gabriele, V. Varano, G. Tomasello, and D. Alfonsi, "R-Funicularity of form found shell structures," *Engineering Structures*, vol. 157, pp. 157–169, 2018, ISSN: 0141-0296. DOI: 10.1016/j.engstruct.2017.12.014.

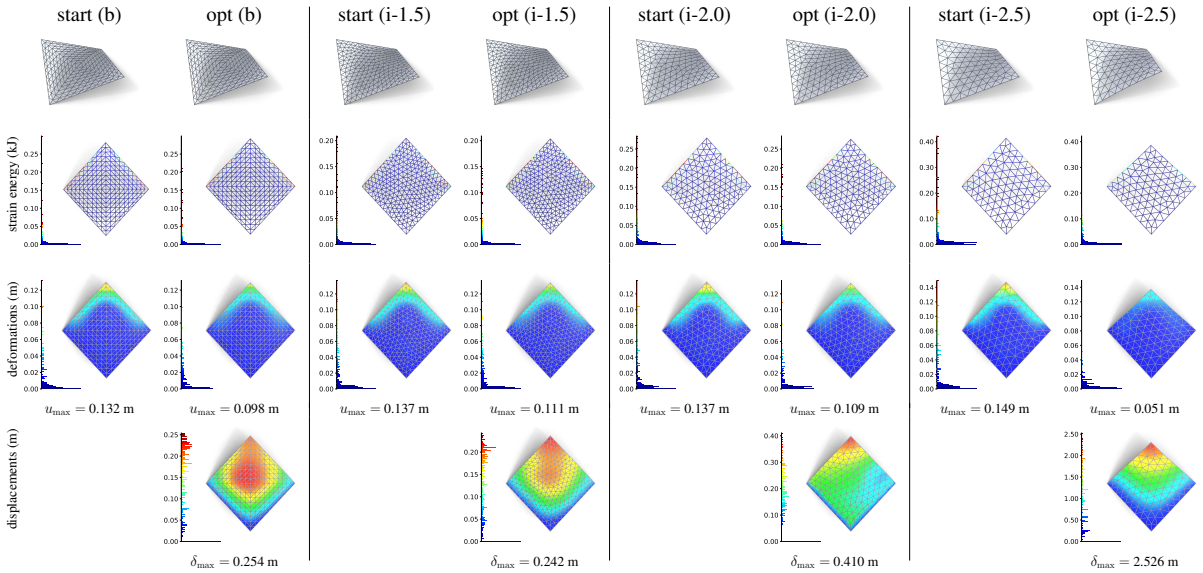


Figure 6: Results on the hyperbolic paraboloid.

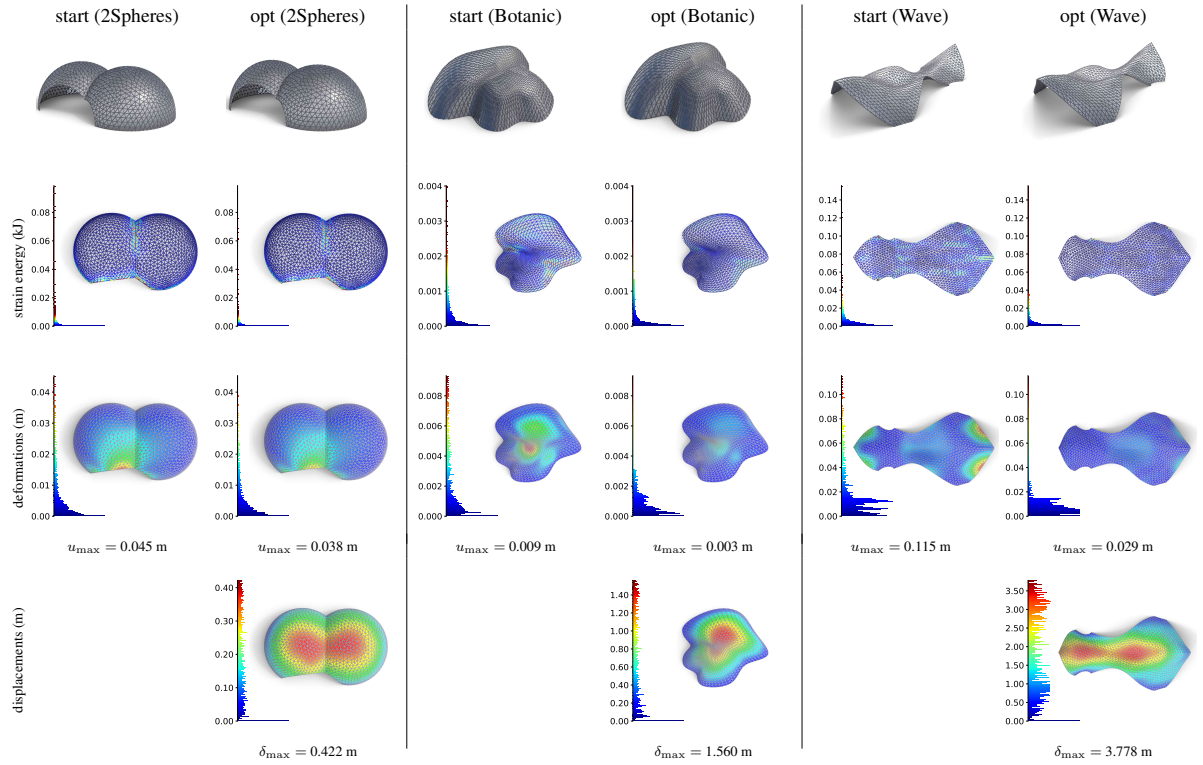


Figure 7: Results on non conventional shapes: 2Spheres, Botanic, and Wave.

[4] S. Adriaenssens, P. Block, D. Veenendaal, and C. Williams, *Shell structures for architecture: form finding and optimization*. Routledge, 2014.

[5] N. Pietroni, D. Tonelli, E. Puppo, M. Froli, R. Scopigno, and P. Cignoni, "Statics Aware Grid Shells," *Comput. Graph. Forum*, vol. 34, no. 2, pp. 627–641, May 2015, ISSN: 0167-7055. DOI: 10.1111/cgf.12590.

[6] J. Ma, H. Lu, T.-U. Lee, Y. Liu, D. W. Bao, and Y. M. Xie, "Topology optimization of shell structures in architectural design," *Architectural Intelligence*, vol. 2, no. 1, p. 22, 2023. DOI: 10.1007/s44223-023-00042-z.

[7] X. Meng, Y. Xiong, Y. M. Xie, Y. Sun, and Z.-L. Zhao, "Shape–thickness–topology coupled optimization of free-form shells," *Automation in Construction*, vol. 142, p. 104 476, 2022, ISSN: 0926-5805. DOI: 10.1016/j.autcon.2022.104476.

[8] E. Grande, M. Imbimbo, and V. Tomei, "Structural optimization of grid shells: Design parameters and combined strategies," *Journal of Architectural Engineering*, vol. 24, no. 1, p. 04 017 027, 2018. DOI: 10.1061/(ASCE)AE.1943-5568.0000286.

[9] F. Laccone, L. Malomo, M. Froli, P. Cignoni, and N. Pietroni, "Automatic Design of Cable-Tensioned Glass Shells," *Computer Graphics Forum*, vol. 39, no. 1, pp. 260–273, 2020. DOI: 10.1111/cgf.13801.

[10] F. Laccone, N. Pietroni, P. Cignoni, and L. Malomo, "Bending-reinforced grid shells for free-form architectural surfaces," *Computer-Aided Design*, vol. 168, p. 103 670, 2024, ISSN: 0010-4485. DOI: 10.1016/j.cad.2023.103670.

[11] K.-U. Bletzinger, M. Firl, J. Linhard, and R. Wüchner, "Optimal shapes of mechanically motivated surfaces," *Computer methods in applied mechanics and engineering*, vol. 199, no. 5-8, pp. 324–333, 2010. DOI: 10.1016/j.cma.2008.09.009.

[12] L. Espath, R. V. Linn, and A. Awruch, "Shape optimization of shell structures based on nurbs description using automatic differentiation," *International Journal for Numerical Methods in Engineering*, vol. 88, no. 7, pp. 613–636, 2011. DOI: 10.1002/nme.3183.

[13] H. Wang, Z. Chen, G. Wen, G. Ji, and Y. Min Xie, "A robust node-shifting method for shape optimization of irregular gridshell structures," *Structures*, vol. 34, pp. 666–677, 2021, ISSN: 2352-0124. DOI: 10.1016/j.istruc.2021.08.003.

[14] M. Ohsaki, T. Ogawa, and R. Tateishi, "Shape optimization of curves and surfaces considering fairness metrics and elastic stiffness," *Structural and Multidisciplinary Optimization*, vol. 27, pp. 250–258, 2004. DOI: 10.1007/s00158-004-0382-3.

[15] A. Favilli, F. Laccone, L. Malomo, P. Cignoni, and D. Giorgi, "Geometric deep learning for statics-aware 3d gridshells," *Computers and Structures*, vol. 34, no. 107238, 292. DOI: 10.1016/j.compstruc.2023.107238.

[16] L. Bruno *et al.*, "Freegrid: A benchmark on design and optimisation of free-edge gridshells," in *Proceedings of the IASS Annual Symposium 2023, Melbourne*, 2023.

[17] L. Bruno *et al.*, "Exploring new frontiers in gridshell design: The freegrid benchmark," *Structures*, vol. 58, p. 105 678, 2023, ISSN: 2352-0124. DOI: 10.1016/j.istruc.2023.105678.

[18] A. Favilli, F. Laccone, P. Cignoni, L. Malomo, and D. Giorgi, "A geometry-preserving shape optimization tool based on deep learning," in *Shell and Spatial Structures*, S. Gabriele, A. Manuello Bertetto, F. Marmo, and A. Micheletti, Eds., Cham: Springer Nature Switzerland, 2024, pp. 549–558, ISBN: 978-3-031-44328-2. DOI: 10.1007/978-3-031-44328-2_57.