DeepForm: A Graph Neural Network Surrogate for Real-Time Tensile Membrane Form-Finding Across Anticlastic Typologies

Sandesh Lamsal
Department of Civil, Architectural and Environmental Engineering
University of Miami, Coral Gables, FL 33146, USA
sandeshlamsal@miami.edu

Abstract

Form-finding of tensile membrane structures, the iterative process of determining the equilibrium shape under prescribed prestress and boundary conditions, is computationally expensive when performed with the Dynamic Relaxation (DR) method, the current industry standard. We propose DeepForm, a Graph Neural Network (GNN) surrogate trained once on a dataset of 10,987 DR solver outputs spanning six geometrically distinct anticlastic membrane typologies (hypar saddle, Scherk minimal surface, Enneper minimal surface, monkey saddle, sinc-difference saddle, and Kresge triangular shell) that directly predicts equilibrium node positions from boundary configurations in milliseconds. The membrane is represented as a graph in which nodes encode geometric and structural connectivity features and edges encode cable or membrane element properties, allowing the message-passing architecture to mimic force propagation through the structure. Per-type displacement normalisation is introduced to handle the large variation in deformation magnitude across typologies (spanning approximately a factor of 6 in deformation scale) while sharing a single network. Unlike physics-informed neural network (PINN) approaches that require retraining per geometry, DeepForm generalises across new boundary configurations within each typology without retraining, enabling millisecond GNN-based form-finding feedback suitable for interactive structural design workflows.

Keywords: form-finding; tensile membrane structures; graph neural network; surrogate model; dynamic relaxation; structural engineering

Notation

Symbol	Meaning	Symbol	Meaning
$\mathcal{G}$	Membrane graph	$\mathbf{h}_i^{(l)}$	Node embedding, layer l
$\mathcal{V}$, $\mathcal{E}$	Node / edge sets	$\boldsymbol{e}_{ij}^{(l)}$	Edge embedding, layer l
$\mathcal{F}$, $\mathcal{B}$	Free / boundary nodes	$\boldsymbol{m}_i^{(l)}$	Aggregated message
$\mathbf{x}_i$	Node feature vector ($\mathbb{R}^8$)	φ_e, φ_n	Edge / node update MLPs
$\mathbf{e}_{ij}$	Edge feature vector ($\mathbb{R}^7$)	w_{ij}	Cotangent aggregation weight
$\mathbf{u}$	Global feature vector ($\mathbb{R}^7$)	α_{ij}, β_{ij}	Opposite angles in adj. triangles
b_i	Boundary indicator	L	Number of MP layers ($= 8$)
ν_i	Node valence	d	Hidden dimension ($= 256$)
$\bar{\sigma}_i$	Mean incident prestress	$\mathbf{p}_i^{\text{init}}$	Initial node position
d_i^{fix}	Distance to nearest fixed node	$\mathbf{p}_i^{\text{eq}}$	Equilibrium node position
$\bar{\ell}_i$	Mean incident edge length	$\Delta\mathbf{p}_k$	Nodal displacement
t_{ij}	Edge type (membrane / cable)	μ_τ, σ_τ	Per-type displacement mean/std
σ_{ij}	Edge prestress	$\mathbf{z}_k$, $\hat{\mathbf{z}}_k$	Normalised target / prediction
ℓ_{ij}, ℓ_{ij}^0	Edge length / rest length	σ_0	Isotropic prestress (kN/m)
$\mathcal{N}(k)$	Neighbourhood of node k	$\mathcal{L}$	Training loss (normalised MSE)

1 Introduction

Tensile membrane structures (TMS) are widely used in contemporary architecture for their ability to span large areas with minimal material, achieving structural efficiency through tension-only loading. Applications include stadium roofs, airport canopies, event pavilions, and permanent shade structures. Unlike conventional framed structures, the equilibrium geometry of a TMS is not prescribed by the designer but emerges from the interaction between boundary conditions, prestress distribution, and material properties, a process known as *form-finding* [1, 2].

Form-finding is an inherently nonlinear geometric problem: the stiffness of the structure depends on its deformed shape, which is itself the unknown. The membrane must simultaneously satisfy force equilibrium at every node while taking a shape geometrically consistent with its boundary constraints. The Dynamic Relaxation (DR) method [1, 2] is the most widely used computational approach. DR solves the equilibrium problem by simulating a fictitious dynamic process, assigning artificial masses to structural nodes, and allowing the system to oscillate until kinetic energy is dissipated and static equilibrium is approached. On

production-density meshes, a typical tensile membrane analysis requires many thousands of DR iterations to converge, taking several seconds to minutes per analysis. For design workflows requiring hundreds or thousands of evaluations (parametric studies, optimisation, or real-time interactive tools), the DR solver becomes a critical bottleneck.

Recent work by Kabasi et al. [3–5] has demonstrated that Physics-Informed Neural Networks (PINNs) can solve the form-finding problem by learning to satisfy the governing PDE of minimal surfaces, avoiding the convergence issues associated with DR. However, the PINN framework requires training a separate network for each new boundary configuration, which may take minutes per geometry. This per-instance training cost fundamentally limits practical utility for interactive design exploration involving many different geometries.

The present work proposes a fundamentally different paradigm: a trained surrogate model that generalises across boundary configurations by learning from a dataset of DR solver outputs. Once trained, DeepForm predicts the equilibrium shape of any new configuration with a single forward pass in milliseconds, without retraining.

Graph Neural Networks [6–8] are particularly well-suited to this problem. The structural connectivity of a membrane, comprising nodes, cable elements, and triangular fabric panels, maps naturally onto a graph, and the message-passing mechanism of GNNs propagates information along structural edges in a manner that conceptually parallels force transmission through the membrane. This structural inductive bias is expected to improve generalisation compared to architectures treating the problem as unstructured regression. The architecture follows the MeshGraphNets framework of Pfaff et al. [9], extended to tensile membrane form-finding with cotangent-weighted aggregation and per-typology displacement normalisation.

Gladstone et al. [10] identified limitations of standard MeshGraphNets architectures on time-independent static problems related to long-range information propagation, and proposed edge-augmented variants to address them. DeepForm addresses this challenge through cotangent-weighted Laplacian aggregation, which encodes the same force-equilibrium weights used in the DR solver itself.

1.1 Contributions

DeepForm is the first data-driven GNN surrogate demonstrated for tensile membrane form-finding across six anticlastic typologies in a single shared network, generalising across new boundary configurations within each trained typology without retraining at inference time. Aggregation weights are derived from the discrete cotangent Laplacian – the same formula used in the DR solver – embedding physics-consistent neighbourhood weighting directly into the GNN message-passing. A per-typology displacement normalisation scheme accounts for the large differences in displacement magnitude across structure types, enabling

effective multi-typology learning within a single network. The approach achieves sub-centimetre mean error on smooth minimal-surface typologies (hypar saddle 0.29 cm, Scherk 0.26 cm, Enneper 0.75 cm) and 2–3 cm mean error on high-curvature typologies (monkey saddle, sinc-difference, Kresge shell). Finally, DeepForm achieves 11× to 46× speedup over the DR solver on identical CPU hardware, enabling real-time form-finding feedback for interactive design tools.

1.2 Paper Organisation

Section 2 reviews related work. Section 3 describes the problem formulation, graph representation, GNN architecture, and training procedure. Section 4 presents experimental results. Section 5 discusses implications, limitations, and future work. Section 6 concludes.

2 Related Work

2.1 Classical Form-Finding Methods

Three families of computational form-finding methods have been widely adopted. The Force Density Method (FDM) [11, 12] linearises the form-finding problem by expressing internal forces as a product of a force density ratio and element length, allowing equilibrium equations to be solved as a linear system. Extensions to non-minimal surfaces and frame-supported boundaries have been proposed [13, 14]. The Updated Reference Strategy [15] handles anisotropic prestress naturally but requires careful parameter selection. Dynamic Relaxation [1, 2] is geometrically nonlinear by formulation and handles arbitrary boundary conditions, but its iterative character is computationally expensive. Stress-adapted finite element form-finding [14] provides a unified framework for form-finding and subsequent structural analysis.

2.2 Physics-Informed Neural Networks for Form-Finding

The application of PINNs to TMS form-finding was pioneered by Kabasi et al. [3], who reformulated form-finding as the solution of the Euler–Lagrange equation for minimal surfaces. Their mesh-free approach avoids DR convergence issues and was demonstrated on several case studies of frame-supported TMS under isotropic prestress. Kabasi et al. [4] extended this work to non-minimal surfaces and anisotropic prestress using a modified Laplace equation. Kabasi et al. [5] further extended the framework to non-minimal surfaces using the general theory of curved interfaces.

These contributions establish PINNs as a viable alternative to mesh-based form-finding methods. However, a fundamental limitation remains: each PINN must be trained from scratch for a specific boundary configuration, taking minutes per geometry. The present work addresses this limitation by learning a general solution operator across a family of problems.

2.3 GNN Surrogates for Structural Mechanics

Surrogate modelling with GNNs has been applied to structural engineering to replace expensive simulations with fast approximate models [16]. Pfaff et al. [9] demonstrated that GNNs can serve as accurate surrogates for mesh-based physical simulation across aerodynamics, structural mechanics, and cloth. Sanchez-Gonzalez et al. [17] showed that graph networks can serve as learnable physics engines for rigid-body inference and control, establishing the encoder-processor-decoder paradigm that DeepForm builds on. Gladstone et al. [10] identified limitations of standard MeshGraphNets on time-independent static problems related to long-range information propagation, and proposed edge-augmented architectures to address them. StructGNN [18] demonstrated GNN surrogates achieving over 99% accuracy for structural displacement prediction in the same journal as this work. Lamsal and Bhandari [19] demonstrated a neural network surrogate achieving 95–99% accuracy for failure-load prediction in concrete D-regions at three orders of magnitude lower computational cost than the reference solver, further evidencing the applicability of data-driven surrogates across structural engineering domains. Convolutional neural networks have been applied to stress field prediction [20]. To the best of our knowledge, no prior work has proposed a GNN surrogate for tensile membrane form-finding that generalises across boundary configurations within a single trained network.

3 Methodology

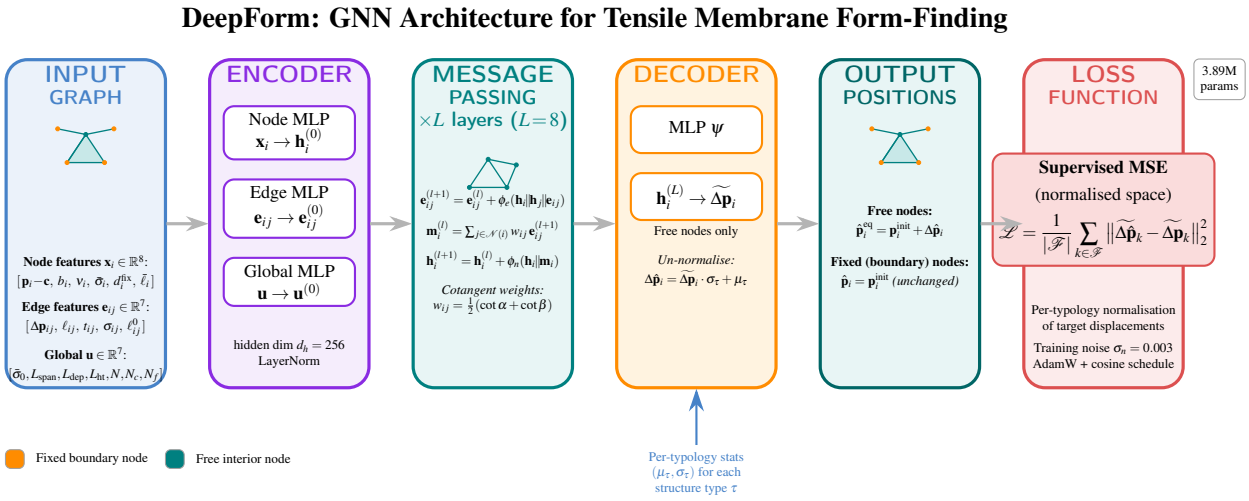

Figure 1: DeepForm GNN architecture for tensile membrane form-finding. Node, edge, and global features are independently encoded by shared MLPs into d_h=256-dimensional embeddings. Eight GraphNetBlock layers (L=8) perform residual edge updates followed by cotangent-weighted aggregation ($w_{ij} = \frac{1}{2}(\cot\alpha + \cot\beta)$, matching the DR force formulation) and residual node updates. The decoder maps final node embeddings to predicted 3-D displacements $\Delta\hat{\boldsymbol{p}}_i$; predictions are un-normalised using per-typology statistics (μ_τ, σ_τ) before reconstruction. Total trainable parameters: 3.89M.

3.1 Problem Formulation

We consider the form-finding problem for tensile membrane structures with prescribed (frame- or cable-supported) boundaries. The input consists of:

- A set of N_b boundary nodes with prescribed 3D positions $\{\mathbf{p}_i\}_{i \in \mathscr{B}}$, representing fixed support points.
- Cable elements $\{(i, j)\}$ connecting boundary nodes with prescribed stiffness EA and prestress F_{pre}.
- A triangulated membrane mesh over the interior domain with prescribed isotropic prestress σ_0 (kN/m).

The output is the set of equilibrium positions of the N_f free interior nodes $\{\mathbf{p}_k\}_{k \in \mathscr{F}}$ such that:

$$\sum_{j \in \mathscr{N}(k)} \mathbf{f}_{kj} = \mathbf{0}, \qquad \forall k \in \mathscr{F} \tag{1}$$

where $\mathbf{f}_{kj}$ denotes the force exerted on node k by neighbour j, and $\mathcal{N}(k)$ is the neighbourhood of node k.

DeepForm predicts *displacement* from the initial flat configuration to the equilibrium position:

$$\Delta\mathbf{p}_k = \mathbf{p}_k^{\text{eq}} - \mathbf{p}_k^{\text{init}}, \qquad k \in \mathscr{F} \tag{2}$$

Predicting displacement rather than absolute position makes the learning task approximately scale-invariant across structures of different sizes [9].

3.2 Graph Representation

The membrane structure is represented as a directed graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ where nodes $\mathscr{V}$ represent structural joints and edges $\mathscr{E}$ are derived from structural connectivity (membrane triangles and cable elements). All edges are bidirectional; for an undirected edge (i, j), both $(i \to j)$ and $(j \to i)$ are included to enable symmetric message passing.

3.2.1 Node Features

Each node v_i is associated with a raw feature vector $\mathbf{x}_i \in \mathbb{R}^8$:

$$\mathbf{x}_i = \left[x_i - c_x,\ y_i - c_y,\ z_i - c_z,\ b_i,\ \nu_i,\ \bar{\sigma}_i,\ d_i^{\text{fix}},\ \bar{\ell}_i\right] \tag{3}$$

where (c_x, c_y, c_z) is the centroid of all nodes, $b_i \in \{0, 1\}$ is a boundary indicator, ν_i is the node valence, $\bar{\sigma}_i$ is the mean incident face prestress (kN/m), d_i^{fix} is the Euclidean distance to the nearest fixed node (m), and $\bar{\ell}_i$ is the mean incident edge length (m).

3.2.2 Edge Features

Each directed edge $(i \to j)$ is associated with a raw feature vector $\mathbf{e}_{ij} \in \mathbb{R}^7$:

$$\mathbf{e}_{ij} = \left[\Delta x_{ij},\ \Delta y_{ij},\ \Delta z_{ij},\ \ell_{ij},\ t_{ij},\ \sigma_{ij},\ \ell_{ij}^0\right] \tag{4}$$

where $(\Delta x, \Delta y, \Delta z) = \mathbf{p}_j - \mathbf{p}_i$ is the relative position vector (m), ℓ_{ij} is the edge length (m), $t_{ij} \in \{0, 1\}$ is the edge type (0: membrane, 1: cable), σ_{ij} is the prestress (kN/m), and ℓ_{ij}^0 is the rest length (m).

3.2.3 Global Features

A global feature vector $\mathbf{u} \in \mathbb{R}^7$ captures structure-level properties:

$$\mathbf{u} = \left[\bar{\sigma}_0,\, L_{\text{span}},\, L_{\text{depth}},\, L_{\text{height}},\, N,\, N_{\text{cables}},\, N_{\text{fixed}}\right] \tag{5}$$

No structure-type one-hot encoding is included in the input features: the model learns the displacement mapping purely from geometric and physical features. The typology label is used only at inference for output denormalisation (Section 3.4), as a post-processing step that does not affect the learned graph representation.

3.3 GNN Architecture

DeepForm adapts the MeshGraphNets encoder-processor-decoder architecture [9], replacing learned-attention aggregation with physics-derived cotangent-Laplacian weights. The total trainable parameter count is 3,891,715.

3.3.1 Encoder

Node features, edge features, and global features are independently projected to a hidden dimension $d = 256$ through shallow multi-layer perceptrons (MLPs) with Layer Normalisation. Each MLP consists of two linear layers of width d=256, followed by Layer Normalisation and ReLU activations.

3.3.2 Message-Passing Layers

The processor consists of $L = 8$ stacked GraphNetBlock layers. Each layer performs three sequential operations: a residual edge update, cotangent-weighted aggregation, and a residual node update:

$$\boldsymbol{e}_{ij}^{(l+1)} = \boldsymbol{e}_{ij}^{(l)} + \varphi_e\left(\boldsymbol{h}_i^{(l)} \,\|\, \boldsymbol{h}_j^{(l)} \,\|\, \boldsymbol{e}_{ij}^{(l)}\right), \tag{6}$$

$$\boldsymbol{m}_i^{(l)} = \sum_{j \in \mathcal{N}(i)} w_{ij}\, \boldsymbol{e}_{ij}^{(l+1)}, \tag{7}$$

$$\boldsymbol{h}_i^{(l+1)} = \boldsymbol{h}_i^{(l)} + \varphi_n\left(\boldsymbol{h}_i^{(l)} \,\|\, \boldsymbol{m}_i^{(l)}\right), \tag{8}$$

where $\|$ denotes feature concatenation, φ_e and φ_n are MLPs with Layer Normalisation, and the cotangent weights are:

$$w_{ij} = \tfrac{1}{2}(\cot\alpha_{ij} + \cot\beta_{ij}), \tag{9}$$

with α_{ij} and β_{ij} the two angles opposite edge (i, j) in the adjacent triangles. These weights are computed once on the initial flat configuration and held fixed throughout the L message-passing layers. The static approximation is adopted primarily because recomputing cotangent weights on intermediate predicted geometries would require differentiating through the triangulation, substantially complicating the training graph. The approximation is further supported by the moderate deformation magnitudes in the dataset (displacement standard deviations of 0.014–0.085 m, see Section 3.4), though for high-curvature typologies the weights on the deformed geometry may differ non-negligibly from initial values. This approximation is expected to be least accurate for high-curvature typologies with larger deformation magnitudes (monkey saddle, sinc-difference, Kresge), which is consistent with their higher prediction errors (Table 2). Edge embeddings are updated at every layer via residual connections, allowing the network to refine edge representations as node embeddings evolve. Dropout (rate 0.10) is applied to node embeddings $\boldsymbol{h}_i^{(l+1)}$ after each block during training.

3.3.3 Decoder

After L message-passing layers, node embeddings are decoded to 3D displacement vectors:

$$\Delta\hat{\boldsymbol{p}}_i = \mathrm{MLP}_{\mathrm{dec}}\left(\boldsymbol{h}_i^{(L)}\right), \qquad \hat{\boldsymbol{p}}_i^{\mathrm{eq}} = \boldsymbol{p}_i^{\mathrm{init}} + \Delta\hat{\boldsymbol{p}}_i, \tag{10}$$

Predicting displacement rather than absolute position encourages the network to learn the deformation field and makes the task approximately scale-invariant [9].

3.4 Normalisation

A critical challenge in training across multiple typologies is the large variation in displacement magnitude. Across the six typologies, the mean per-sample deformation standard deviation ranges from $\sim$0.014 m (hypar saddle) to $\sim$0.085 m (monkey saddle), a factor of approximately 6. Training with global statistics would suppress gradients for low-displacement typologies and allow high-displacement typologies to dominate the loss.

To address this, per-typology displacement normalisation is applied. For each structure type τ, displacement statistics $(\mu_\tau, \sigma_\tau) \in \mathbb{R}^3$ are computed from the training split of that type only:

$$\mathbf{z}_k = \frac{\Delta\mathbf{p}_k - \mu_\tau}{\sigma_\tau}, \qquad k \in \mathscr{F},\ \tau = \mathrm{type}(k) \tag{11}$$

At inference, predictions $\hat{\mathbf{z}}_k$ are un-normalised using stored type statistics before reconstruction (Eq. 10). Node, edge, and global features use shared z-score normalisation computed

from the full training set.

3.5 Training Data Generation

3.5.1 Anticlastic Geometry Requirement

This work considers mechanically prestressed, frame- or cable-supported membranes, for which tension-only equilibrium requires anticlastic geometry ($K < 0$) at every point. Pneumatically supported membranes, which use internal pressure to maintain equilibrium and can therefore take synclastic shapes, lie outside the scope of this paper. All six typologies in the dataset are constructed with boundary conditions that mathematically guarantee anticlastic equilibrium shapes ($K < 0$ everywhere).

3.5.2 Structure Typologies

Six anticlastic typologies are included in the dataset:

- **Hypar saddle.** Rectangular quad boundary with 4 fixed corners at prescribed heights.
- **Scherk minimal surface.** Anticlastic boundary derived from Scherk's first surface.
- **Enneper minimal surface.** Boundary parameterised by Enneper's parametric surface.
- **Monkey saddle.** Higher-order saddle boundary with three valleys and three peaks.
- **Sinc-difference saddle.** Anticlastic boundary defined by a sinc-function difference.
- **Kresge-inspired triangular shell.** Triangular plan boundary with three support points and sinusoidal rise along each edge, inspired by the triangular boundary geometry of the MIT Kresge Auditorium; implemented here as a prestressed tensile membrane.

Fig. 2 shows the DR equilibrium shapes for all six typologies, illustrating the geometric diversity of the dataset.

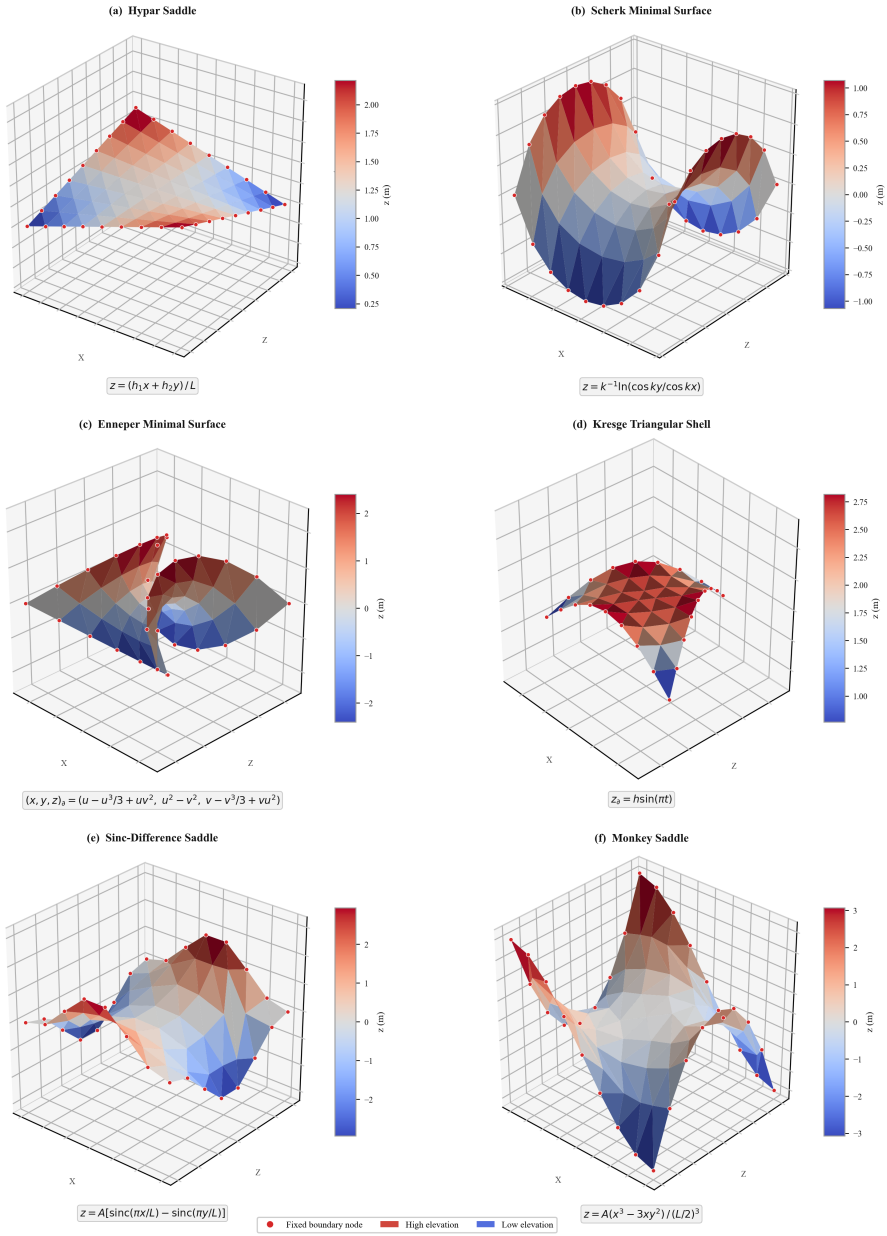

Figure 2: Dynamic-relaxation equilibrium shapes for six anticlastic membrane typologies in the DeepForm training dataset. Colour encodes nodal height (blue = low, red = high); diffuse shading emphasises curvature. Orange dots mark fixed boundary nodes. (a) Hypar saddle. (b) Scherk minimal surface. (c) Enneper minimal surface. (d) Kresge triangular shell. (e) Sinc-difference saddle. (f) Monkey saddle.

3.5.3 Dynamic Relaxation Solver

The DR solver uses the cotangent-weighted Laplacian force formulation for membrane triangles [21]:

$$\mathbf{f}_{ij\to i} = \frac{\sigma_0}{2} \cot\theta_k \left(\mathbf{p}_j - \mathbf{p}_i\right) \tag{12}$$

where θ_k is the interior angle at vertex k opposite edge (i, j) and σ_0 is the isotropic prestress (kN/m). Convergence is declared when the maximum nodal residual falls below 10^{-3} times the mean membrane prestress force $\sigma_0 \bar{\ell}$, where $\bar{\ell}$ is the mean edge length.

3.5.4 Dataset Statistics

A total of 10,987 converged equilibrium solutions are used for training and validation, divided 80/20 into training (8,789) and validation (2,198) sets using stratified sampling by structure type. An independent set of 30 held-out test cases per typology (180 total), generated separately from the training distribution, is reserved exclusively for final evaluation and never seen during training or hyperparameter selection.

3.6 Loss Function

The training loss is the mean squared displacement error in normalised space:

$$\mathscr{L} = \frac{1}{|\mathscr{F}|} \sum_{k \in \mathscr{F}} \|\hat{\mathbf{z}}_k - \mathbf{z}_k\|_2^2 \tag{13}$$

where $\hat{\mathbf{z}}_k$ and $\mathbf{z}_k$ are the predicted and target per-typology normalised displacements at free node k. During training, small Gaussian noise ($\sigma_n = 0.003$ in normalised feature space) is added to free node features to improve robustness [22].

3.7 Training Procedure

The dataset is split 80/20 into training and validation sets using stratified sampling by typology to ensure proportional representation in both splits. All random operations are seeded at 42 for reproducibility. The model is trained using the AdamW optimiser [23] with initial learning rate 1×10^{-3}, weight decay 10^{-4}, and mini-batch size 32. A cosine annealing schedule decays the learning rate to 10^{-5} over the training run. Graphs from different typologies are collated into disconnected batches using PyTorch Geometric [24]. The loss function is the mean squared error on free-node displacements in normalised space (Eq. 13). Early stopping with patience 80 epochs terminates training when the validation loss fails to improve for 80 consecutive epochs.

4 Experimental Results

4.1 Training Convergence

Fig. 3 shows the training and validation loss curves and the per-epoch per-typology validation error. The model trained for 374 epochs before early stopping triggered (patience 80), achieving a best validation loss of 0.01206 at epoch 294 on 10,987 samples. Training took approximately 3 hours on an Apple M3 Pro (MPS device). The training and validation curves track each other consistently throughout training, with no sign of divergence, indicating the model generalises without severe overfitting, consistent with weight decay (10^{-4}) and dropout (0.10). Per-typology normalisation resolves the displacement-scale imbalance described in Section 3.4; minimal-surface typologies (hypar, Scherk, Enneper) converge to lower normalised error than high-curvature types (monkey saddle, sinc-difference, Kresge), consistent with the complexity of their equilibrium shapes.

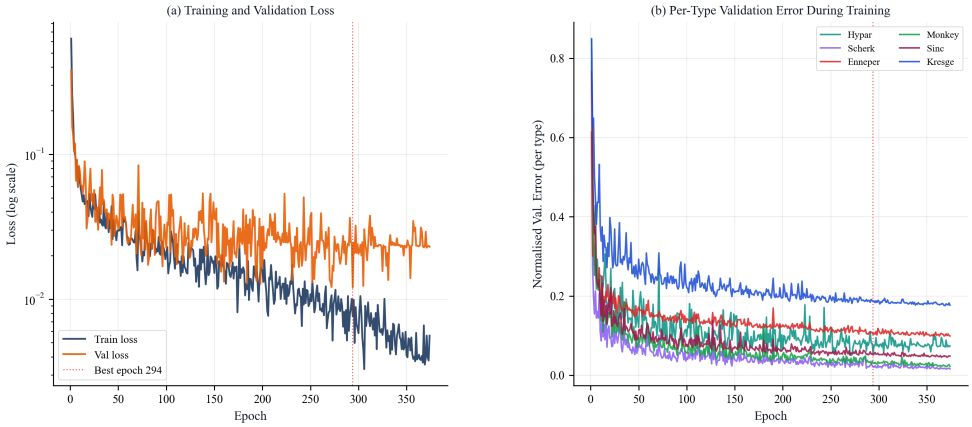

Figure 3: Training convergence on 10,987 samples across six anticlastic typologies (80/20 stratified split by type). (a) Training and validation loss (log scale); dotted line marks the best epoch. (b) Per-type normalised validation error during training: minimal-surface typologies (Scherk, Enneper, Hypar) converge faster and to lower error than high-curvature types (Monkey saddle, Sinc saddle, Kresge).

Table 1 summarises the training configuration and convergence metrics.

4.2 Form-Finding Accuracy

Table 2 summarises quantitative results on 30 held-out random test cases per typology.

Fig. 4 shows the benchmark scorecard across all six typologies.

Table 1: Training configuration and convergence metrics.

Hyperparameter / Metric	Value
Hidden dimension d_h	256
Message-passing layers L	8
Dropout rate	0.10
Learning rate	1×10^{-3}
LR schedule	Cosine annealing
Weight decay	10^{-4}
Mini-batch size	32
Optimiser	AdamW
Training noise std σ_n	0.003
Random seed	42
Trainable parameters	3,891,715
Training epochs (early stopped)	374
Best validation loss	0.01206
Training time (Apple M3 Pro)	$\sim$3 h

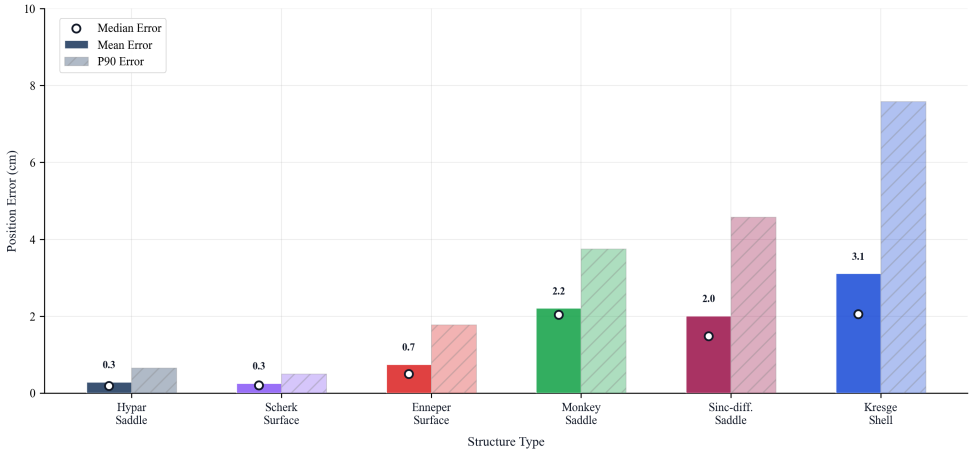

Figure 4: Mean and P90 position errors (cm) for the six anticlastic typologies, each evaluated on 30 held-out random samples. Solid bars show mean error; hatched bars show P90 error; white circles mark the median. Hypar, Scherk, and Enneper achieve sub-centimetre mean errors; monkey saddle, sinc-difference saddle, and Kresge shell reach 2–3 cm.

The hypar saddle achieves the highest accuracy (0.29 cm mean, 0.19 cm median), representing approximately 0.036% of an 8 m reference span. Scherk and Enneper minimal surfaces also achieve sub-centimetre mean errors (0.26 cm and 0.75 cm respectively), consistent with their smooth, well-conditioned equilibrium shapes.

The mean/median ratio across typologies is 1.1–1.6×, indicating the model produces consistent predictions with only occasional outliers on extreme parameter combinations. For monkey saddle (2.21 cm mean, 2.04 cm median) and sinc-difference saddle (2.01 cm

Table 2: Per-typology prediction accuracy on 30 held-out random test cases. Errors in centimetres (physical units). Speedup is wall-clock DR time divided by DeepForm inference time on identical CPU.

Typology	Mean (cm)	Median (cm)	P90 (cm)	Speedup
Hypar saddle	**0.29**	**0.19**	0.65	11×
Scherk minimal surface	**0.26**	0.21	0.50	46×
Enneper minimal surface	**0.75**	0.50	1.78	29×
Sinc-difference saddle	2.01	1.48	4.58	41×
Monkey saddle	2.21	2.04	3.75	45×
Kresge triangular shell	3.11	2.05	7.58	25×

mean, 1.48 cm median), the ratio is near unity, showing that errors are distributed uniformly across the test set rather than being driven by rare extreme cases. The larger mean/median gap for hypar (0.29/0.19 = 1.5×) and Kresge (3.11/2.05 = 1.5×) reflects mild positive skew typical of geometrically diverse parameter sweeps.

4.3 Qualitative Form Comparison

Fig. 5 shows the three stages of form-finding for a representative hypar saddle: the initial flat configuration, the DR solver ground-truth equilibrium, and the DeepForm GNN prediction. The predicted surface is visually indistinguishable from the DR ground truth.

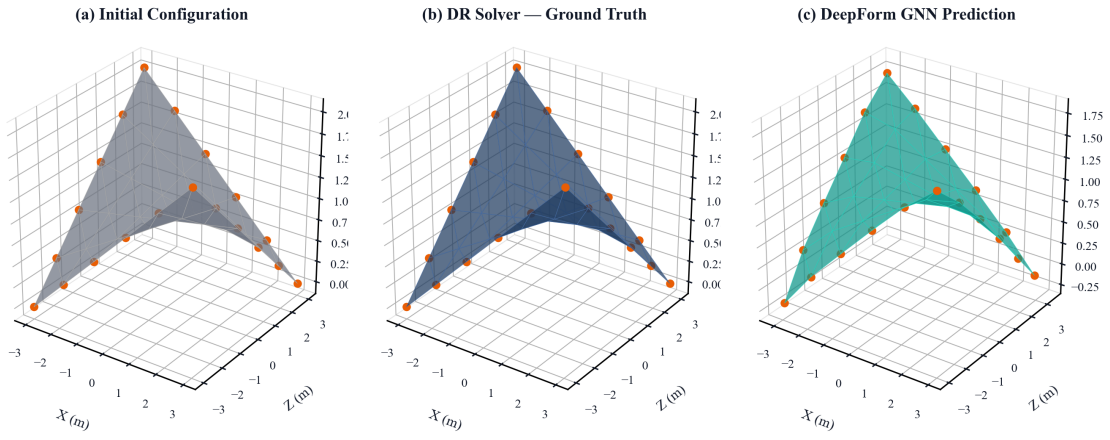

Figure 5: Form-finding sequence for a representative hypar saddle (span = 6 m, rise = 2 m, pre-stress = 3 kN/m): (a) initial flat configuration, (b) DR solver ground-truth equilibrium, and (c) DeepForm GNN prediction. Orange dots mark fixed boundary nodes. The predicted surface (c) is visually indistinguishable from the DR ground truth (b).

4.4 Spatial Accuracy

Fig. 6 overlays each predicted equilibrium surface on the DR ground-truth surface for all six typologies. The semi-transparent grey ghost is largely hidden beneath the predicted surface for hypar and Scherk (panels a–b); for Enneper (panel c) small offsets are visible at the boundary curves, consistent with its sub-centimetre but slightly higher mean error of 0.75 cm. For Kresge, sinc-difference, and monkey saddle (panels d–f), small offsets between the prediction and the grey ghost become visible in regions of high curvature and large deformation. A shared 0–4 cm colour scale across all six panels localises where the largest position errors occur; values above 4 cm are clipped to red (reached by sinc-difference and Kresge at their highest-curvature nodes). The overall form is well recovered for all six typologies, consistent with the quantitative results in Table 2.

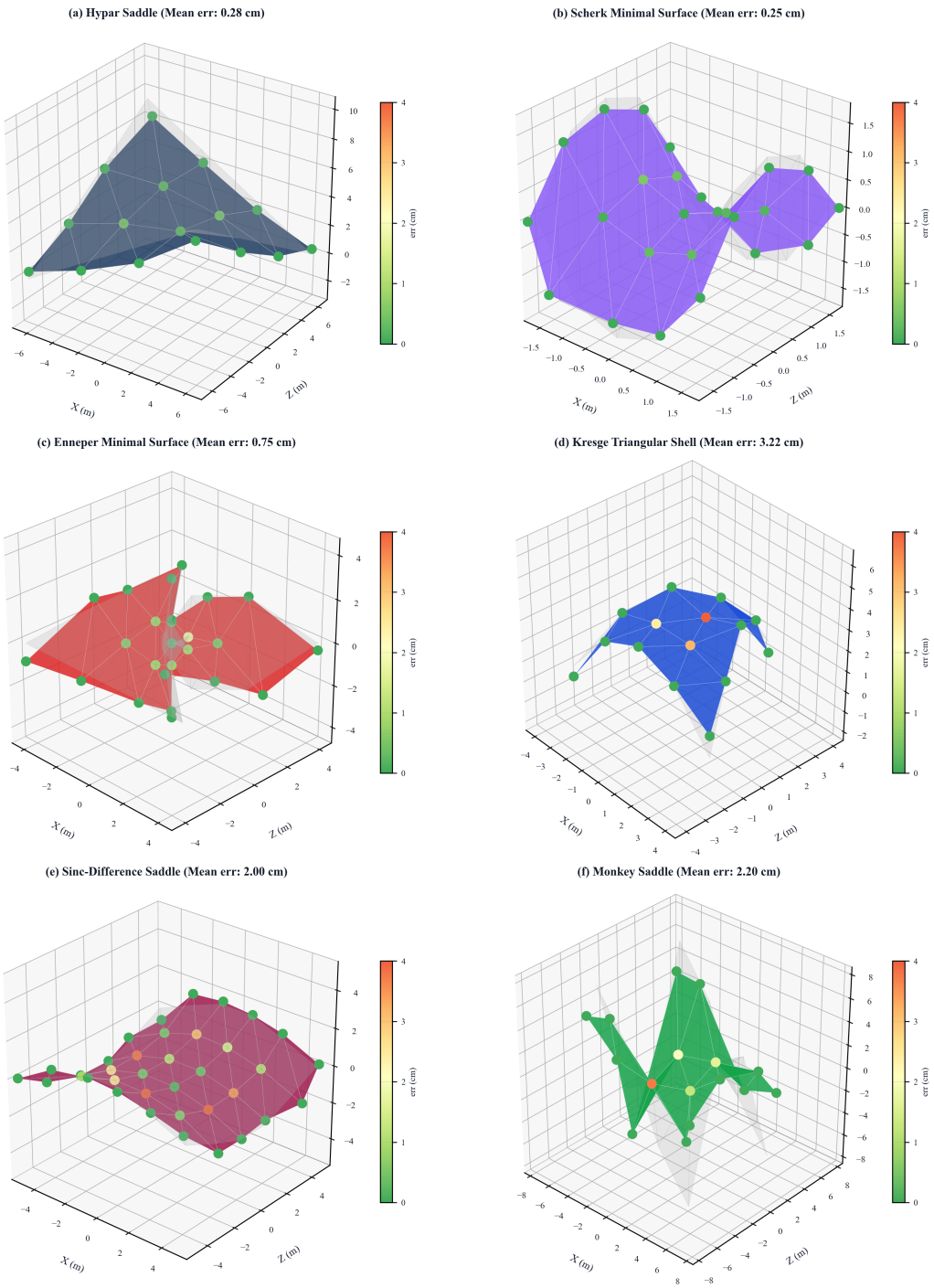

Figure 6: Spatial accuracy of DeepForm across all six typologies (representative test instances; 30-case benchmark averages in Table 2). Each panel overlays the DR ground-truth surface (semi-transparent grey) with the DeepForm prediction (typology colour); all nodes are coloured by per-node 3-D position error on a shared green (0 cm) to red (4 cm) scale (values above 4 cm clipped to red); fixed boundary nodes are prescribed and therefore carry zero predicted error (shown in green). (a) Hypar saddle (mean: 0.28 cm); (b) Scherk minimal surface (mean: 0.25 cm); (c) Enneper minimal surface (mean: 0.75 cm); (d) Kresge triangular shell (mean: 3.22 cm); (e) Sinc-difference saddle (mean: 2.00 cm); (f) Monkey saddle (mean: 2.20 cm).

4.5 Computational Speed

Table 3 compares wall-clock times of the DR solver and DeepForm on identical CPU hardware (Apple M3 Pro, single core).

Table 3: Computational time comparison on CPU, averaged over 30 test cases per typology. DeepForm inference includes graph construction and forward pass. DR timings correspond to the mesh densities used in this study (3–5 nodes per edge, ∼30–80 free nodes per typology) and use a vectorised NumPy implementation with convergence tolerance of $10^{-3}\times$ mean prestress force $\sigma_0\bar{\ell}$. Production-density meshes (mesh density 10 or higher, 200+ free nodes) typically require 10,000–20,000 iterations and several seconds to minutes; DeepForm inference time is expected to scale sub-linearly with mesh size owing to its fixed-depth architecture; benchmarking on production-density meshes is left as future work.

Typology	DR (ms)	DeepForm (ms)	Speedup
Hypar saddle	∼110	∼10	11×
Scherk minimal surface	∼457	∼10	46×
Enneper minimal surface	∼289	∼10	29×
Sinc-difference saddle	∼405	∼10	41×
Monkey saddle	∼446	∼10	45×
Kresge triangular shell	∼252	∼10	25×

DeepForm inference is consistently around 10 ms across all six typologies, regardless of structural complexity. The lower speedup for hypar (11×) reflects the fact that DR converges in fewer iterations for the simpler hypar geometry; the GNN advantage is largest for typologies such as Scherk (46×) and monkey saddle (45×), where DR requires more iterations to satisfy the equilibrium tolerance.

5 Discussion

5.1 Comparison with PINN-Based Form-Finding

Table 4 compares DeepForm with the PINN-based approaches of Kabasi et al. [3–5].

The key distinction is the training paradigm. PINN methods train a separate network for each boundary configuration. DeepForm trains once and generalises. These approaches address different design needs: PINN methods may be preferable when high accuracy on a single, fixed configuration is required and per-geometry training time is acceptable; DeepForm is optimised for early-stage design exploration requiring rapid evaluation of many configurations.

Table 4: Comparison of DeepForm with PINN-based form-finding approaches. 'Per-geom.' in the Retrain column indicates retraining required per boundary configuration.

Method	Scope	Retrain?	Inference
Kabasi et al. [3]	Frame-supported TMS, isotropic prestress	Per-geom.	Minutes
Kabasi et al. [4]	Frame-supported, anisotropic prestress	Per-geom.	Minutes
Kabasi et al. [5]	Non-minimal surfaces, curved interfaces	Per-geom.	Minutes
DeepForm (this work)	6 anticlastic typologies	Once only	$<$10 ms

5.2 The Mean–Median Gap

The mean/median ratio of 1.1–1.6$\times$ across typologies indicates the model produces consistent predictions across the test set without systematic outliers. For the three higher-error typologies (monkey saddle, sinc-difference, Kresge), a small number of extreme configurations such as very large spans and high-amplitude boundary conditions produce higher errors that modestly inflate the mean above the median. This is consistent with the geometric difficulty of these typologies: the static cotangent weights computed on the initial flat configuration deviate most from the true equilibrium weights for surfaces that undergo large curvature changes, precisely the high-error typologies. Edge-augmented architectures [10] that propagate long-range boundary information are a natural next step to address accuracy on large-span, high-curvature configurations.

5.3 Limitations

Several limitations should be acknowledged.

First, the held-out test cases are drawn from the same parameter distribution as the training data: spans, prestress values, corner heights, and mesh densities are sampled from the same ranges as the training set. The model's accuracy therefore demonstrates within-distribution generalisation across new boundary configurations within each typology, rather than out-of-distribution generalisation to unseen parameter regimes or to anticlastic surfaces outside the six trained typologies (e.g., Costa or Helicoid minimal surfaces). Quantifying performance under distribution shift is left as future work.

Second, prediction accuracy varies across typologies. Sub-centimetre mean errors are

achieved on smooth minimal-surface typologies (hypar, Scherk, Enneper), while geometrically more complex typologies (monkey saddle, sinc-difference, Kresge) produce 2–3 cm mean errors. The mean–median gap of 1.1–1.6× on the complex typologies indicates occasional outlier predictions on extreme parameter combinations that lie at the boundary of the training distribution.

Third, the training dataset covers only isotropic prestress. Anisotropic fabric materials, common in engineering practice, are not yet modelled.

Fourth, all training examples use mesh densities of 3–5 nodes per edge (approximately 30–80 free interior nodes per structure). The model's generalisation to finer meshes has not been evaluated. This is not merely an evaluation gap – finer meshes yield larger graphs with more free nodes than any training example, which may require architectural extensions such as graph coarsening or multi-scale message-passing beyond the current fixed-depth framework.

5.4 Future Work

Several directions are identified for extending this work. Replacing stochastic boundary generators with closed-form parametric equations for all typologies would produce more uniformly distributed training samples and reduce the mean–median error gap on complex typologies. Adding long-range skip connections between distant boundary nodes, following Gladstone et al. [10], would improve accuracy for large-span and high-curvature surfaces; equivariant GNN architectures [25] that respect the E(3) symmetry of 3D space are a natural candidate for further improving sample efficiency and generalisation to unseen orientations. Incorporating warp/weft stiffness ratios as additional edge features would extend the framework to anisotropic fabric materials common in engineering practice. A near-term goal is deployment as a real-time feedback module within an interactive structural design platform.

6 Conclusion

This paper has presented DeepForm, a Graph Neural Network surrogate model for real-time tensile membrane form-finding. DeepForm is trained once on a dataset of 10,987 Dynamic Relaxation solver outputs spanning six anticlastic membrane typologies. At inference time, any new boundary configuration is processed in under 10 ms on CPU, without any retraining.

The key technical contributions are: (i) a MeshGraphNets-style architecture with residual edge updates and cotangent-weighted aggregation that embeds the same physics as the DR solver; (ii) relative position vectors in edge features that encode spatial force direction;

(iii) per-typology displacement normalisation handling approximately a factor of 6 in deformation scale across typologies; and (iv) displacement prediction making the model approximately scale-invariant.

Experimental results demonstrate 0.29 cm mean error on hypar saddle and sub-centimetre accuracy on Scherk (0.26 cm) and Enneper (0.75 cm) minimal surfaces, with speedups of $11\times$ to $46\times$ over the DR solver across all six typologies. This is the first data-driven GNN surrogate demonstrated for tensile membrane form-finding across multiple anticlastic typologies in a single shared network.

References

[1] M. R. Barnes. *Form-finding and analysis of tension space structures by dynamic relaxation*. PhD thesis, City University London, 1977.

[2] A. S. Day. An introduction to dynamic relaxation. *The Engineer*, 219:218–221, 1965.

[3] S. Kabasi, A. L. Marbaniang, and S. Ghosh. Physics-informed neural networks for the form-finding of tensile membranes by solving the Euler–Lagrange equation of minimal surfaces. *Thin-Walled Structures*, 182:110309, 2023.

[4] S. Kabasi, A. L. Marbaniang, and S. Ghosh. Form-finding of frame-supported non-minimal tensile membrane structures for anisotropic prestress using physics-informed neural networks. *Structural and Multidisciplinary Optimization*, 67:56, 2024.

[5] S. Kabasi, A. L. Marbaniang, and S. Ghosh. Form finding of non-minimal tensile membrane structures based on the general theory of curved interfaces using physics-informed neural networks. *Proceedings of the Institution of Civil Engineers — Engineering and Computational Mechanics*, 178(2):39–55, 2025.

[6] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009.

[7] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. Neural message passing for quantum chemistry. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1263–1272, 2017.

[8] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, C. Gulcehre, F. Song, A. Ballard, J. Gilmer, G. E. Dahl, A. Vaswani, K. Allen, C. Nash, V. Langston,

C. Dyer, N. Heess, D. Wierstra, P. Kohli, M. Botvinick, O. Vinyals, Y. Li, and R. Pascanu. Relational inductive biases, deep learning, and graph networks. arXiv preprint arXiv:1806.01261, 2018.

[9] T. Pfaff, M. Fortunato, A. Sanchez-Gonzalez, and P. Battaglia. Learning mesh-based simulation with graph networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021. Available: https://openreview.net/forum?id=roNqYL0_XP.

[10] R. J. Gladstone, H. Rahmani, V. Suryakumar, H. Meidani, M. D'Elia, and A. Zareei. Mesh-based GNN surrogates for time-independent PDEs. *Scientific Reports*, 14:3394, 2024.

[11] K. Linkwitz and H. J. Schek. Einige Bemerkungen zur Berechnung von vorgespannten Seilnetzkonstruktionen. *Ingenieur-Archiv*, 40(3):145–158, 1971.

[12] H. J. Schek. The force density method for form finding and computation of general networks. *Computer Methods in Applied Mechanics and Engineering*, 3(1):115–134, 1974.

[13] R. B. Haber and J. F. Abel. Initial equilibrium solution methods for cable reinforced membranes. Part I — Formulations. *Computer Methods in Applied Mechanics and Engineering*, 30(3):263–284, 1982.

[14] R. Wuechner and K. U. Bletzinger. Stress-adapted numerical form finding of prestressed surfaces by the updated reference strategy. *International Journal for Numerical Methods in Engineering*, 64(2):143–166, 2005.

[15] K. U. Bletzinger and E. Ramm. A general finite element approach to the form finding of tensile structures by the updated reference strategy. *International Journal of Space Structures*, 14(2):131–145, 1999.

[16] H. Adeli and C. Yeh. Perceptron learning in engineering design. *Microcomputers in Civil Engineering*, 4(4):247–256, 1989.

[17] A. Sanchez-Gonzalez, N. Heess, J. T. Springenberg, J. Merel, M. Riedmiller, R. Hadsell, and P. Battaglia. Graph networks as learnable physics engines for inference and control. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 4467–4476, 2018.

[18] Y.-T. Chou, W.-T. Chang, J. G. Jean, K.-H. Chang, Y.-N. Huang, and C.-S. Chen. StructGNN: An efficient graph neural network framework for static structural analysis. *Computers & Structures*, 299:107385, 2024.

[19] Sandesh Lamsal and Rubi Bhandari. A strut and tie neural network surrogate for failure-load prediction of concrete D-regions designed by the compatible stress field method. *engrXiv*, 2026. https://doi.org/10.31224/7250.

[20] Z. Nie, H. Jiang, and L. B. Kara. Stress field prediction in cantilevered structures using convolutional neural networks. *Journal of Computing and Information Science in Engineering*, 20(1):011002, 2020.

[21] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proceedings of SIGGRAPH*, pages 317–324, 1999.

[22] A. Sanchez-Gonzalez, J. Godwin, T. Pfaff, R. Ying, J. Leskovec, and P. Battaglia. Learning to simulate complex physics with graph networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 8459–8468, 2020.

[23] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019. Available: https://openreview.net/forum?id=Bkg6RiCqY7.

[24] M. Fey and J. E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *Proceedings of the ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019. Available: https://arxiv.org/abs/1903.02428.

[25] V. G. Satorras, E. Hoogeboom, and M. Welling. E(n) equivariant graph neural networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 9323–9332, 2021.