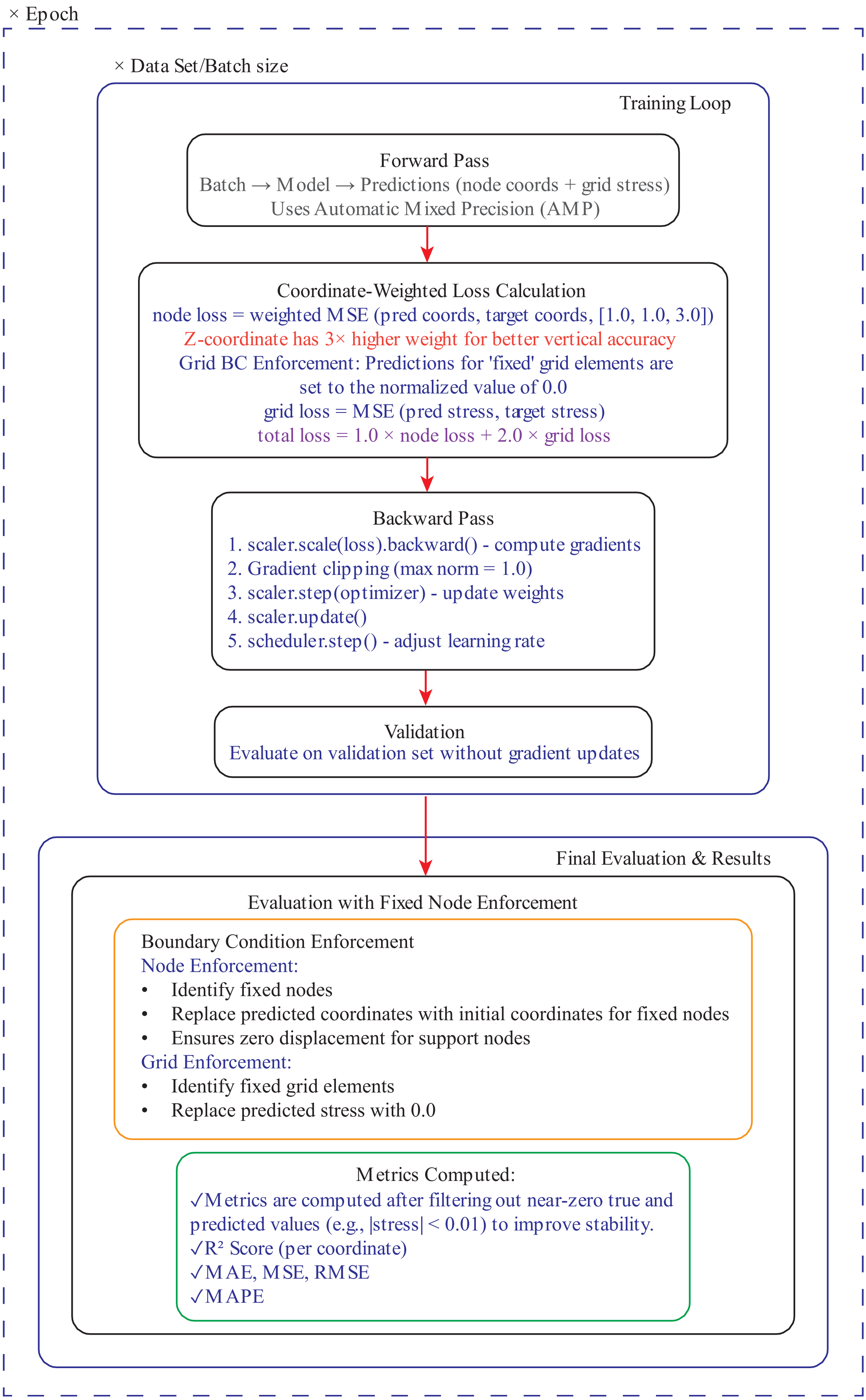


Fig. 4.8: Training Loop of the GNN model.

4.4 Model Training and Testing

Model training is conducted as described in Section 4.3. In this section, we will evaluate the model's performance. Fig. 4.9 illustrates the Training history plots (loss vs. epochs). Model testing can be done in two stages. In the first stage, we will evaluate the overall model performance (Performance Assessment Stage) using evaluation metrics. In the second stage (Predictive Accuracy Analysis), we will evaluate the model's performance by predicting two grid-shells and comparing the predictions with actual data.

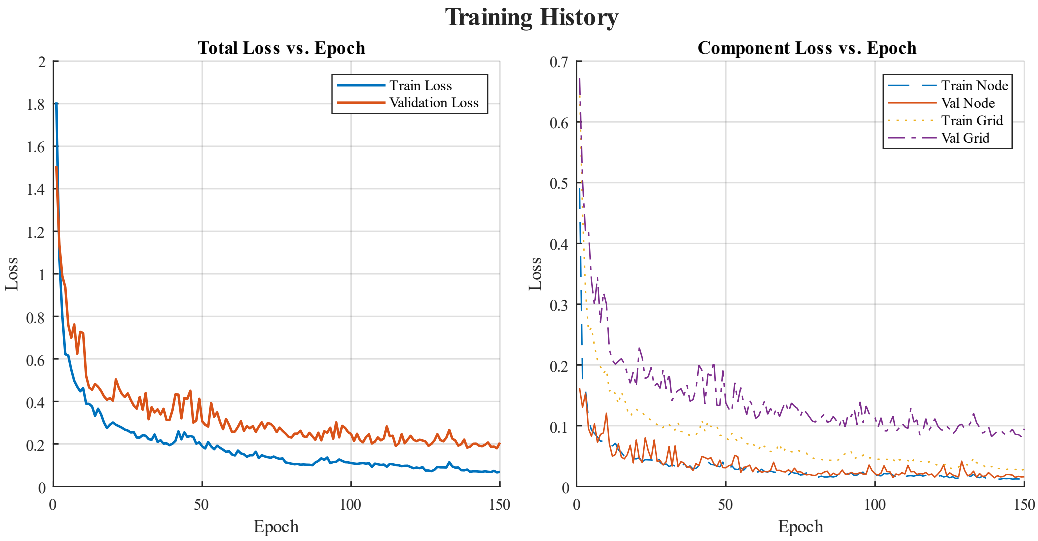


Fig. 4.9: Training history curves for node and grid embeddings.

4.4.1 Performance Assessment Stage

The first stage of model testing involves a quantitative assessment using established regression evaluation metrics. These metrics provide objective measures of the model's predictive accuracy by comparing the outputs from the Graph Neural Network (GNN) — specifically, the predicted node coordinates and grid stresses — against ground-truth values derived from form-finding. In this study, four key metrics are utilized: The Coefficient of Determination (R^2), Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE) [119], [120].

These metrics are particularly well-suited for regression tasks in structural engineering, where predictions involve continuous variables such as displacements (measured in meters) and stresses. They facilitate the evaluation of both overall model fit and error magnitudes. MAE and RMSE focus on absolute deviations, which are helpful for coordinate predictions in meters. In contrast, MAPE provides insights into relative error, expressed as percentages, making it ideal for normalized comparisons across different scales.

The Coefficient of Determination (R^2) quantifies the proportion of variance in the dependent variable that is predictable from the independent variables. It is calculated as:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{4.6}$$

where y_i is the actual value, $\hat{y}_i$ is the predicted value, $\bar{y}$ is the mean of the actual values, and n is the number of samples. R^2 ranges from -∞ to 1, with values closer to 1 indicating a better fit (i.e., the model explains most of the data variability), while negative values suggest the model performs worse than a simple mean-based predictor. However, R^2 can be sensitive to outliers and does not directly measure prediction bias, making it complementary to error-based metrics [119], [121].

The Mean Absolute Error (MAE) measures the average magnitude of errors in the predictions, without considering their direction. Its formula is:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{4.7}$$

The Mean Absolute Error (MAE) measures the distance between our coordinate predictions and the actual values, expressed in meters. It provides a clear indication of the average error for each prediction. Unlike squared-error metrics, MAE is less affected by outliers, making it useful for structural applications where accurate placement—like node movements—matters for design safety. We use MAE to evaluate the Graph Neural Network's (GNN) performance in predicting final node positions and axial stresses, ensuring that average errors remain within acceptable limits, such as sub-millimeter precision for detailed models (for displacements). However, MAE does not penalize larger errors more heavily, which may not fully reflect the impact of significant mistakes in critical structural elements [120], [122].

The Root Mean Square Error (RMSE) addresses this limitation by emphasizing larger errors through squaring. It is defined as:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{4.8}$$

Like MAE, RMSE is expressed in meters for dimensional consistency, making it directly comparable to the scale of the data. RMSE is widely used in regression evaluation for its sensitivity to outliers, which is crucial in grid-shell analysis, where extreme errors in stress predictions could indicate failure modes. In practice, RMSE is computed separately for each coordinate (X, Y, Z) and stress values, allowing targeted analysis—e.g., higher RMSE in Z might highlight vertical deformation challenges. A key advantage is its differentiability, which aids in model optimization, although it can be inflated by variance in the data distribution [119], [121].

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\% \tag{4.9}$$

with safeguards for zero-division (e.g., adding a small epsilon). Expressed as a percentage, MAPE is ideal for comparing relative errors across different magnitudes, such as varying stress levels in grids. It is beneficial in this study for stress predictions, where relative accuracy matters for material utilization assessments. However, MAPE can be biased toward low-magnitude values and unstable near zero, necessitating careful application alongside absolute metrics like MAE and RMSE [120], [123].

The metrics are calculated on the validation set after converting predictions back to physical units, which allows for real-world interpretation. Acceptability thresholds (e.g., $R^2 > 0.9$, MAE < 0.01 m) are based on benchmarks from structural engineering. Additionally, we provide a breakdown by coordinate for each node, prioritizing the Z coordinate due to its significance in relation to gravity.

Fig. 4.10 illustrates the accuracy plots for the X, Y, and Z coordinates, as well as the axial stresses of the grid members. Table 4.1 presents the evaluation metric values for node predictions, while Table 4.2 displays those for member stress predictions.

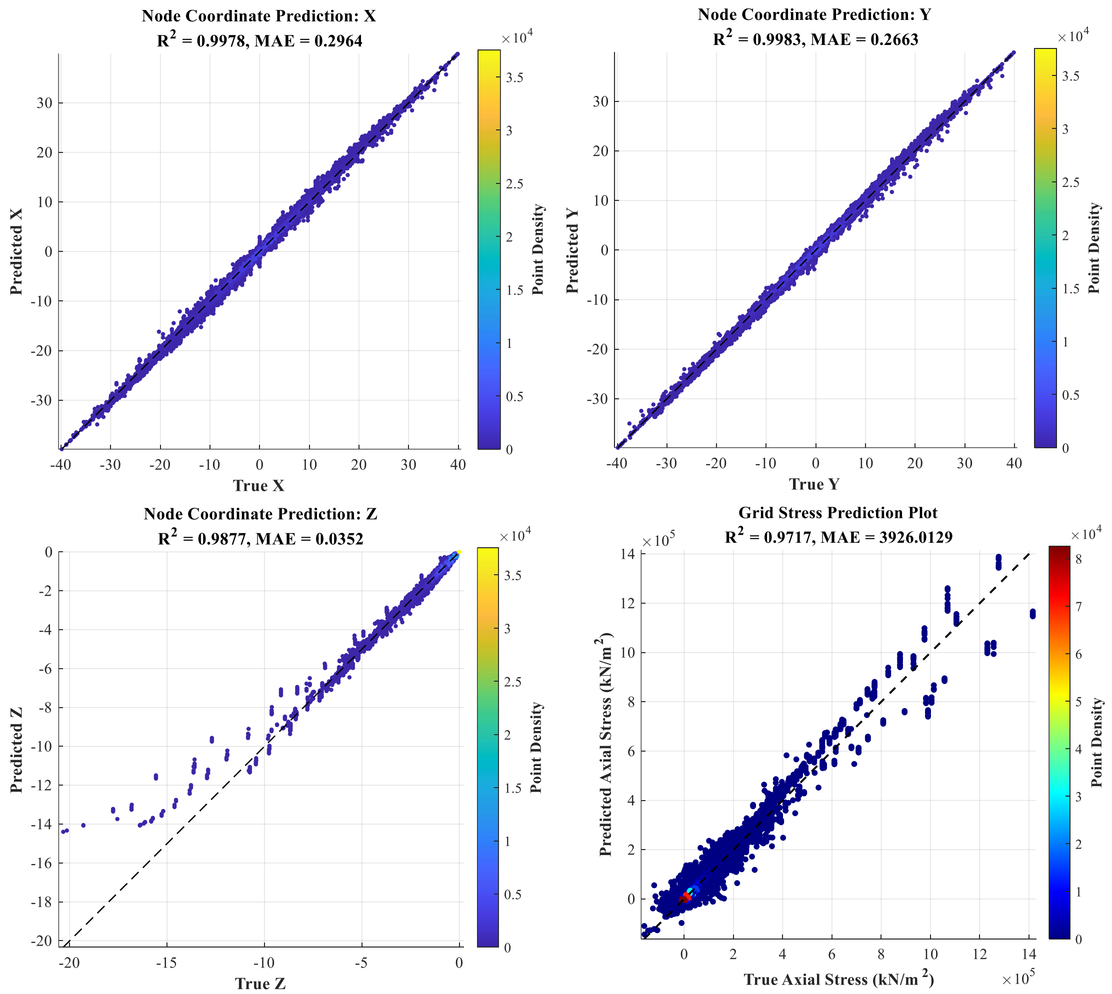


Fig. 4.10: Accuracy plots of X, Y, Z coordinates and axial stresses.

Table 4.1: Evaluation metric values for node predictions.

Prediction	R^2	MAE (m)	Root Mean Square Error (m)	MAPE (%)
X	0.9978	0.2964	0.4526	80.7
Y	0.9983	0.2663	0.4072	35.2
Z	0.9877	0.0352	0.1246	7.2

Table 4.2: Evaluation metric values for grid predictions.

Prediction	R^2	MAE (kPa)	Root Mean Square Error (kPa)	MAPE (%)
Member stresses	0.9717	3926	8364	211.7

The evaluation metrics presented in Section 4.4.1 reveal a high-performing GNN model overall, with particularly strong results in node coordinate predictions (R^2 = 0.9978 for X, 0.9983 for Y, and 0.9877 for Z) and reasonable accuracy in grid stress predictions (R^2 = 0.9717). However, discrepancies across metrics—such as elevated MAPE values despite low absolute errors—warrant deeper analysis. This section elaborates on the potential reasons for these accuracies, drawing from the model's architecture (Section 4.3.2), data characteristics (Section 4.2.2), and domain-specific factors in grid-shell mechanics. Explanations are grounded in the interplay between graph structure, training dynamics, and physical phenomena, with references to relevant literature on GNNs in structural engineering.

1. **Node Prediction Analysis**

Node predictions focus on the final deformed coordinates (X, Y, Z), which represent structural displacements after form-finding. The high R^2 scores across all axes indicate that the model captures over 99.7% of the variance in X and Y, and 98.77% in Z, suggesting an excellent overall fit. This can be attributed to the heterogeneous GAT architecture's effective message passing, which leverages intra-panel (node-to-node) and inter-panel (grid-to-node) connections to propagate geometric and load information across the graph. The bidirectional edges and edge attributes (distances) enhance spatial awareness, allowing the model to learn deformation patterns induced by gravity and wind loads.

- **X and Y Coordinates:** The near-perfect R^2 (0.9978 for X, 0.9983 for Y) reflects the model's ability to predict minimal horizontal shifts, as grid-shell form-finding primarily involves vertical adjustments. MAE (0.2964 m for X, 0.2663 m for Y) and RMSE (0.4526 m for X, 0.4072 m for Y) are low in absolute terms, indicating average errors well below typical grid-shell spans (5 – 50 m), which is suitable for engineering tolerances. However, the inflated MAPEs (80.7% for X, 35.2% for Y) arise from the relative-error formulation:

actual horizontal displacements are often near zero (e.g., due to symmetric loading or fixed boundaries), leading to division by small denominators and resulting in inflated percentages. This is a known limitation of MAPE in datasets with low-magnitude targets.

- **Z Coordinate**: The slightly lower R^2 (0.9877) compared to X/Y is offset by vastly superior error metrics (MAE: 0.0352 m, RMSE: 0.1246 m, MAPE: 7.2%), highlighting the model's focus on vertical deformations—the dominant mode in tensioned grid-shells. The weighted loss function improves accuracy (coordinate weights: [1.0, 1.0, 3.0] for X, Y, and Z), prioritizing Z during training to align with the laws of gravity. The very low MAPE results from larger absolute Z displacements (e.g., sagging or arching), thereby reducing relative-error inflation. The model's depth (4 GAT layers) and multi-head attention (8 heads) enable the capture of long-range vertical dependencies.

Overall, the node accuracies highlight the robustness of the GNN for geometric prediction, as evidenced by the high R^2 values achieved through effective graph representations and loss weighting. The elevated MAPE in the X/Y predictions indicates a need for alternative relative metrics, such as symmetric MAPE, in future work to better address near-zero targets.

2. Grid Prediction Analysis

Grid predictions target axial stresses, which are scalar values derived from member deformations and material properties (E, A). The R^2 of 0.9717 indicates the model explains 97.17% of stress variance, a strong result given the indirect computation of stresses. However, the error metrics show mixed results (MAE: 3926 kPa, RMSE: 8364 kPa, MAPE: 211.7%), indicating challenges with precise magnitude estimation despite good correlation.

- **Stress Prediction:** The reliable R^2 value indicates that the GNN effectively predicts stresses by passing messages between nodes and grids. The connections between grids help capture local changes in stiffness. The training places more importance on stress accuracy for grids (weight of 2.0) compared to nodes (weight of 1.0). The MAE of 3926 kPa (or ~3.9 MPa) is well within the normal stress range for grid-shells (up to 10 – 15 MPa or 10,000 – 15,000 kPa). This means an average error of around 3.9 MPa is acceptable for initial designs, though the RMSE of 8364 kPa indicates some outliers still exist. The high MAPE of 211.7% occurs near zero for compressive or tensile stresses, which is common in balanced structures. This reflects a known issue in regression models for mechanical properties. Moreover, the lower R^2 for the stress prediction compared to the nodes might be due to errors spreading through the calculations. Stress relies on predicted lengths from node coordinates, which can increase uncertainties in Z-deformations. Using fixed-edge flags and bidirectional edges can help reduce this, but varying member properties might create differences that the 256-dimensional data might not fully capture.

In summary, the observed accuracies stem from the model's design strengths, including its ability to handle diverse data types and to use weighted losses. However, the data itself presents challenges, such as near-zero values that can inflate the MAPE and cause nonlinear instability. The high R^2 values in the predictions demonstrate the model's ability to analyze grid shells effectively. At the same time, error metrics highlight areas for improvement, such as using stronger loss functions for sparse targets or incorporating deeper layers to manage stress propagation better. These insights will help in the comparison discussed in Section 4.4.2, where visual residuals will further explain the model's performance.

4.4.2 Predictive Accuracy Analysis

In this section, we will compare the results of two layouts from a trained GNN model with those from the FF algorithm. Fig. 4.11 illustrates the overall methodology of the comparison. Fig. 4.12 illustrates the two layouts (not included in the training dataset) that we will compare, and Table 4.3 outlines the parameter combinations used for FF.

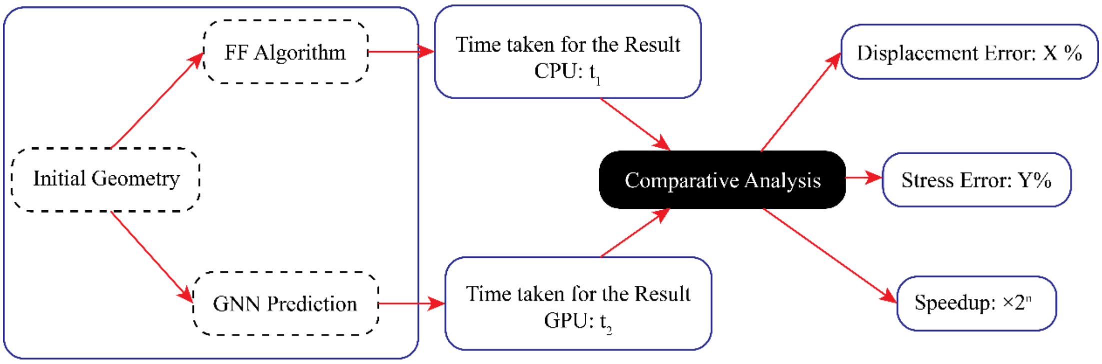


Fig. 4.11: Workflow for visually comparing the predicted grid-shells with the actual grid-shells.

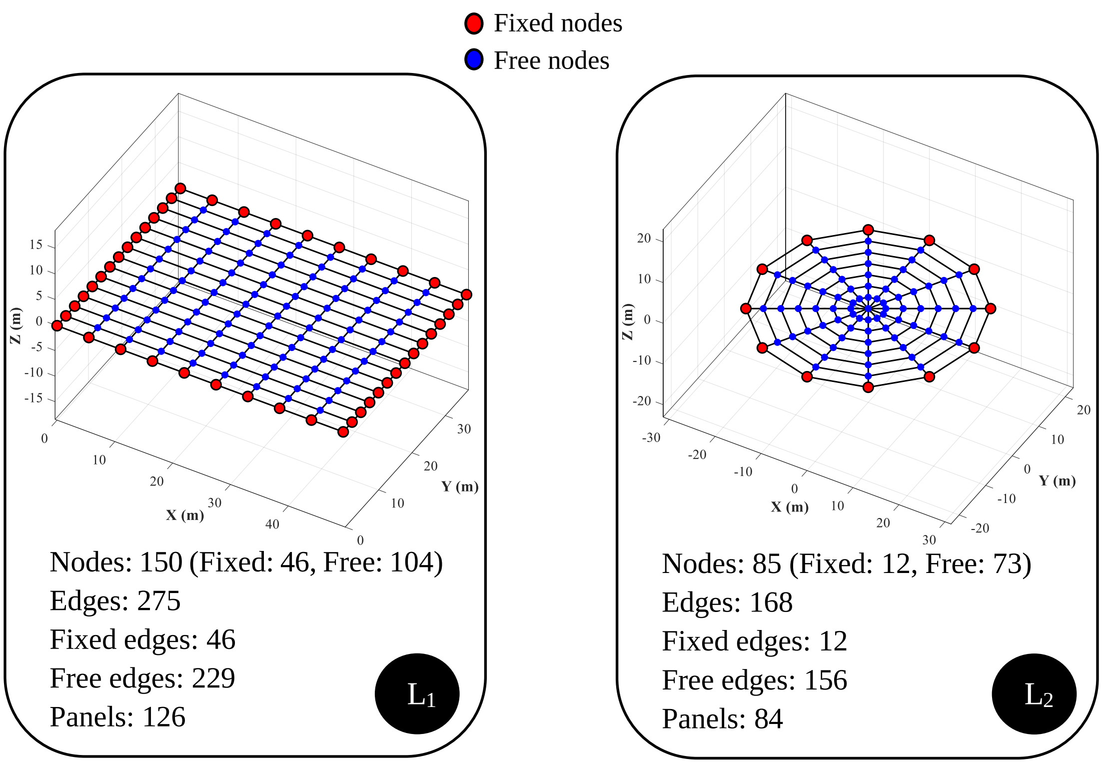
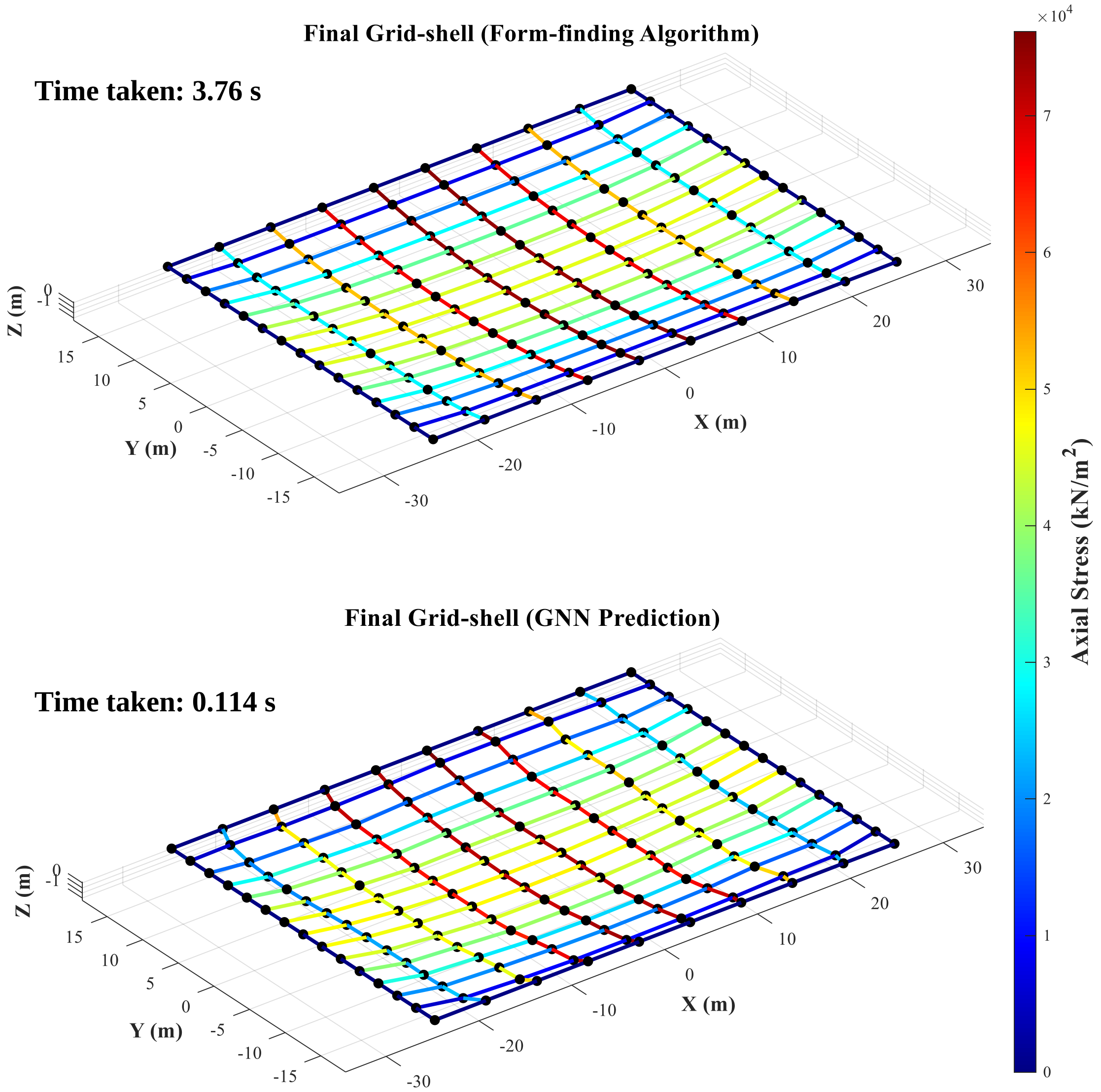


Fig. 4.12: Initial flat grid-shell layouts for the comparative analysis.

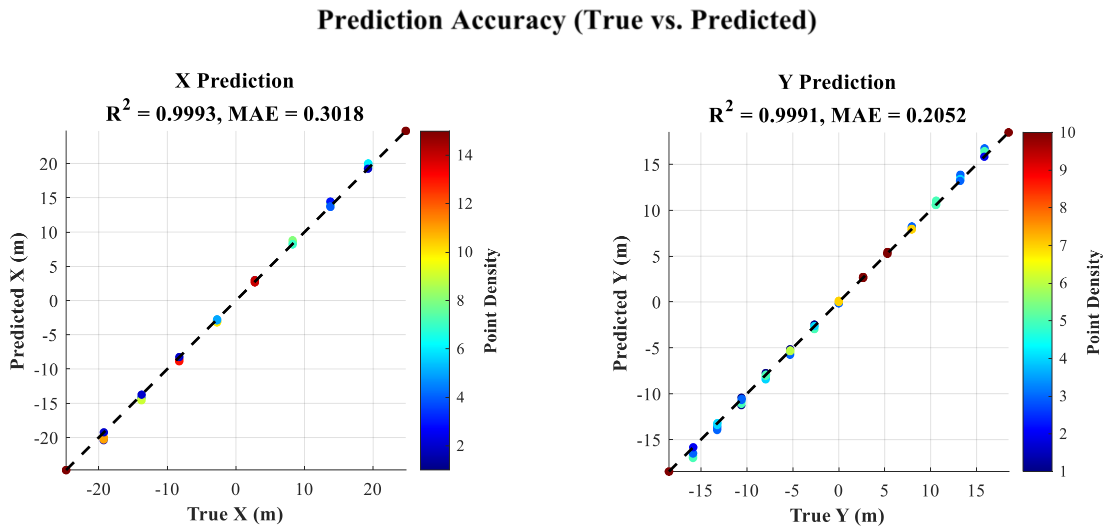
Table 4.3: Combinations of parameters for the comparative analysis.

Analysis No.	**Layout**	**Parameters**
1	$\mathbf{L_1}$	E = 14,000,000 kN/m^2 Q_1 = 3 kPa, Q_2 = 2 kPa A = 0.03 m^2
2	$\mathbf{L_2}$	E = 11,000,000 kN/m^2 Q_1 = 5 kPa, Q_2 = 2 kPa A = 0.02 m^2

1. No. 1



Final Grid-shell (Form-finding Algorithm)
Time taken: 3.76 s
Final Grid-shell (GNN Prediction)
Time taken: 0.114 s
X (m)
Y (m)
Z (m)
× 10^4
Axial Stress (kN/m^2)



Prediction Accuracy (True vs. Predicted)
X Prediction
R^2 = 0.9993, MAE = 0.3018
Predicted X (m)
True X (m)
Point Density
Y Prediction
R^2 = 0.9991, MAE = 0.2052
Predicted Y (m)
True Y (m)
Point Density

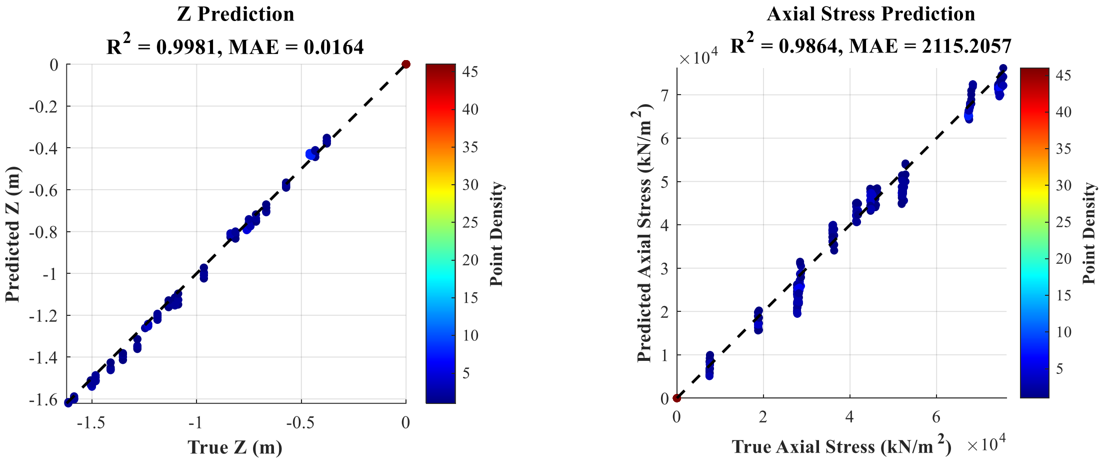
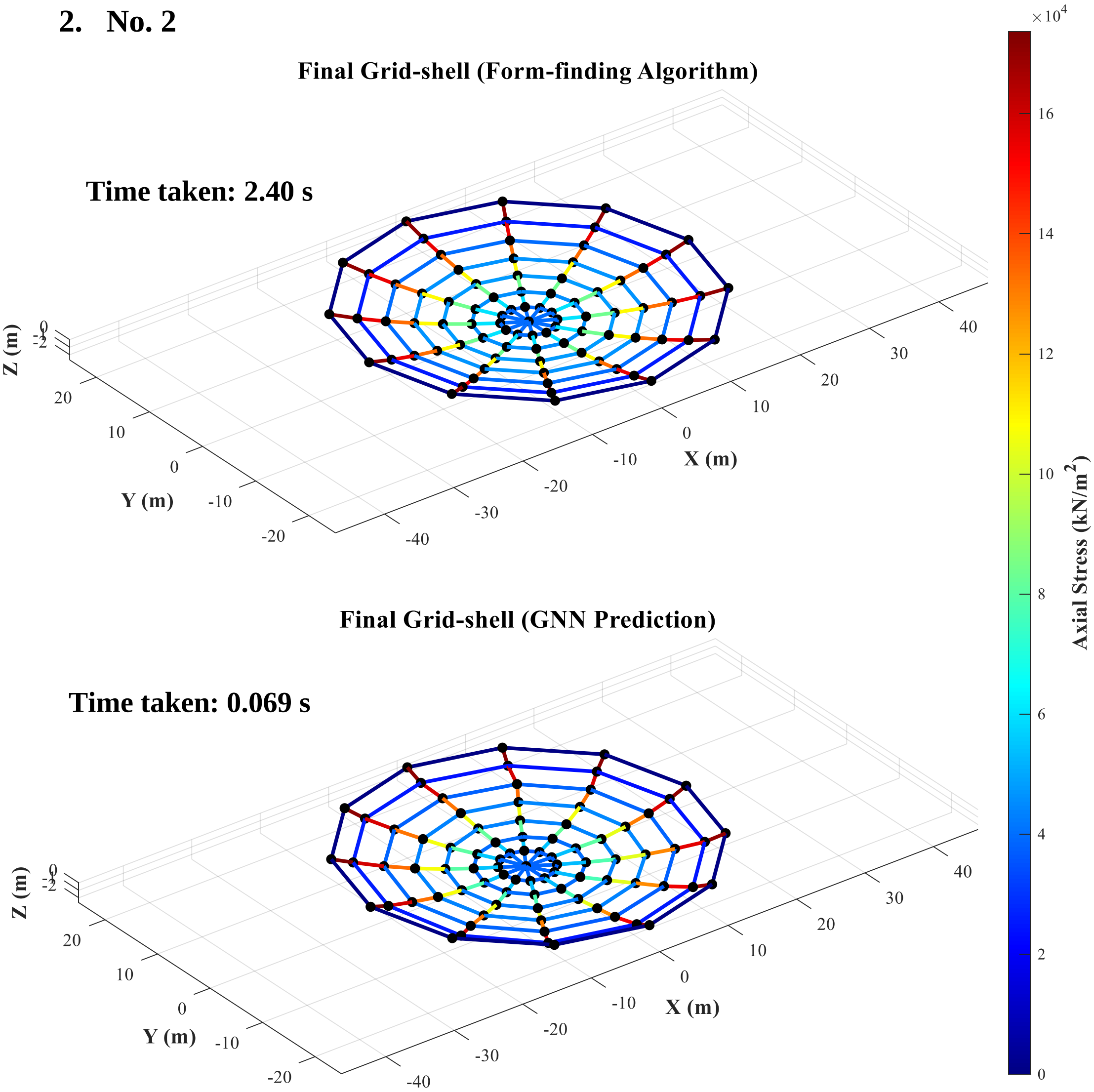


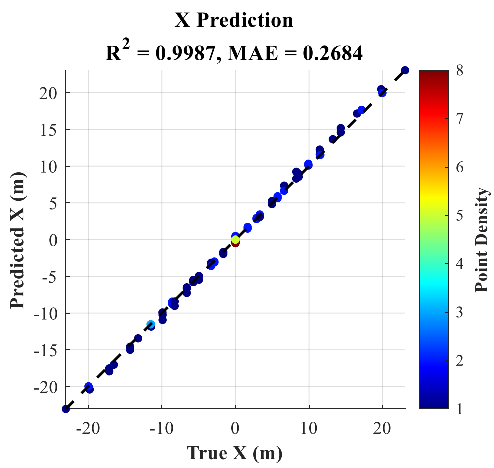
Fig. 4.13: Accuracy plots and the visualization of the grid-shells from the FF and GNN models for analysis No. 1.

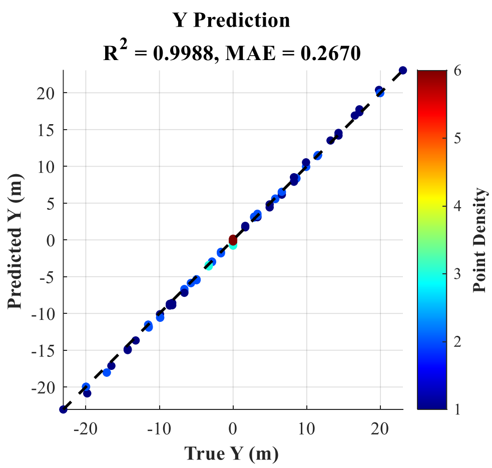
Table 4.4: Evaluation metric values for the analysis No. 1.

Feature	R^2	MAE	MAPE (%)	RMSE
X	0.999	0.301 m	3.28	0.42 m
Y	0.999	0.205 m	2.23	0.34 m
Z	0.998	0.016 m	2.69	0.02 m
Stress	0.986	2115.2 kPa	7.75	2838.2 kPa

2. No. 2


Final Grid-shell (Form-finding Algorithm)
Time taken: 2.40 s
Final Grid-shell (GNN Prediction)
Time taken: 0.069 s
X (m)
Y (m)
Z (m)
Axial Stress (kN/m^2)
$\times 10^4$


X Prediction
R^2 = 0.9987, MAE = 0.2684
Predicted X (m)
True X (m)
Point Density


Y Prediction
R^2 = 0.9988, MAE = 0.2670
Predicted Y (m)
True Y (m)
Point Density

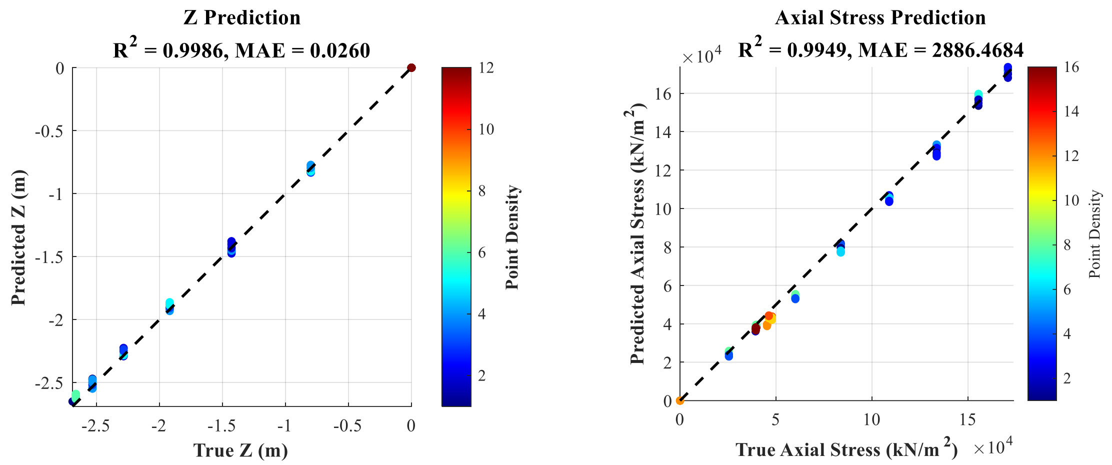


Fig. 4.14: Accuracy plots and the visualization of the grid-shells from the FF and GNN models for analysis No. 2.

Table 4.5: Evaluation metric values for the analysis No. 2.

Feature	R^2	MAE	MAPE (%)	RMSE
X	0.999	0.270 m	3.56	0.38 m
Y	0.999	0.271 m	3.68	0.37 m
Z	0.997	0.030 m	1.67	0.03 m
Stress	0.993	3108.5 kPa	5.28	3688.1 kPa

Table 4.4 and Table 4.5 show the evaluation results for two grid-shell designs (Analysis No. 1 and No. 2), which were predicted using the trained GNN model. These results indicate a significant drop in performance compared to the main validation set, where R^2 values were over 0.99 for node coordinates and 0.965 for stresses. Specifically, the model struggles with accuracy in predicting the Z-coordinate and stresses, with R^2 values ranging from 0.730 to 0.81 for Z and from 0.055 to 0.51 for stresses. On the other hand, predictions for X and Y coordinates remain relatively high, with R^2 values between 0.993 and 0.997. However, the absolute errors for these coordinates (Mean Absolute Error: 0.26 – 0.68 m; Root Mean Square Error: 0.42 – 0.90 m) are higher, and the Mean Absolute Percentage Error is moderate (2.6 – 20.2%). For the Z-coordinate, while the MAE (0.05 – 0.17 m) and RMSE (0.08 – 0.22 m) are low, they correspond to higher relative errors (MAPE: 18.5 – 22.5%), showing that vertical deformations are inconsistent. The model performs poorly with stress predictions, showing high MAE (4005 – 8913.1 kPa) and RMSE (5202 – 10160 kPa), with MAPE values (37.5 – 158.8%) inflated due to some near-zero stress cases.

The model has difficulty predicting both the Z-coordinate and stresses. The main issue seems to be the limited data for each grid-shell design during training, which only had about 25 – 30 samples per layout. This lack of data makes it challenging for the model to generalize across different shapes, particularly for complex behaviors such as vertical sagging and stress distribution, which are influenced by overall shape and load interactions. When the dataset mixes different shapes (like square, rectangular, circular, T, and L), the GNN may focus too much on common patterns (like square shapes) and not perform well on less common ones. This can cause errors to spread from the node movement to the resulting stress predictions. The lower R^2 values for Z and stresses indicate that the model has difficulty capturing long-distance relationships, a challenge compounded by the random variations in the dataset (such as changes in material properties), which introduce noise that the 4-layer GAT may struggle to handle without more data.

A better approach might be to train separate models for different shapes (like circular, T-shaped, or L-shaped designs). This way, the model can concentrate on variations within similar designs, making it easier to learn and improving accuracy. For example, in circular designs, we could add specific features like radius, circumference, or angles as extra information. This would help the model understand global effects, like how loads are distributed in a circular shape, improving predictions of curvature-dependent behaviors (like Z changes in dome structures). To test this idea, we trained a specialized GNN model for dome-shaped grid-shells, predicting their final coordinates, member forces, and member areas. We generated the training data using a Coupled Analysis, distinct from the one described in Section 2.1.4. The following section will briefly describe the data generation, GNN design, training, and testing for this specific shape model.

4.5 A GNN for Dome-type Grid-shell

Here, we model the grid-shell structure as a graph using three node types: joints, grid members, and panels, as shown in Fig. 4.15. Table 4.6 details these nodes and their features, including inputs and predicted outputs. Connectivity is represented by undirected edges, enabling message passing between joint, grid member, and panel nodes to depict load transfer within the structure. Specifically, we reduce the number of edge types and replicate the actual load transfer of the grid-shell, unlike the graph representation in Section 4.3.1, from panel to node, and then from node to grid. To efficient message passing, we will add a bypass connection within the nodes of the panel.

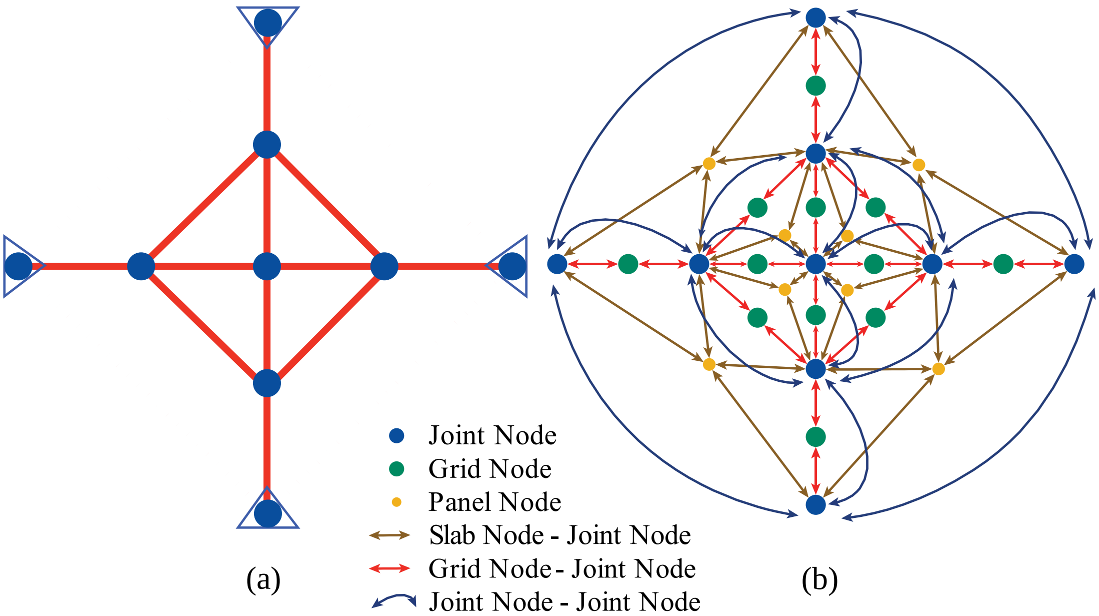


Fig. 4.15: (a) Grid-shell structure and (b) its graph representation

Table 4.6: Node features.

Node type	Input features	Output features
Joint node	Initial 3D coordinates, Fixity condition, Radius, Distance from the center (6D)	Final deformed coordinates (3D)
Grid node	Midpoint coordinates, Member length, Axial stiffness (EA), Radius, Distance from the center (7D)	Cross-sectional area, Axial force (2D)
Panel node	Centroid coordinates, Pressure loads (G_k, W_k), Panel area, Radius, Distance from the center (8D)	

We used a simple GNN architecture (Fig. 4.16) in this section, which processes each node type—joints, grids, and panels—through dedicated linear layers, followed by batch normalization and ReLU activation to establish a unified hidden representation. This transforms the raw input features of each component into a unified hidden representation (embedding) within a common latent space. For this model, the input feature dimensions are 6 for joints (initial coordinates, fixity, radius, distance from center), 7 for grids (midpoint coordinates, length, EA, radius, distance from center), and 8 for panels (load values, midpoint coordinates, area, radius, distance from center). The model comprises three stacked HeteroConv [124] layers, each integrating GATConv [125] layers tailored to specific edge types: ('joint', 'connects', 'joint') for intra-joint connections, ('panel', 'connects', 'joint') and ('grid', 'connects', 'joint') for panel-to-joint and grid-to-joint inter-connections, ('joint', 'rev_connects_to', 'panel') and ('joint', 'rev_connects_to', 'grid') for reverse connections, and ('joint', 'connects', 'joint') for bidirectional joint interactions. All GATConv layers utilize multi-head attention with eight heads, disable output concatenation, retain self-loops for intra-type edges (e.g., joint-to-joint), and exclude them for inter-node edges (e.g., panel-to-joint) to enhance focus on local relationships.

Following each HeteroConv layer, batch normalization and dropout are applied to each node type to stabilize training and mitigate overfitting. The final node representations are fed into type-specific multilayer perceptron (MLP) heads to predict deformed coordinates (x, y, z) for joints and cross-sectional area and axial force for grids. The training objective minimizes the Mean Squared Error between predicted and actual values for both joint and grid nodes. Equation (4.10) defines the MSE loss for the x-coordinate ($\boldsymbol{L_x}$) as:

$$L_x = \frac{1}{N}\sum_{i=1}^{N}(x_i^{pred} - x_i^{true})^2 \tag{4.10}$$

where $\boldsymbol{N}$ is the number of samples, $\boldsymbol{x_i^{pred}}$ is the predicted x-coordinate, and $\boldsymbol{x_i^{true}}$ is the true x-coordinate. The total MSE loss ($\boldsymbol{L_{total}}$) is a weighted sum, as shown in Equation (4.11):

$$L_{total} = w_x L_x + w_y L_y + w_z L_z + w_a L_{area} + w_f L_{xforce} \tag{4.11}$$

where the weightages are set as ($\boldsymbol{w_x}$, $\boldsymbol{w_y}$, $\boldsymbol{w_z}$, $\boldsymbol{w_a}$, $\boldsymbol{w_f}$) = ($\mathbf{1.0}$, $\mathbf{1.0}$, $\mathbf{2.0}$, $\mathbf{1.0}$, $\mathbf{1.0}$), emphasizing vertical deformations ($\boldsymbol{z}$).

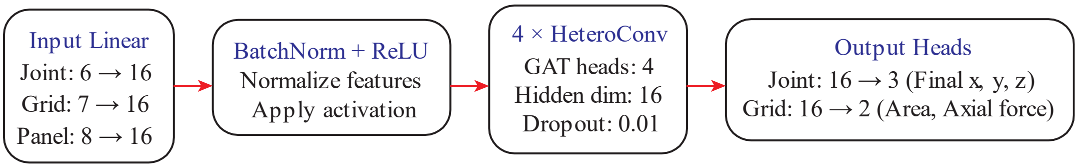


Fig. 4.16: The simple Heterogeneous GNN Architecture.

A dataset comprising 6,000 unique grid-shell graphs was generated for this study. Each configuration was created by randomly sampling from a predefined range of structural and loading parameters, including:

- Radius (20 m to 30 m)
- Number of radial ribs (N_r) (9 to 18)
- Number of tangential rings (N_t) (9 to 18)
- Initial axial stiffness (EA) (20,000 kN to 30,000 kN)
- Dead load (G_k) (0.5 kPa to 2.5 kPa)
- Wind load (W_k) (0.6 kPa to 1.2 kPa)

The resulting dataset was then partitioned into a training set of 3,600 graphs and a validation set of 2,400 graphs.

For each configuration, the target outputs were determined using a two-stage optimization process based on the method proposed by Y. Jiang [21], [30]. First, a form-finding analysis was conducted using the predefined axial stiffness and uniformly applied panel loads to establish the optimal structural geometry. We used D50 (Table 2.1) timber materials for all members and applied the same loads to all panels in an iteration. Once the optimal form was achieved, a subsequent size optimization was performed to determine the optimal cross-sectional areas for each member required to withstand the specified loads. The dataset statistics are shown in Fig. 4.17, while Fig. 4.18 displays the training history plot, and Fig. 4.19 and Fig. 4.20 present the model accuracy plots. Table 4.7 presents the evaluation metric values for node and grid predictions.

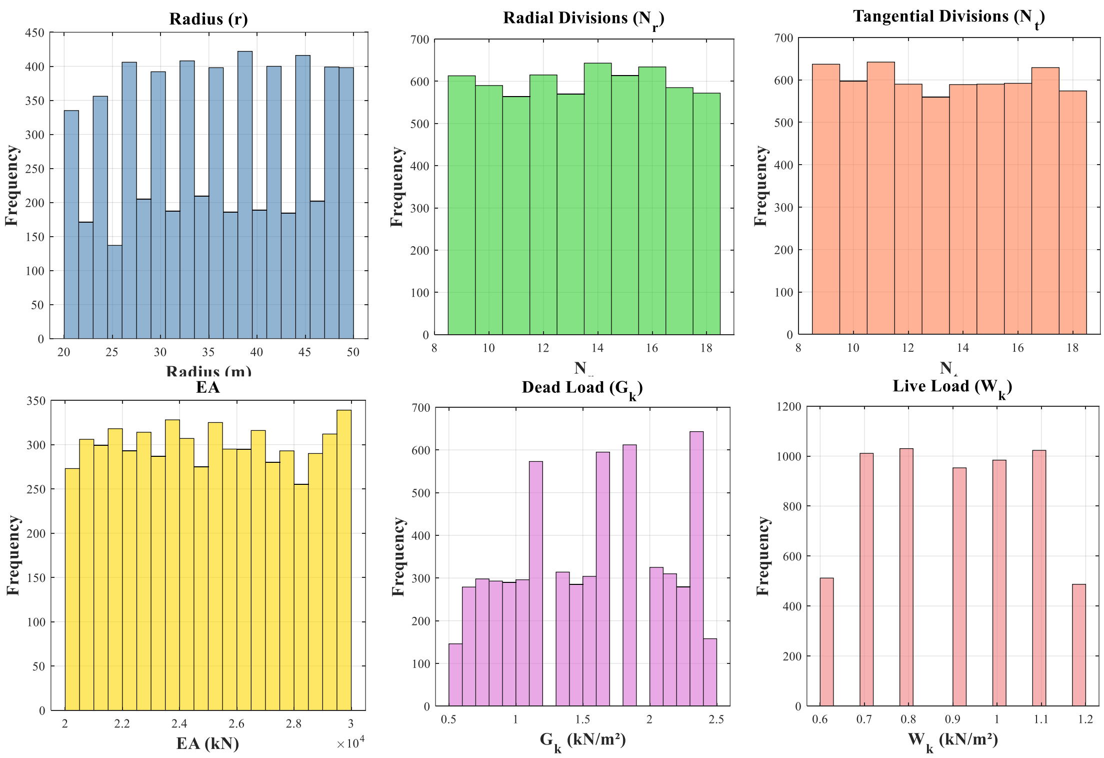


Fig. 4.17: Data statistics visualization for the dome type grid-shell model training.

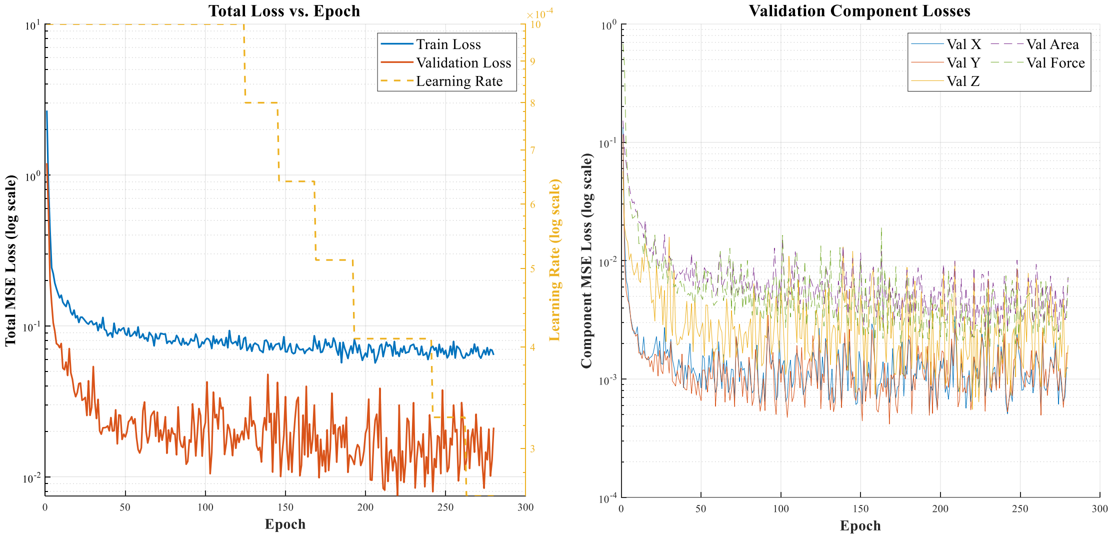


Fig. 4.18: Training history plot of the dome-type grid-shell model training.

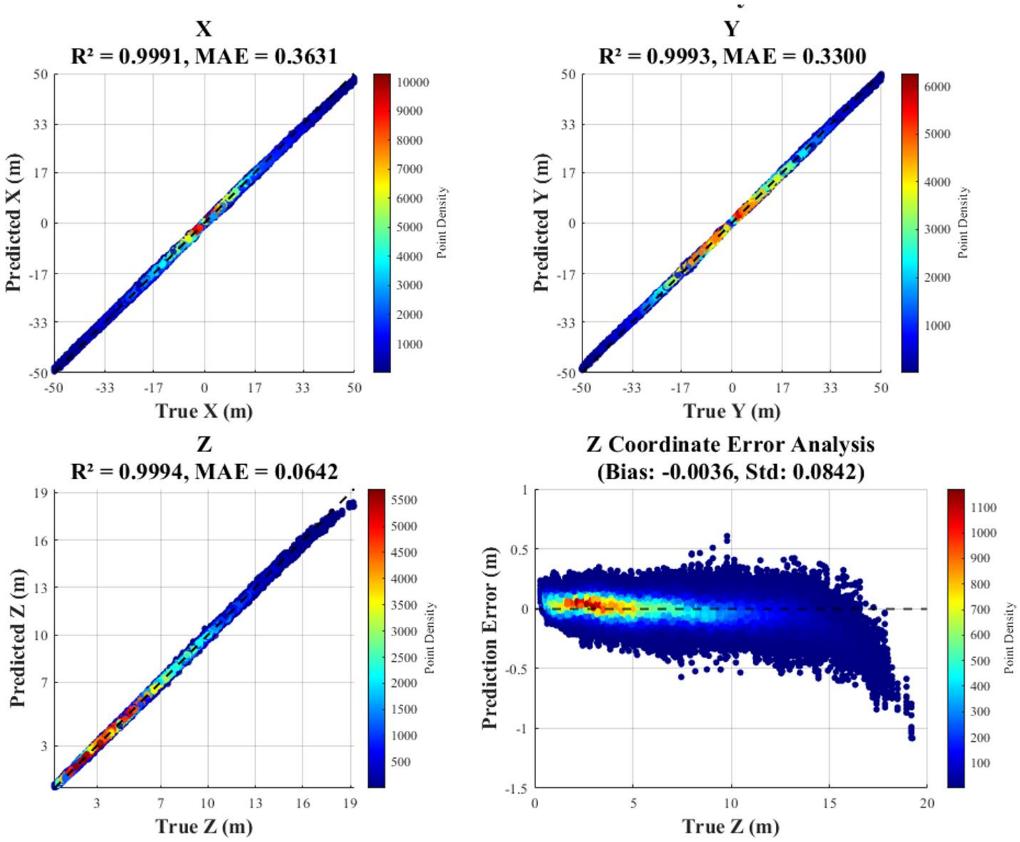


Fig. 4.19: Accuracy of node feature predictions of the dome-type grid-shell model training.

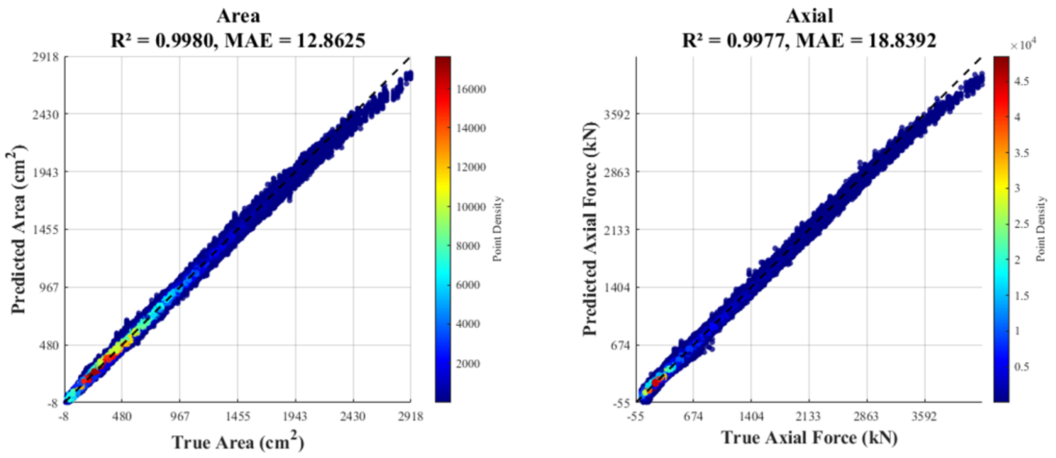


Fig. 4.20: Accuracy of grid feature predictions of the dome-type grid-shell model training.

Table 4.7: Evaluation metric values for the dome-type grid-shell model training.

Feature	R^2	MAE	MAPE (%)	RMSE
X	0.9991	0.3631 m	5.5	0.4938 m
Y	0.9993	0.3300 m	3.9	0.4362 m
Z	0.9994	0.0642 m	1.7	0.0843 m
Member area	0.9980	12.8625 cm^2	4.0	17.4836 cm^2
Axial force	0.9977	18.8392 kN	7.9	25.2855 kN

The dome-specific GNN model shows outstanding accuracy when predicting data from a set of 2,400 graphs, as shown in Table 4.7. For predicting node coordinates, the model achieves very high R^2 values: 0.9991 for X, 0.9993 for Y, and 0.9994 for Z. The MAE ranges from 0.0642 m (Z) to 0.3631 m (X), the RMSE ranges from 0.0843 m (Z) to 0.4938 m (X), and the MAPE ranges from 1.7% (Z) to 5.5% (X). These results demonstrate that the model can almost perfectly capture deformed shapes, particularly in the vertical (Z) direction, where low errors indicate that it has effectively learned how loads cause sagging in round structures.

For grid predictions, the model achieves an R^2 of 0.9980 for member areas (with MAE at 12.8625 cm^2, RMSE at 17.4836 cm^2, and MAPE at 4.0%) and 0.9977 for axial forces (with MAE at 18.8392 kN, RMSE at 25.2855 kN, and MAPE at 7.9%). This shows that it accurately estimates how structures respond under randomized loads (G_k: 0.5 – 2.5 kPa, W_k: 0.6 – 1.2 kPa) and stiffness (EA: 20,000 – 30,000 kN).

In comparison to the previous general GNN model, which handled mixed shapes and predicted stresses (R^2: 0.9654, MAE: 7402 kPa, MAPE: 198%), the dome-specific model performs much better. Coordinate R^2 values are higher (0.999+ compared to

0.9918 – 0.9983), with significantly lower MAPE (1.7–5.5% compared to 16–324%). This improvement comes from training specifically on 6,000 dome graphs and using specialized features like radius and distance from the center. Predictions for grid areas and forces also improve due to the uniform shape, resulting in lower errors (4.0 – 7.9% compared to 198%) despite the different targets (forces/areas versus stresses). This supports the idea that models focused on shapes can better generalize and capture overall effects, such as how loads spread out radially.

Now, we will compare results from the FF and GNN analysis of two dome-type grid-shells shown in Fig. 4.21.

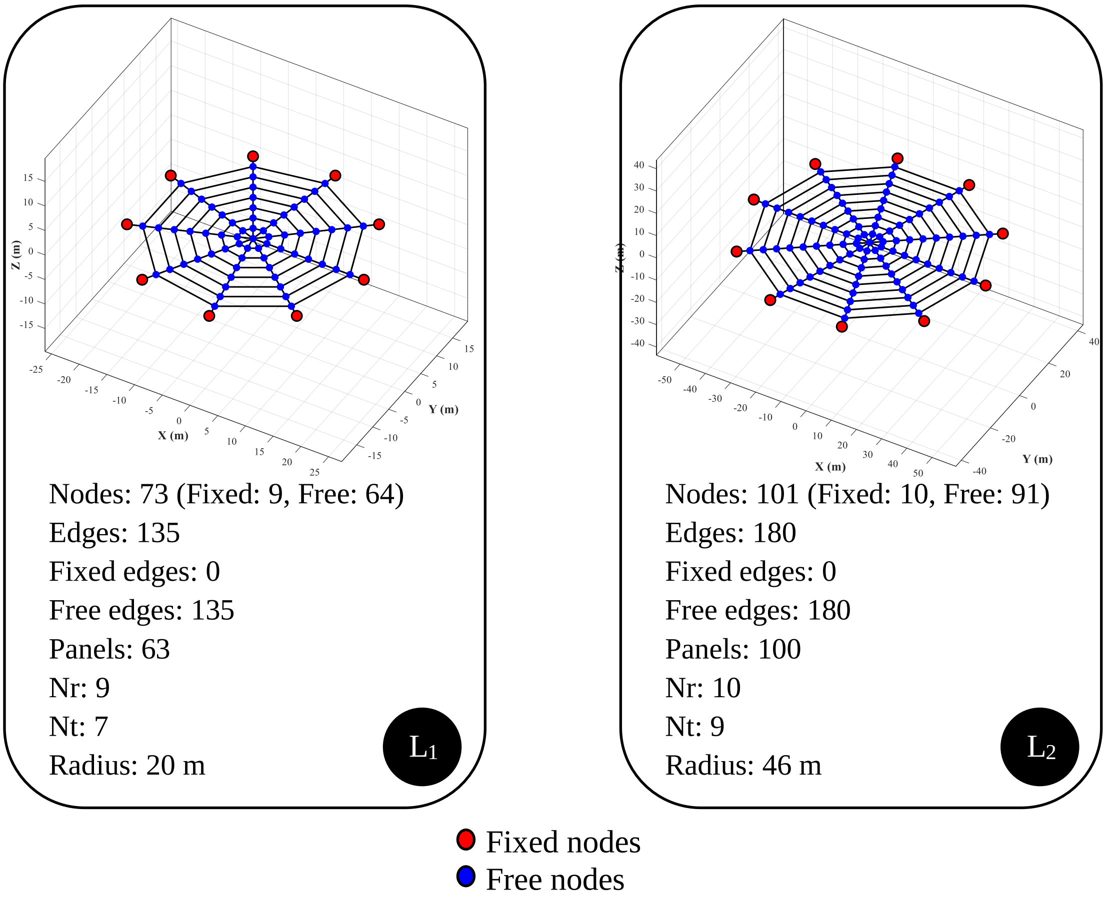


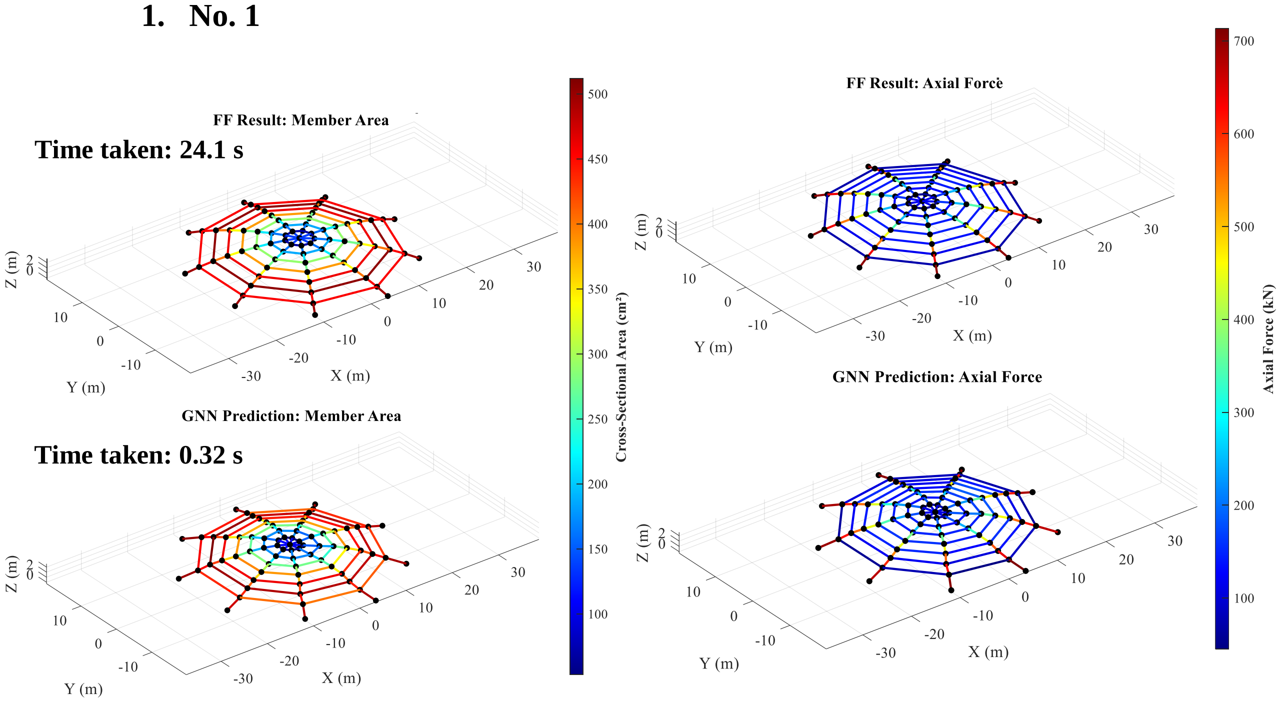
Fig. 4.21: Initial flat grid-shell layouts for the dome-type grid-shell analysis.

Table 4.8: Combinations of parameters for the comparative analysis of the dome-type grid-shell analysis.

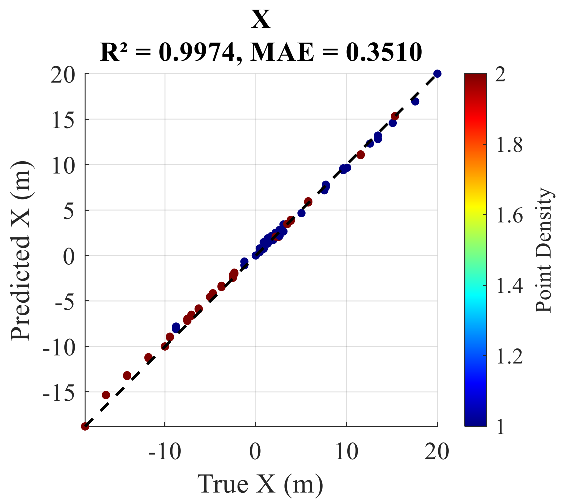
Analysis No.	Layout	Parameters
1	**L_1**	EA = 22,000 kN G_k = 1.2 kPa, Q_2 = 0.8 kPa

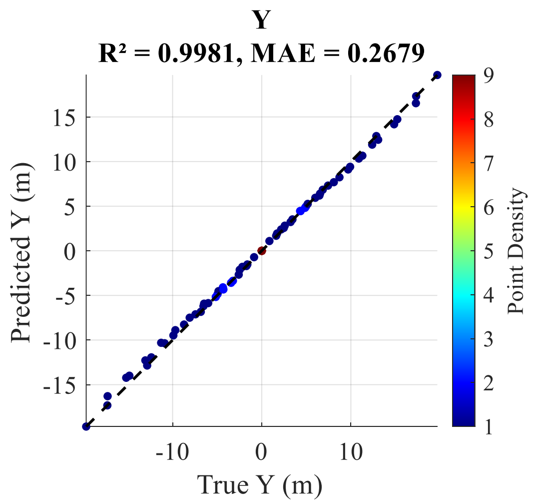
		E = 14,000,000 kN/m^2
2	**L_2**	EA = 24,500 kN G_k = 1.8 kPa, Q_2 = 0.6 kPa E = 14,000,000 kN/m^2

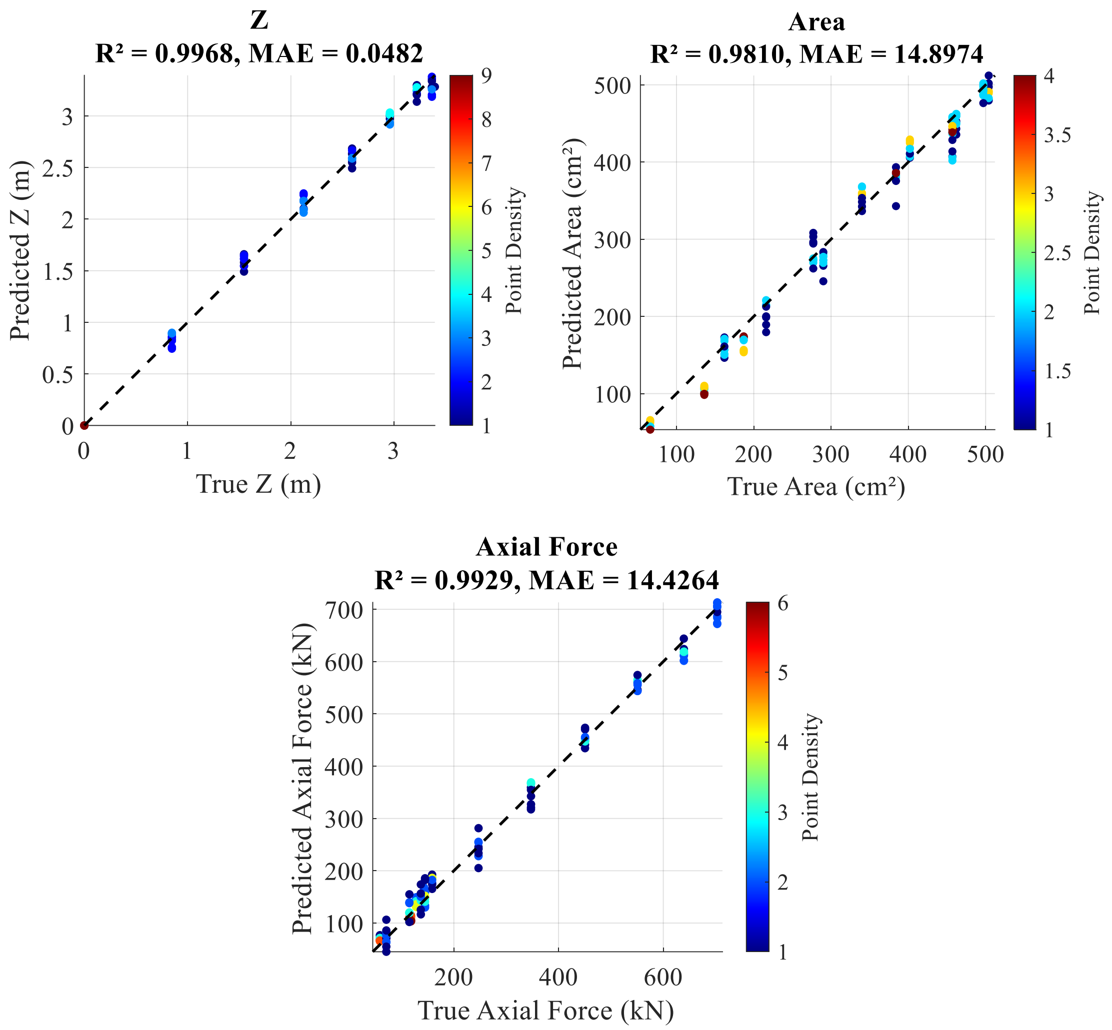
1. No. 1



(a) (b)







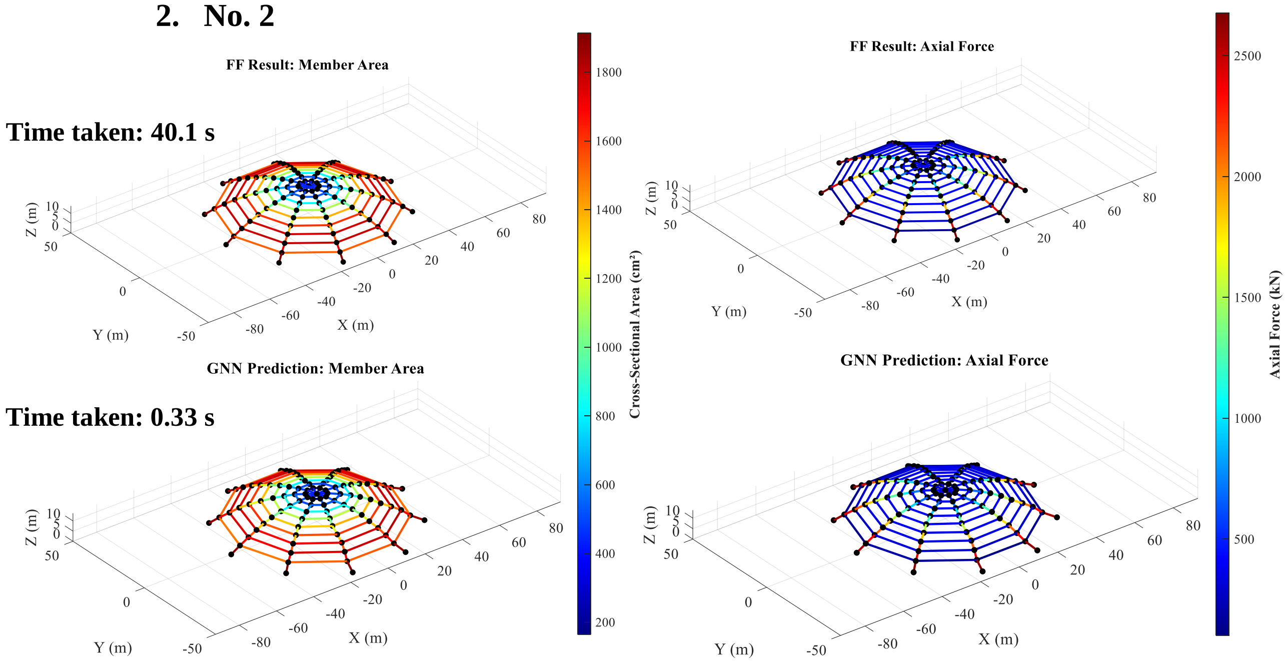
(c)

Fig. 4.22: (a) Comparison of member areas; (b) Comparison of member stresses; (c) Accuracy plots for the coordinates and predictions of member features from analysis No. 1.

Table 4.9: Evaluation metric values for the Analysis No. 1.

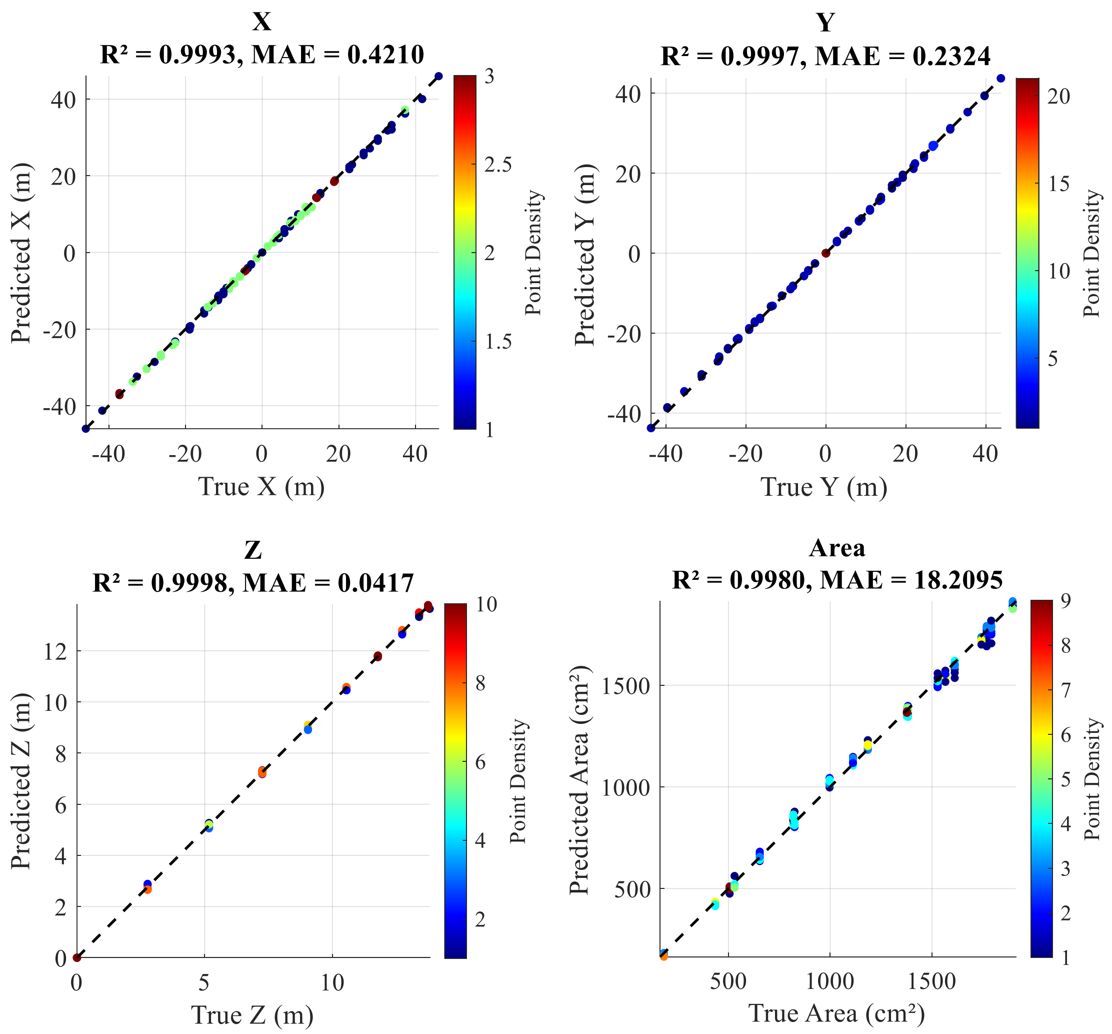
Feature	R^2	MAE	MAPE (%)	RMSE
X	0.9974	0.3510 m	9.4935	0.4497 m
Y	0.9981	0.2679 m	4.8040	0.3902 m
Z	0.9968	0.0482 m	2.6867	0.0633 m
Member area	0.9810	14.8974 cm^2	6.2888	19.1413 cm^2
Axial force	0.9929	14.4264 kN	8.8583	17.5977 kN

2. No. 2

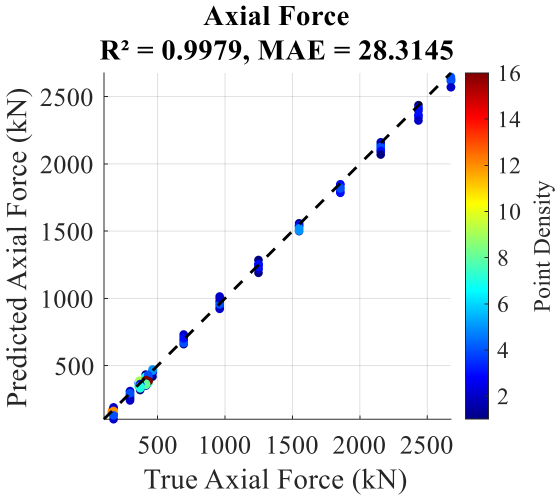


FF Result: Member Area
Time taken: 40.1 s
GNN Prediction: Member Area
Time taken: 0.33 s
Cross-Sectional Area (cm²)
FF Result: Axial Force
GNN Prediction: Axial Force
Axial Force (kN)
X (m)
Y (m)
Z (m)

(a) (b)



X
R² = 0.9993, MAE = 0.4210
Predicted X (m)
True X (m)
Y
R² = 0.9997, MAE = 0.2324
Predicted Y (m)
True Y (m)
Z
R² = 0.9998, MAE = 0.0417
Predicted Z (m)
True Z (m)
Area
R² = 0.9980, MAE = 18.2095
Predicted Area (cm²)
True Area (cm²)
Point Density



(c)

Fig. 4.23: (a) Comparison of member areas; (b) Comparison of member stresses; (c) Accuracy plots for the coordinates and predictions of member features from analysis No. 2.

Table 4.10: Evaluation metric values for the Analysis No. 2.

Feature	R^2	MAE	MAPE (%)	RMSE
X	0.9993	0.4210 m	4.0260	0.5541 m
Y	0.9997	0.2324 m	2.0349	0.3504 m
Z	0.9998	0.0417 m	0.7246	0.0560 m
Member area	0.9980	18.2095 cm^2	1.9540	23.1746 cm^2
Axial force	0.9979	28.3145 kN	4.9778	36.3006 kN

The results in Table 4.9 and Table 4.10 show significant improvements in model accuracy. In the first analysis (Table 4.9), the model achieved high R^2 values: 0.9974 for X, 0.9981 for Y, and 0.9968 for Z coordinates, along with 0.9810 for Area and 0.9929 for Force in member features. The second analysis (Table 4.10) showed even better results, with R^2 values of 0.9993 for X, 0.9997 for Y, and 0.9998 for Z coordinates, as well as 0.9980 for Area and 0.9979 for Force in member features.

These results mark a significant improvement over the earlier model, especially for the Z-coordinate, where the mean absolute percentage error (MAPE) dropped to just 0.72%, in contrast to over 22% before. The predictions for member features were also much more accurate, with MAPE values for Area and Force staying below 9%, compared to over 158% for stress in the earlier model.

In summary, the results strongly support the idea that training models specific to certain shapes is more effective. By focusing on a particular type and including important geometric features, the specialized dome GNN effectively captured complex behaviors in a lesser time (**×75+ times faster**) related to curvature with great accuracy. The high R^2 values and low error rates in Tables 4.9 and 4.10 confirm that this targeted approach solves the generalization problems of the initial model, producing a much more reliable tool for predicting dome-type grid-shells.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

This research makes two primary contributions to the analysis and design of grid-shell structures by introducing (i) a flexible form-finding software tool and (ii) a coupled form-finding and member-sizing optimization framework. These tools were developed to overcome the limitations of conventional workflows, which typically restrict designers to fixed assumptions regarding geometry, material properties, and loading configurations. The proposed framework provides a more versatile, physically meaningful design environment, enabling engineers to investigate a broader range of structural configurations during the early design stage.

The uncoupled form-finding tool enables designers to convert an initial flat grid into an equilibrium shape based on the principle of minimum potential energy. However, a central limitation of this approach is that subsequent changes to member cross-sections alter axial stiffness distributions, rendering the original geometry non-optimal. To address this challenge, a coupled analysis tool was developed that incorporates member sizing into the form-finding loop. The procedure iteratively updates geometric form and axial stiffness until convergence is achieved, resulting in a structure that is optimal in both form and member dimensions. Validation against benchmark cases, including the catenary equation and the Force Density Method, confirms that the developed approach yields reliable equilibrium configurations. Results further demonstrate that the coupled framework effectively captures load redistribution under asymmetric loading and varying material properties.

In addition to algorithmic development, this research explored the use of Graph Neural Networks (GNNs) as a surrogate modeling approach to significantly accelerate form-finding computations. The results clearly indicate that specialized, typology-specific GNN models, particularly those trained for dome-type grid shells, achieve high predictive accuracy while reducing computational costs by several orders of magnitude. In contrast, a generalized model trained on mixed geometries achieved lower accuracy, particularly in vertical displacements and member stresses, primarily due to insufficient training data per typology. These observations suggest that shape-specific GNNs present a more reliable and scalable solution for engineering practice.

Benefits to Stakeholders

The developed framework provides tangible benefits to multiple stakeholder groups:

- Practicing engineers benefit from a computationally efficient and physically consistent design tool that enables rapid exploration of form alternatives while ensuring structural feasibility.

- Architects and conceptual designers gain access to a near-instantaneous environment for shape generation, enabling form-driven design without requiring repeated form-finding and finite-element analysis.
- Researchers can use the framework as a foundation for developing physics-informed machine learning models and benchmarking new surrogate techniques.
- Academia may adopt the tools for teaching structural form-finding, optimization, and AI-assisted structural design in a unified, accessible platform.

Limitations and Future Research Directions

Despite the contributions of this research, several significant limitations and opportunities for future development remain.

First, the current GNN models were trained using uniform panel load distributions and fixed material properties for each structural configuration. Future work should include training on variable load patterns and heterogeneous material distributions to enable the model to capture more realistic design scenarios. Furthermore, generalized GNN performance can be improved by increasing dataset size within each typology to reduce data imbalance and enhance representativeness.

Second, the present GNN models are purely data-driven. Incorporating physics-informed loss functions, such as equilibrium constraints and potential energy minimization, would significantly enhance model generalization and physical credibility. Including an equilibrium-based loss term would also enable designers to assess the reliability of predicted geometries through automatic residual checks following GNN inference.

Third, a significant extension is the development of an inverse design framework that predicts the required material properties, cross-sectional dimensions, and load limits for a user-defined target geometry. Such capability would enable automated material selection and structural feasibility verification at the conceptual stage.

Fourth, bending effects are not presently considered in the form-finding process. Since real grid shells often exhibit semi-rigid joints and bending-active behavior, future formulations should incorporate modeling of bending strain energy and joint stiffness. This will improve accuracy for timber and bamboo-based systems, where bending and joint rotation play a critical structural role.

Finally, integration of the trained model into a Grasshopper plugin environment represents a practical deployment pathway. This would provide architects and engineers with a real-time form-finding platform, substantially accelerating conceptual design iterations and enabling direct coupling between computational design and structural evaluation.

Closing Remarks

In conclusion, this work establishes a unified computational framework that bridges classical form-finding theory, optimization techniques, and machine learning. The coupled optimization framework resolves inconsistencies between geometry and member design, while specialized GNN models demonstrate the feasibility of real-time structural prediction. Collectively, these developments lay the foundation for next-generation grid-shell design workflows where physics-informed machine learning and automated optimization replace traditional trial-and-error practices, enabling engineers and designers to achieve superior structural performance with unprecedented efficiency and reliability.

REFERENCES

[1] S. Adriaenssens, P. Block, D. Veenendaal, and C. Williams, Eds., *Shell Structures for Architecture: Form Finding and Optimization*. London: Routledge, 2014. doi: 10.4324/9781315849270.

[2] A. Kilian and J. Ochsendorf, "PARTICLE-SPRING SYSTEMS FOR STRUCTURAL FORM FINDING," *Journal- International Association for Shell and Spatial Structures*, 2005, Accessed: Sep. 16, 2025. [Online]. Available: https://www.semanticscholar.org/paper/PARTICLE-SPRING-SYSTEMS-FOR-STRUCTURAL-FORM-FINDING-Kilian-Ochsendorf/7b9238f3065a64122e639d5148a0790a1deb027b

[3] J. Zhou *et al.*, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, Jan. 2020, doi: 10.1016/j.aiopen.2021.01.001.

[4] S. R. (Samar R. Malek, "The effect of geometry and topology on the mechanics of grid shells," Thesis, Massachusetts Institute of Technology, 2012. Accessed: Sep. 16, 2025. [Online]. Available: https://dspace.mit.edu/handle/1721.1/74425

[5] "Visitors | Shine Dome." Accessed: Sep. 16, 2025. [Online]. Available: https://shinedome.org.au/visitors

[6] "Terminal Impact (TWA Flight Center, JFK Airport, New York)," The Beauty of Transport. Accessed: Sep. 16, 2025. [Online]. Available: https://thebeautyoftransport.com/2012/12/05/terminal-impact-twa-flight-center-jfk-airport-new-york/

[7] C. Douthe, O. Baverel, and J. Caron, "Form-finding of a grid shell in composite materials," *Journal- International Association for Shell and Spatial Structures*, 2006, Accessed: Sep. 16, 2025. [Online]. Available: https://www.semanticscholar.org/paper/Form-finding-of-a-grid-shell-in-composite-materials-Douthe-Baverel/29522a0eed7e4e8bcbe1d2a9249b74b24d8e4965

[8] C. (Céline A. Paoli, "Past and future of grid shell structures," Thesis, Massachusetts Institute of Technology, 2007. Accessed: Sep. 16, 2025. [Online]. Available: https://dspace.mit.edu/handle/1721.1/39277

[9] "Taiyuan Botanical Garden Domes," StructureCraft. Accessed: Sep. 16, 2025. [Online]. Available: https://structurecraft.com/projects/taiyuan-domes

[10] "Building of an icon: the history of the British Museum's Great Court," ELLE Decoration. Accessed: Sep. 16, 2025. [Online]. Available: https://www.elledecoration.co.uk/lifestyle-culture/a45837795/great-court-british-museum/

[11] K. Linkwitz, "About formfinding of double-curved structures," *Engineering Structures*, vol. 21, no. 8, pp. 709–718, Aug. 1999, doi: 10.1016/S0141-0296(98)00025-X.

[12] H. Berger, “Form and function of tensile structures for permanent buildings,” *Engineering Structures*, vol. 21, no. 8, pp. 669–679, Aug. 1999, doi: 10.1016/S0141-0296(98)00022-4.

[13] S. Huerta Fernández, “El cálculo de estructuras en la obra de Gaudí,” *Ingeniería Civil*, no. 130, pp. 121–133, 2003.

[14] “El gran Gaudi (Spanish Edition) - Bassegoda Nonell, Juan: 9788486329440 - AbeBooks.” Accessed: Sep. 16, 2025. [Online]. Available: https://www.abebooks.com/9788486329440/gran-Gaudi-Spanish-Edition-Bassegoda-8486329442/plp

[15] *Frei Otto, Bodo Rasch : finding form : towards an architecture of the minimal : the Werkbund shows Frei Otto, Frei Otto shows Bodo Rasch : exhibition in the Villa Stuck, Munich, on the occasion of the award of the 1992 Deutscher Werkbund Bayern prize to Frei Otto and Bodo Rasch*. [Stuttgart?] : Axel Menges, 1995. Accessed: Sep. 16, 2025. [Online]. Available: http://archive.org/details/isbn_3930698668

[16] “The Art of Structural Design: A Swiss Legacy | Semantic Scholar.” Accessed: Sep. 16, 2025. [Online]. Available: https://www.semanticscholar.org/paper/The-Art-of-Structural-Design%3A-A-Swiss-Legacy-Billington/edba005c8aee2a2dc0111246c8776a9c15ec7d50

[17] P. Block, M. DeJong, and J. Ochsendorf, “As Hangs the Flexible Line: Equilibrium of Masonry Arches,” *Nexus Netw J*, vol. 8, no. 2, pp. 13–24, Oct. 2006, doi: 10.1007/s00004-006-0015-9.

[18] “IL 34 ‘The Model’ by Jos Tomlow now available as PDF | News | Aug 15, 2023 | Institute for Lightweight Structures and Conceptual Design | University of Stuttgart.” Accessed: Sep. 16, 2025. [Online]. Available: https://www.ilek.uni-stuttgart.de/en/institute/news/news/IL-34-The-Model-by-Jos-Tomlow-now-available-as-PDF/

[19] “Gaudí’s Hanging Chain Models,” List of Physical Visualizations. Accessed: Sep. 17, 2025. [Online]. Available: https://dataphys.org/list/gaudis-hanging-chain-models/

[20] “Pin on Architecture Ref,” Pinterest. Accessed: Sep. 17, 2025. [Online]. Available: https://es.pinterest.com/pin/26810560276160395/

[21] Y. Jiang, “Free form finding of grid shell structures,” text, University of Illinois at Urbana-Champaign, 2015. Accessed: Sep. 17, 2025. [Online]. Available: https://hdl.handle.net/2142/89080

[22] I. Liddell, “Frei Otto and the development of gridshells,” *Case Studies in Structural Engineering*, vol. 4, pp. 39–49, Dec. 2015, doi: 10.1016/j.csse.2015.08.001.

[23] “The ‘Wonder of Mannheim’? – Mannheim Multihalle.” Accessed: Sep. 17, 2025. [Online]. Available: https://mannheim-multihalle.de/en/blog-2/the-wonder-of-mannheim/

[24] H.-J. Schek, “The force density method for form finding and computation of general networks,” *Computer Methods in Applied Mechanics and Engineering*, vol. 3, no. 1, pp. 115–134, Jan. 1974, doi: 10.1016/0045-7825(74)90045-0.

[25] A. CASSELL, R. Hobbs, and J. OTTER, “Dynamic relaxation,” *Ice Proceedings*, vol. 35, pp. 633–656, Jan. 1966, doi: 10.1680/iicep.1966.8604.

[26] S. Adriaenssens, L. Ney, E. Bodarwe, and C. Williams, “Finding the Form of an Irregular Meshed Steel and Glass Shell Based on Construction Constraints | Journal of Architectural Engineering | Vol 18, No 3,” *Journal of Architectural Engineering*, vol. 18, no. 3, pp. 206–213, Aug. 2012, doi: 10.1061/(ASCE)AE.1943-5568.0000074.

[27] K.-U. Bletzinger, R. Wüchner, F. Daoud, and N. Camprubí, “Computational methods for form finding and optimization of shells and membranes,” *Computer Methods in Applied Mechanics and Engineering*, vol. 194, no. 30–33, pp. 3438–3452, Aug. 2005, doi: 10.1016/j.cma.2004.12.026.

[28] P. (Philippe C. V. Block, “Thrust Network Analysis : exploring three-dimensional equilibrium,” Thesis, Massachusetts Institute of Technology, 2009. Accessed: Sep. 17, 2025. [Online]. Available: https://dspace.mit.edu/handle/1721.1/49539

[29] J. N. Richardson, S. Adriaenssens, R. Filomeno Coelho, and P. Bouillard, “Coupled form-finding and grid optimization approach for single layer grid shells,” *Engineering Structures*, vol. 52, pp. 230–239, Jul. 2013, doi: 10.1016/j.engstruct.2013.02.017.

[30] Y. Jiang, T. Zegard, W. F. Baker, and G. H. Paulino, “Form-finding of grid-shells using the ground structure and potential energy methods: a comparative study and assessment | Structural and Multidisciplinary Optimization.” Accessed: Sep. 17, 2025. [Online]. Available: https://link.springer.com/article/10.1007/s00158-017-1804-3

[31] F. He, Y. Xiao, S. Yuchi, H. Asadi, R. Wüchner, and P. D’Acunto, *Enhancing Non-Linear Force Density Method through Combinatorial Equilibrium Modelling*. 2024.

[32] J.-P. Jasienski, Y. Shen, P. O. Ohlbrock, D. Zastavni, and P. D’Acunto, “A computational implementation of Vector-based 3D Graphic Statics (VGS) for interactive and real-time structural design,” *Computer-Aided Design*, vol. 171, p. 103695, Jun. 2024, doi: 10.1016/j.cad.2024.103695.

[33] P. O. Ohlbrock and P. D’Acunto, “A Computer-Aided Approach to Equilibrium Design Based on Graphic Statics and Combinatorial Variations,” *Computer-Aided Design*, vol. 121, p. 102802, Apr. 2020, doi: 10.1016/j.cad.2019.102802.

[34] L. Mei and Q. Wang, “Structural Optimization in Civil Engineering: A Literature Review,” *Buildings*, vol. 11, no. 2, p. 66, Feb. 2021, doi: 10.3390/buildings11020066.

[35] T. Müller and E. Van der Klashorst, "A Quantitative Comparison Between Size, Shape, Topology and Simultaneous Optimization for Truss Structures," *Latin American Journal of Solids and Structures*, vol. 14, pp. 2221–2242, Dec. 2017, doi: 10.1590/1679-78253900.

[36] T. Tang *et al.*, "Topology Optimization: A Review for Structural Designs Under Statics Problems," *Materials*, vol. 17, no. 23, p. 5970, Dec. 2024, doi: 10.3390/ma17235970.

[37] P. Li *et al.*, "Optimization Design for Steel Trusses Based on a Genetic Algorithm," *Buildings*, vol. 13, no. 1496, 2023, doi: https://doi.org/10.3390/buildings13061496.

[38] A. Kaveh, K. Biabani Hamedani, S. Milad Hosseini, and T. Bakhshpoori, "Optimal design of planar steel frame structures utilizing meta-heuristic optimization algorithms," *Structures*, vol. 25, pp. 335–346, Jun. 2020, doi: 10.1016/j.istruc.2020.03.032.

[39] A. Liew, "Constrained Force Density Method optimisation for compression-only shell structures," *Structures*, vol. 28, pp. 1845–1856, Dec. 2020, doi: 10.1016/j.istruc.2020.09.078.

[40] V. Tomei, D. Faiella, F. Cascone, and E. Mele, "Structural grammar for design optimization of grid shell structures and diagrid tall buildings," *Automation in Construction*, vol. 143, p. 104588, Nov. 2022, doi: 10.1016/j.autcon.2022.104588.

[41] "[PDF] Computational methods for form finding and optimization of shells and membranes | Semantic Scholar." Accessed: Sep. 17, 2025. [Online]. Available: https://www.semanticscholar.org/paper/Computational-methods-for-form-finding-and-of-and-Bletzinger-W%C3%BCchner/ea2953cbbed7bc426ab171ad838e653c45e1d4bb

[42] L. Ruiz, F. Gama, and A. Ribeiro, "Graph Neural Networks: Architectures, Stability, and Transferability," *Proc. IEEE*, vol. 109, no. 5, pp. 660–682, May 2021, doi: 10.1109/JPROC.2021.3055400.

[43] V. Ponzi and C. Napoli, "Graph Neural Networks: Architectures, Applications, and Future Directions," *IEEE Access*, vol. 13, pp. 62870–62891, 2025, doi: 10.1109/ACCESS.2025.3558752.

[44] J. Shlomi, P. Battaglia, and J. Roch Vlimant, "Graph neural networks in particle physics - IOPscience." Accessed: Sep. 17, 2025. [Online]. Available: https://iopscience.iop.org/article/10.1088/2632-2153/abbf9a

[45] J. Li, Y. Ji, Y. Zhang, J. Ding, C. Li, and X. Li, "Graph neural networks for network science: A mini review," *EPL*, vol. 150, no. 2, p. 21001, Apr. 2025, doi: 10.1209/0295-5075/adc767.

[46] B. Sanchez-Lengeling, E. Reif, A. Pearce, and A. B. Wiltschko, "A Gentle Introduction to Graph Neural Networks," *Distill*, vol. 6, no. 9, p. e33, Sep. 2021, doi: 10.23915/distill.00033.

[47] “Facebook,” *Wikipedia*. Oct. 18, 2025. Accessed: Oct. 20, 2025. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Facebook&oldid=1317601772

[48] “Twitter,” *Wikipedia*. Oct. 18, 2025. Accessed: Oct. 20, 2025. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Twitter&oldid=1317497446

[49] “Instagram,” *Wikipedia*. Oct. 20, 2025. Accessed: Oct. 20, 2025. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Instagram&oldid=1317817858

[50] “A Comprehensive Introduction to Graph Neural Networks (GNNs).” Accessed: Oct. 20, 2025. [Online]. Available: https://www.datacamp.com/tutorial/comprehensive-introduction-graph-neural-networks-gnns-tutorial

[51] “Revisiting Graph Neural Networks for Link Prediction - Meta Research,” Meta Research. Accessed: Oct. 20, 2025. [Online]. Available: https://research.facebook.com/publications/revisiting-graph-neural-networks-for-link-prediction/

[52] N. Nourian, M. El-Badry, and M. Jamshidi, “Design Optimization of Truss Structures Using a Graph Neural Network-Based Surrogate Model.” Accessed: Sep. 17, 2025. [Online]. Available: https://www.mdpi.com/1999-4893/16/8/380

[53] F. Parisi, S. Ruggieri, R. Lovreglio, M. P. Fanti, and G. Uva, “On the use of mechanics-informed models to structural engineering systems: Application of graph neural networks for structural analysis,” *Structures*, vol. 59, p. 105712, Jan. 2024, doi: 10.1016/j.istruc.2023.105712.

[54] P. Zhao, W. Liao, Y. Huang, and X. Lu, “Intelligent beam layout design for frame structure based on graph neural networks,” *Journal of Building Engineering*, vol. 63, p. 105499, Jan. 2023, doi: 10.1016/j.jobe.2022.105499.

[55] C. Zhang, M. Tao, C. Wang, C. Yang, and J. Fan, “Differentiable automatic structural optimization using graph deep learning,” *Advanced Engineering Informatics*, vol. 60, p. 102363, Apr. 2024, doi: 10.1016/j.aei.2024.102363.

[56] C. Kupwiwat, K. Hayashi, and M. Ohsaki, “Frontiers | Deep deterministic policy gradient and graph convolutional network for bracing direction optimization of grid shells.” Accessed: Sep. 17, 2025. [Online]. Available: https://www.frontiersin.org/journals/built-environment/articles/10.3389/fbuil.2022.899072/full

[57] L. Bleker, K.-M. M. Tam, and P. D’Acunto, “Logic-Informed Graph Neural Networks for Structural Form-Finding,” *Advanced Engineering Informatics*, vol. 61, p. 102510, Aug. 2024, doi: 10.1016/j.aei.2024.102510.

[58] R. Pastrana, E. Medina, I. M. de Oliveira, S. Adriaenssens, and R. P. Adams, “Real-time design of architectural structures with differentiable mechanics and neural networks,” Mar. 17, 2025, *arXiv*: arXiv:2409.02606. doi: 10.48550/arXiv.2409.02606.

[59] T. V. Mele, R. Matthias, and L. Lorenz, "Geometry-based Understanding of Structures," *Journal of the International Association for Shell and Spatial Structures*, vol. 53, Accessed: Sep. 18, 2025. [Online]. Available: https://www.semanticscholar.org/paper/OF-THE-INTERNATIONAL-ASSOCIATION-FOR-SHELL-AND-%3A-J-Mele-Lachauer/f4e42ec53a1939a9f7a66114da03c79979302c07

[60] K.-M. M. Tam, "Learning Discrete Equilibrium: Trans-topological Inverse Pattern and Force Design Using Machine Learning & Automatic Differentiation," ETH Zurich, 2023. doi: 10.3929/ethz-b-000661366.

[61] A. Borgart, "The relationship between geometric and mechanical properties of shell structures," 2024, doi: 10.4233/uuid:00ab51d6-55db-453a-9a62-23dfbd439b7b.

[62] G. D. Nikolov, "Bayesian Optimization for Weight Minimization of Gridshells in a Complex Design Space," 2025, Accessed: Oct. 03, 2025. [Online]. Available: https://repository.tudelft.nl/record/uuid:19413211-52c0-41a0-a1fc-302fa9a6dec0

[63] "Inextensional deformation." Accessed: Oct. 03, 2025. [Online]. Available: https://www.phoogenboom.nl/b17_handout_5.pdf

[64] "fminunc - Find minimum of unconstrained multivariable function - MATLAB." Accessed: Sep. 25, 2025. [Online]. Available: https://in.mathworks.com/help/optim/ug/fminunc.html

[65] "Cholesky's Decomposition - an overview | ScienceDirect Topics." Accessed: Sep. 26, 2025. [Online]. Available: https://www.sciencedirect.com/topics/engineering/choleskys-decomposition

[66] A. Manguri, H. Hassan, N. Saeed, and R. Jankowski, "Topology, Size, and Shape Optimization in Civil Engineering Structures: A Review," *CMES - Computer Modeling in Engineering and Sciences*, vol. 142, no. 2, pp. 933–971, Jan. 2025, doi: 10.32604/cmes.2025.059249.

[67] P. T. Boggs and J. W. Tolle, "Sequential Quadratic Programming," *Acta Numerica*, vol. 4, pp. 1–51, Jan. 1995, doi: 10.1017/S0962492900002518.

[68] H. Xu, Q. Deng, Z. Zhange, and S. Lin, "A hybrid differential evolution particle swarm optimization algorithm based on dynamic strategies | Scientific Reports." Accessed: Sep. 26, 2025. [Online]. Available: https://www.nature.com/articles/s41598-024-82648-5

[69] A. G. Gad, "Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review | Archives of Computational Methods in Engineering." Accessed: Sep. 26, 2025. [Online]. Available: https://link.springer.com/article/10.1007/s11831-021-09694-4

[70] P. Hajela, E. Lee, and C.-Y. Lin, "Genetic Algorithms in Structural Topology Optimization," in *Topology Design of Structures*, M. P. Bendsøe and C. A. M. Soares, Eds., Dordrecht: Springer Netherlands, 1993, pp. 117–133. doi: 10.1007/978-94-011-1804-0_10.

[71] R. Eberhart and J. Kennedy, “A new optimizer using particle swarm theory | IEEE Conference Publication | IEEE Xplore.” Accessed: Sep. 26, 2025. [Online]. Available: https://ieeexplore.ieee.org/document/494215

[72] J. Wang, H. Ouyang, C. Zhang, S. Li, and J. Xiang, “A novel intelligent global harmony search algorithm based on improved search stability strategy | Scientific Reports.” Accessed: Sep. 26, 2025. [Online]. Available: https://www.nature.com/articles/s41598-023-34736-1

[73] X. She Yang, “Firefly Algorithms for Multimodal Optimization | SpringerLink.” Accessed: Sep. 26, 2025. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-04944-6_14

[74] M. Ohsaki, “Optimization of Finite Dimensional Structures | Makoto Ohsaki | Taylor.” Accessed: Sep. 27, 2025. [Online]. Available: https://www.taylorfrancis.com/books/mono/10.1201/EBK1439820032/optimization-finite-dimensional-structures-makoto-ohsaki

[75] K. M. Mueller, M. Liu, S. A. Burns, K. M. Mueller, M. Liu, and S. A. Burns, “Fully Stressed Design of Frame Structures and Multiple Load Paths,” *Journal of Structural Engineering*, vol. 128, no. 6, pp. 806–814, Jan. 2002.

[76] H. Cao, M. Li, L. Nie, Y. Xie, and F. Kong, “Vertex-based graph neural network classification model considering structural topological features for structural optimization,” *Computers & Structures*, vol. 305, p. 107542, Dec. 2024, doi: 10.1016/j.compstruc.2024.107542.

[77] 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*, vol. 299, p. 107385, Aug. 2024, doi: 10.1016/j.compstruc.2024.107385.

[78] P. Zhao, W. Liao, Y. Huang, and X. Lu, “Intelligent beam layout design for frame structure based on graph neural networks,” *Journal of Building Engineering*, vol. 63, p. 105499, Jan. 2023, doi: 10.1016/j.jobe.2022.105499.

[79] Y. Zhao, H. Li, H. Zhou, H. R. Attar, T. Pfaff, and N. Li, “A review of graph neural network applications in mechanics-related domains,” *Artif Intell Rev*, vol. 57, no. 11, p. 315, Oct. 2024, doi: 10.1007/s10462-024-10931-y.

[80] A. Kaveh, *Structural Mechanics: Graph and Matrix Methods*. Research Studies Press, 2004.

[81] A. Kaveh, “Optimal Analysis of Structures by Concepts of Symmetry and Regularity | SpringerLink.” Accessed: Sep. 27, 2025. [Online]. Available: https://link.springer.com/book/10.1007/978-3-7091-1565-7

[82] M. Defferrard, X. Bresson, and P. Vandergheynst, “Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering,” Feb. 05, 2017, *arXiv*: arXiv:1606.09375. doi: 10.48550/arXiv.1606.09375.

[83] T. N. Kipf and M. Welling, “Semi-Supervised Classification with Graph Convolutional Networks,” Feb. 22, 2017, *arXiv*: arXiv:1609.02907. doi: 10.48550/arXiv.1609.02907.

[84] W. L. Hamilton, R. Ying, and J. Leskovec, “Inductive Representation Learning on Large Graphs,” Sep. 10, 2018, *arXiv*: arXiv:1706.02216. doi: 10.48550/arXiv.1706.02216.

[85] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, “Graph Attention Networks,” Feb. 04, 2018, *arXiv*: arXiv:1710.10903. doi: 10.48550/arXiv.1710.10903.

[86] Ii. Makarov, D. Kiselev, N. Nikitinsky, and L. Subelj, “Survey on graph embeddings and their applications to machine learning problems on graphs [PeerJ].” Accessed: Sep. 27, 2025. [Online]. Available: https://peerj.com/articles/cs-357/

[87] N. Li, W. Li, Y. Gao, Y. Li, and J. Bao, “Node-Level Graph Regression With Deep Gaussian Process Models | Semantic Scholar.” Accessed: Sep. 27, 2025. [Online]. Available: https://www.semanticscholar.org/paper/Node-Level-Graph-Regression-With-Deep-Gaussian-Li-Li/dd88d74dbf1e17899b8daf3b6ae72ddd30159260

[88] X. Ling *et al.*, “Multilevel Graph Matching Networks for Deep Graph Similarity Learning,” *IEEE Trans. Neural Netw. Learning Syst.*, vol. 34, no. 2, pp. 799–813, Feb. 2023, doi: 10.1109/TNNLS.2021.3102234.

[89] Y. Xie, M. Gong, Y. Gao, A. K. Qin, and X. Fan, “Frontiers | A Multi-Task Representation Learning Architecture for Enhanced Graph Classification.” Accessed: Sep. 27, 2025. [Online]. Available: https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2019.01395/full

[90] F. Xia *et al.*, “Graph Learning: A Survey,” *IEEE Trans. Artif. Intell.*, vol. 2, no. 2, pp. 109–127, Apr. 2021, doi: 10.1109/TAI.2021.3076021.

[91] H. Gao and S. Ji, “Graph U-Nets,” May 11, 2019, *arXiv*: arXiv:1905.05178. doi: 10.48550/arXiv.1905.05178.

[92] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, “Neural Message Passing for Quantum Chemistry,” Jun. 12, 2017, *arXiv*: arXiv:1704.01212. doi: 10.48550/arXiv.1704.01212.

[93] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, “A Comprehensive Survey on Graph Neural Networks,” *IEEE Trans. Neural Netw. Learning Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021, doi: 10.1109/TNNLS.2020.2978386.

[94] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, “How Powerful are Graph Neural Networks?,” Feb. 22, 2019, *arXiv*: arXiv:1810.00826. doi: 10.48550/arXiv.1810.00826.

[95] Q. Li, Z. Han, and X.-M. Wu, "Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning," Jan. 22, 2018, *arXiv*: arXiv:1801.07606. doi: 10.48550/arXiv.1801.07606.

[96] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The Graph Neural Network Model | IEEE Journals & Magazine | IEEE Xplore." Accessed: Sep. 28, 2025. [Online]. Available: https://ieeexplore.ieee.org/document/4700287

[97] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. Salakhutdinov, and A. Smola, "Deep Sets," Apr. 14, 2018, *arXiv*: arXiv:1703.06114. doi: 10.48550/arXiv.1703.06114.

[98] C. Wu, S. Jiang, and S. Zhu, "GRIDSNET: A graph neural network approach to predicting nonlinear buckling capacity of imperfect single-layer gridshells," *Thin-Walled Structures*, vol. 208, p. 112851, Mar. 2025, doi: 10.1016/j.tws.2024.112851.

[99] "Kullback–Leibler divergence - Wikipedia." Accessed: Oct. 20, 2025. [Online]. Available: https://en.wikipedia.org/wiki/Kullback%E2%80%93Leibler_divergence

[100] A. Favilli, F. Laccone, P. Cignoni, L. Malomo, and D. Giorgi, "Geometric deep learning for statics-aware grid shells," *Computers & Structures*, vol. 292, p. 107238, Feb. 2024, doi: 10.1016/j.compstruc.2023.107238.

[101] "Kangaroo - Live Physics Engine for Rhino 3D." Accessed: Oct. 20, 2025. [Online]. Available: https://simplyrhino.co.uk/3d-modelling-software/kangaroo

[102] "Karamba3D." Accessed: Oct. 20, 2025. [Online]. Available: https://karamba3d.com/

[103] P. Zhao, W. Liao, Y. Huang, and X. Lu, "Beam layout design of shear wall structures based on graph neural networks," *Automation in Construction*, vol. 158, p. 105223, Feb. 2024, doi: 10.1016/j.autcon.2023.105223.

[104] "Eurocode 5: Design of timber structures | Eurocodes: Building the future." Accessed: Oct. 04, 2025. [Online]. Available: https://eurocodes.jrc.ec.europa.eu/EN-Eurocodes/eurocode-5-design-timber-structures

[105] J. Olivier Lachaud, "DGtal: Half-edge data structure, triangulated surfaces and polygonal surfaces." Accessed: Oct. 04, 2025. [Online]. Available: https://dgtal-team.github.io/doc-nightly/moduleHalfEdgeMesh.html

[106] "HalfEdge – three.js docs." Accessed: Oct. 04, 2025. [Online]. Available: https://threejs.org/docs/#examples/en/math/convexhull/HalfEdge

[107] L. Tychonievich, "Half-edge data structure." Accessed: Oct. 04, 2025. [Online]. Available: https://cs418.cs.illinois.edu/website/text/halfedge.html

[108] "Principal component analysis," *Wikipedia*. Oct. 02, 2025. Accessed: Oct. 04, 2025. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Principal_component_analysis&oldid=1314713685

[109] F. Tampieri, “V.5 - NEWELL’S METHOD FOR COMPUTING THE PLANE EQUATION OF A POLYGON,” in *Graphics Gems III (IBM Version)*, D. Kirk, Ed., San Francisco: Morgan Kaufmann, 1992, pp. 231–232. doi: 10.1016/B978-0-08-050755-2.50052-X.

[110] S. E. Leon, G. H. Paulino, A. Pereira, I. F. M. Menezes, and E. N. Lages, “A Unified Library of Nonlinear Solution Schemes,” *Appl. Mech. Rev*, vol. 64, no. 040803, Aug. 2012, doi: 10.1115/1.4006992.

[111] “EN 338 - Table 1 - Strength classes - Characteristic Values.” Accessed: Oct. 04, 2025. [Online]. Available: http://onlinestructuraldesign.com/calcs/Wood_documentation/Wood_element_EN338_Table1.htm

[112] S. Adriaenssens, L. NEY, E. BODARWE, and C. Williams, “Dutch Maritime Museum: Form-finding of an irregular faceted skeletal shell - Part b,” Jan. 2010.

[113] “numpy.linspace — NumPy v2.4.dev0 Manual.” Accessed: Oct. 09, 2025. [Online]. Available: https://numpy.org/devdocs//reference/generated/numpy.linspace.html

[114] “StandardScaler — scikit-learn 1.7.2 documentation.” Accessed: Oct. 10, 2025. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

[115] “torch_geometric.data.HeteroData — pytorch_geometric documentation.” Accessed: Oct. 09, 2025. [Online]. Available: https://pytorch-geometric.readthedocs.io/en/2.5.0/generated/torch_geometric.data.HeteroData.html

[116] “Automatic Mixed Precision — PyTorch Tutorials 2.9.0+cu128 documentation.” Accessed: Oct. 31, 2025. [Online]. Available: https://docs.pytorch.org/tutorials/recipes/recipes/amp_recipe.html

[117] “AdamW — PyTorch 2.9 documentation.” Accessed: Oct. 31, 2025. [Online]. Available: https://docs.pytorch.org/docs/stable/generated/torch.optim.AdamW.html

[118] “ReduceLROnPlateau — PyTorch 2.9 documentation.” Accessed: Oct. 31, 2025. [Online]. Available: https://docs.pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.ReduceLROnPlateau.html

[119] O. Rainio, J. Teuho, and R. Klén, “Evaluation metrics and statistical tests for machine learning,” *Sci Rep*, vol. 14, no. 1, p. 6086, Mar. 2024, doi: 10.1038/s41598-024-56706-x.

[120] C. Miller, T. Portlock, D. M. Nyaga, and J. M. O’Sullivan, “A review of model evaluation metrics for machine learning in genetics and genomics,” *Front Bioinform*, vol. 4, p. 1457619, Sep. 2024, doi: 10.3389/fbinf.2024.1457619.

[121] F. Emmert-Streib and M. Dehmer, “Evaluation of Regression Models: Model Assessment, Model Selection and Generalization Error,” *Machine Learning and*

Knowledge Extraction, vol. 1, no. 1, pp. 521–551, Mar. 2019, doi: 10.3390/make1010032.

[122] A. V. Tatachar, “Comparative Assessment of Regression Models Based On Model Evaluation Metrics,” vol. 08, no. 09, 2021.

[123] E. Lewinson, “A Comprehensive Overview of Regression Evaluation Metrics,” NVIDIA Technical Blog. Accessed: Oct. 09, 2025. [Online]. Available: https://developer.nvidia.com/blog/a-comprehensive-overview-of-regression-evaluation-metrics/

[124] “torch_geometric.nn.conv.HeteroConv — pytorch_geometric documentation.” Accessed: Oct. 10, 2025. [Online]. Available: https://pytorch-geometric.readthedocs.io/en/2.5.1/generated/torch_geometric.nn.conv.HeteroConv.html

[125] “torch_geometric.nn.conv.GATConv — pytorch_geometric documentation.” Accessed: Oct. 10, 2025. [Online]. Available: https://pytorch-geometric.readthedocs.io/en/2.5.3/generated/torch_geometric.nn.conv.GATConv.html