EG-ICE 2025
GLASGOW

Graph Neural Networks for Fast Structural Analysis to support Interactive Design Decision-Making

SHIH-PU KUO [1], PIERLUIGI D'ACUNTO [2], IAN F.C. SMITH [3]

[1] TUM Georg Nemetschek Institute Artificial Intelligence for the Built World, Munich, Germany
[2] Professorship of Structural Design, School of Engineering and Design, Technical University of Munich, Munich, Germany
[3] School of Architecture, Civil and Environmental Engineering, Swiss Federal Institute of Technology, Lausanne, Switzerland

ABSTRACT

This study presents a graph-based surrogate modeling framework for fast prediction of structural responses to support the preliminary design of multi-story buildings under static wind loads. Structures are encoded as graphs, with joints as nodes and beams and columns as edges, enriched by topological and physical attributes. Message-passing Graph Neural Networks (GNNs) are trained on a synthetic dataset to learn a predictive mapping from structural form, load paths, and nodal structural responses. The model predicts displacements, shear forces, and bending moments with high accuracy and demonstrates promising generalization for taller, unseen configurations. Once trained, the GNNs provide near-instantaneous predictions, enabling rapid exploration of design alternatives. To explore its educational potential, this model is also integrated into a Rhino-Grasshopper environment to support early-stage design workflows and interactive feedback. This approach offers a fast and scalable alternative to traditional simulations, helping users make more informed decisions during conceptual design.

KEYWORDS

Graph Neural Networks, Extrapolation, Structural Analysis, Decision-Making

1. INTRODUCTION

Assessing structural performance under diverse load scenarios is essential for informed decision-making during the early stages of structural design. In the conceptual phase, where approximately 80% of project costs are irreversibly committed, novice designers often explore a wide range of spatial and structural configurations with limited access to timely analytical feedback (Arciszewski, T., & Lakmazaheri, S., 2001).

While experienced engineers weigh design specifications and trade-offs, novice designers often rely on simplified assumptions in the absence of immediate feedback (Boulanger, S., & Smith, I., 2001). Traditional simulation tools such as FEM software are typically too time-consuming and input-intensive to support rapid iteration in early-stage workflows. As a result, decisions are commonly based on intuition or delayed expert review, limiting opportunities for timely learning and iterative refinement.

Early structural feedback plays a key role in developing engineering intuition and design reasoning (Horikx, M. P., 2022; D'Acunto, P., et al., 2024). However, in both academic and professional contexts, accessible and responsive structural analysis tools remain limited. Many existing methods require detailed inputs, specialized software, or expert interpretation, which pose barriers to early-stage design workflows. This has led to increasing interest in data-driven methods for performance estimation during the conceptual design process (Kazemi, P., et al., 2023).

Machine learning (ML) has emerged as a promising alternative to traditional simulation, offering rapid-response predictive models for faster structural feedback (Thai, H.-T, 2022; Kazemi, P., et al., 2024). Despite this potential, most conventional ML models require fixed input formats and lack the flexibility to accommodate geometric and topological variations inherent in structural systems. These models often struggle to extrapolate beyond their training distribution, which limits their reliability for generalized structural reasoning. Graph Neural Networks (GNNs) are thus introduced in this study, which are inherently suited to representing and learning from graph-structured data such as building frames.

GNNs operate on graph-structured data and learn by propagating information through a system using a process known as message-passing, a process well-aligned with how forces propagate through structural frames (Battaglia, P. W., et al., 2018; Zhou, J., et al., 2020). In this framework, joints are modeled as nodes, and structural members are set as edges. Recent research has applied GNNs to structural tasks such as performance prediction and topology optimization, demonstrating their effectiveness in structural design workflows (Chang, K.-H., & Cheng, C.-Y., 2020; Bleker, L., et al., 2023; Bleker, L., et al., 2024).

Once trained, these GNN-based models function as surrogate models, capable of delivering near-instantaneous predictions of structural response under specified loading conditions. This enables rapid feedback and efficient exploration of large design spaces, significantly accelerating the early design process (Whalen, E., & Mueller, C., 2022).

This study proposes a GNN-based surrogate modeling framework that predicts nodal displacements, shear forces, and bending moments in multi-story building structures by encoding each configuration as a graph that integrates both topological relationships and physical attributes. The model is trained on a synthetically generated dataset and evaluated on both in-distribution and out-of-distribution cases to assess its generalization capacity, particularly in scenarios where conventional ML or analytical methods would struggle.

Beyond predictive performance, the proposed framework has been preliminarily developed as an interactive design tool to explore its potential in structural education. This environment enables users to generate and assess building alternatives in real time, receiving immediate, visual feedback on predicted structural behavior. While still in an early stage of development, the tool is intended to support the development of structural intuition, enhance early-stage design decision-making, and bridge the gap between conceptual design and engineering reasoning.

The main contributions of this work are:

- A graph-based representation scheme for encoding structural configurations and member properties.
- A GNN architecture capable of predicting values for multiple structural response variables under static wind loading.
- A demonstration of the model's generalization to unseen structural variants, establishing it as a reusable surrogate for early-stage performance prediction and educational feedback.

The remainder of this paper is organized as follows. Section 2 presents the methodology, covering graph representation, the GNN model, dataset generation, and the interactive prediction workflow. Section 3 reports the results, including model performance, inference time and efficiency, and its educational application. Section 4 discusses key implications and limitations. Section 5 concludes with a summary and future directions.

2. METHODOLOGY

This section outlines the process of modeling building structures as graphs and using GNNs to predict structural responses under wind loads. It includes the graph representation of the structural system, the GNN architecture, the generation of a synthetic training dataset, and the integration of the trained model into an interactive design environment.

2.1 Graph representation

In this study, each multi-story building is represented as a graph, where joints are treated as nodes, and structural members such as columns and beams are modeled as edges. The connectivity among these elements defines the graph topology, which allows the model to process relational information in a way that may resemble how forces propagate through the structural system. This representation mimics the physical behavior of load transmission using graph-based information flow.

To construct each graph, three categories of parameters are considered. Topological parameters define how joints are interconnected by structural members, specifying the connectivity of the structural system and forming the graph's edge index. Metric parameters describe the geometric and physical characteristics of nodes and edges, including nodal coordinates, member dimensions, boundary conditions, material properties, and applied loads. Response variables represent the structural outputs at each joint under these loading conditions, including displacements, shear forces, and bending moments. These values serve as supervised learning targets during model training.

Each graph is composed of four core components that reflect the previously described parameter categories: node features store the metric attributes assigned to each joint; edge attributes capture physical properties of the connecting members; the edge index defines the node adjacency relationships; and node targets represent the nodal structural responses at each joint under applied loads, as illustrated in Figure 1.

This graph-based formulation provides a structured and scalable framework for encoding structural systems, allowing the GNN to infer complex behaviors based on topology and material configuration. This encoding preserves structural connectivity and load paths directly, enabling the model to learn relational dependencies that would be difficult to capture using conventional vectorized input formats.

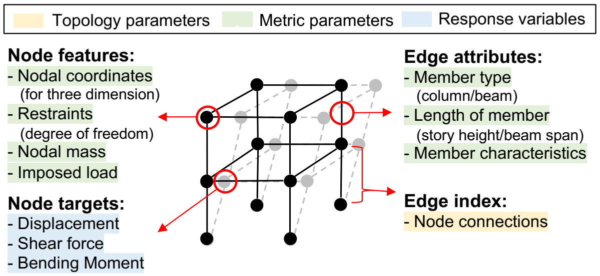


Figure 1: Definition of parameters in a building graph.

2.2 GNN architecture

The GNN model developed in this study is designed to learn the mapping between graph-structured building representations and corresponding structural response fields. The architecture of this model is composed of three main components (see Figure 2), each responsible for a specific stage in the prediction process:

- Input Layer: Converts raw node features into high-dimensional latent embeddings to capture complex feature interactions.
- Message Passing Layers: A series of GNN layers that iteratively update node representations by aggregating information from neighboring nodes and edge attributes. This process reflects how forces are transmitted through structural systems. The number of layers is tuned to ensure information reaches distant nodes as the building topology evolves.
- Readout Layer: Decodes the final node embeddings to produce predicted structural responses at each joint. This layer serves as the regression head of the model.

This GNN model was implemented using PyTorch Geometric (Fey, M., & Lenssen, J. E., 2019) and trained for 100 epochs with the ADAM optimizer (Kingma, D. P., & Ba, J., 2017) and a mean absolute error (MAE) loss function. Model architecture, including the number of layers, embedding dimensions, and activation functions, was refined through hyperparameter tuning to achieve a suitable balance between capacity and generalization.

Serving as a surrogate model, this GNN provides a fast, data-driven approximation of structural simulations. After training, it can deliver nodal response under static wind loads within milliseconds, effectively bypassing the need for iterative FEM analyses. This capability proves especially valuable during the early stages of design, where iteration is frequent and rapid evaluation of multiple alternatives is essential. The graph-based framework enables the model to generalize across diverse building configurations by operating directly on structural geometry and topology, without depending on manually defined solver settings or analysis configurations. In doing so, the GNN not only replicates structural behavior but also offers a reusable and scalable solution for efficient structural assessment.

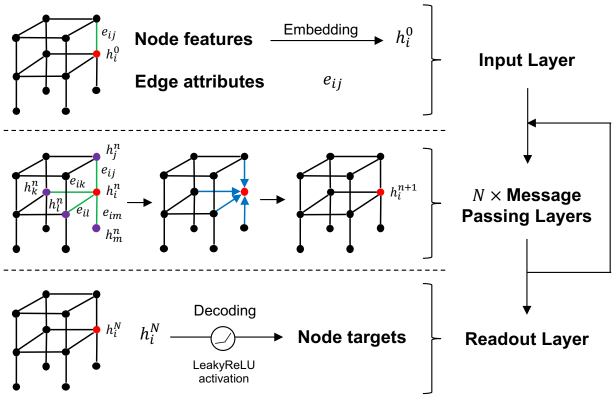


Figure 2: Architecture of GNN model.

2.3 Synthetic dataset generation

To train and evaluate the GNN, a synthetic dataset was created to simulate a wide range of early-stage building designs. Each sample in the dataset represents a distinct multi-story building configuration, structured as a graph with randomized design specifications that reflect practical design variability, as shown in Table 1.

The generation process begins by setting site-related constraints, such as total floor area and the length-to-width (L/W) ratio of the building footprint. These values are sampled within defined bounds to simulate realistic early-stage design scenarios.

Based on this footprint, the structural topology is determined by assigning a number of stories, positioning joints on each floor, and defining maximum beam spans. To capture differences in structural stiffness and geometry, standardized cross-sectional profiles are assigned to columns and beams from a range of commonly used sections.

The dataset comprises 3,000 building alternatives with 6 to 20 stories, of which 90% are used for training and the remaining 10% constitute an in-distribution (ID) test set. In addition, an out-of-distribution (OOD) test set of 300 graphs is constructed using taller buildings with 21 to 25 stories. This set allows assessment of the model's ability to generalize beyond the structural configurations seen during training.

The analysis was performed using a linear elastic static solver from the FEM software SAP2000 (CSI, 2002). Each model is subjected to self-weight and static wind loads, in accordance with Eurocode EN 1991-1-4 (BSI, 2005) and the German annex DIN EN 1991-1-4/NA:2010-12 (DIN, 2010). The resulting nodal displacements, shear forces, and bending moments are extracted as ground-truth targets for supervised training of the GNN. This synthetic dataset provides a diverse foundation for learning structural behavior across a broad design space.

Table 1: The range of design specifications.

Design Specification	Training, ID Test Set	OOD Test Set
Story count	6-20	21-25
Floor area (m^2)	500-1000	
Building L/W ratio	1.0-3.0	
Story height (m)	4	
Max. beam span (m)	6	
Column cross-section	SHS 300x12.5-SHS 400x16	
Beam cross-section	HE 200A-HE 400A	

2.4 Interactive prediction workflow

A prototype interface was developed to integrate the trained GNN surrogate model into an interactive early-stage design workflow. The system is implemented within Rhino (McNeel, R., & others, 2024), a widely used computer-aided design (CAD) platform in architecture and engineering. Parametric control is accommodated through Grasshopper (Ibid.), a visual programming environment that operates as a plugin within Rhino and enables rule-based generation of structural geometry.

The interface is configured to interactively generate design alternatives based on user-defined design specifications. Each configuration is automatically converted into a graph representation consistent with the GNN model's input format, with joints and members encoded as nodes and edges, respectively. Prediction is performed using embedded Python scripts that preprocess the graph and pass it to the trained model. The GNN model serves as a surrogate predictor for nodal structural responses under static wind loads, enabling rapid performance estimation during design exploration. These results are visualized in Rhino as scaled predicted quantities, providing intuitive feedback on structural behavior. The complete process is illustrated in Figure 3.

Although the GNN model is trained to predict displacement, shear force, and bending moment, the current prototype interface focuses on displacement as a preliminary application, with future development planned to include additional structural responses. This workflow demonstrates the feasibility of integrating graph-based machine learning with parametric modeling environments, enabling real-time structural feedback during conceptual exploration. It also provides a foundation for educational applications, where immediate performance insights can strengthen structural intuition and support design reasoning.

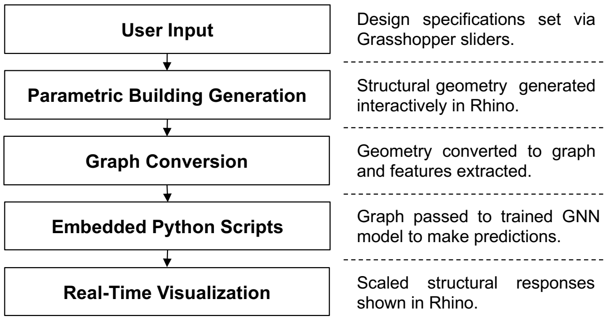


Figure 3: Interactive workflow for structural prediction.

3. RESULTS

This section presents the model's predictive performance, inference efficiency, and potential applications in early-stage structural design and education. It reports accuracy results across training, in-distribution, and out-of-distribution test sets, followed by a comparison of inference time against conventional FEM simulations. The section concludes with a demonstration of how the model can support real-time structural feedback and conceptual reasoning within a visual programming environment.

3.1 Model performance

The model's performance was evaluated using the mean absolute percentage error (MAPE), calculated separately for displacement, shear force, and bending moment at each node. Results are reported across three dataset splits: training set, in-distribution test set, and out-of-distribution test set.

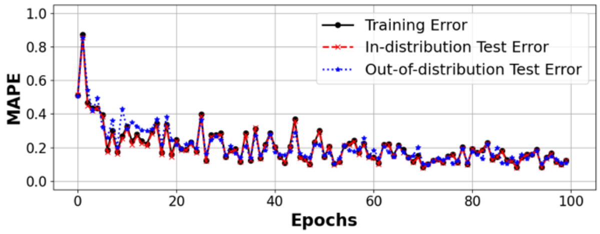
Particular attention is given to the model's ability to generalize beyond its training distribution, especially to unseen configurations in the out-of-distribution test set. MAPE is defined as Equation (1):

$$\mathrm{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i-\hat{y}_i}{y_i}\right| \qquad (1)$$

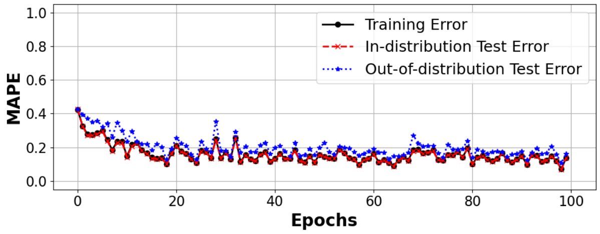
where y_i is the ground truth value, $\hat{y}_i$ is the predicted value, and n is the number of predictions.

Figure 4 shows the MAPE trends over 100 training epochs for displacement, shear force, and bending moment, respectively. As shown in Figure 4(a), the displacement model converges rapidly within the first 10 epochs, with training and in-distribution test curves closely aligned throughout. The out-of-distribution test curve stabilizes after approximately 50 epochs, showing only minor fluctuations, which suggests strong generalization for displacement prediction. The full 100-epoch range is included to demonstrate that the model remains stable throughout training and does not suffer from overfitting, despite early convergence.

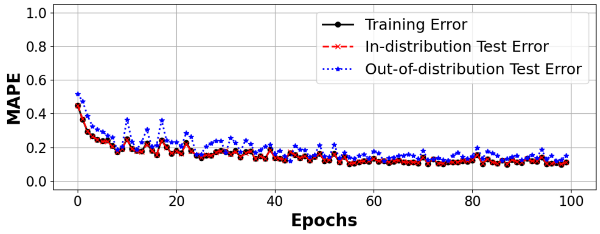
Figure 4(b) demonstrates a similarly stable pattern for shear force prediction. The training and in-distribution test curves closely follow one another, while the out-of-distribution test curve remains slightly higher but consistent, with low variability across epochs. This indicates good performance with minimal overfitting.



(a) Displacement



(b) Shear force



(c) Bending moment

Figure 4: MAPE curves of GNN model.

Figure 4(c) shows the MAPE curve for bending moment, which presents slightly more fluctuation in the out-of-distribution test set, particularly in the early stages of training. While training and in-distribution test performance remain stable, the greater variability on unseen data suggests that the bending moment is more sensitive to geometric and topological variation. Despite this, all curves show a downward trend and reach a stable range, confirming that the model successfully captures key structural behaviors across varying response types.

Final MAPE results are summarized in Table 2. The model achieves strong accuracy across all dataset splits, with mean prediction errors below 10% for both the training set and in-distribution test set, and below 12% for the out-of-distribution test set. These results confirm the model's ability to learn meaningful structural relationships from the graph-encoded representations and to generalize well to previously unseen configurations.

Although individual structural responses may differ in sensitivity to local geometric variations and discontinuities in stiffness, the model consistently converges across all outputs and maintains low error across diverse building scenarios. The relatively low extrapolation error, in particular, supports the claim that graph-based learning can extend beyond interpolation and capture structural response patterns across configurations with varying story counts and geometric proportions.

Table 2: MAPE comparison across datasets.

Structural Responses	Training Set	ID Test Set	OOD Test Set
Displacement	8.2%	8.0%	9.7%
Shear force	7.1%	7.0%	10.7%
Bending moment	9.8%	10.0%	12.0%

3.2 Inference time and efficiency

To evaluate the efficiency of the proposed surrogate model, its inference time against conventional FEM simulations using SAP2000 is compared. A batch of 100 structural configurations was tested on the same machine under consistent conditions. As shown in Table 3, the simulation of a single structural configuration, which includes model setup, load application, and response extraction, typically takes approximately 20 seconds using SAP2000 on a standard workstation. The time required for FEM analysis also increases with the complexity of the structural model, including the number of stories and connectivity. In contrast, the trained GNN generates predictions for nodal structural responses in less than 0.1 seconds. This allows hundreds of design alternatives to be evaluated within minutes, making it feasible to incorporate structural analysis into iterative workflows and early-stage design exploration.

The GNN model eliminates the need for manual input and significantly reduces evaluation time. This speed advantage makes the GNN especially suitable for early-stage workflows where designers may need to evaluate dozens or even hundreds of alternatives. The ability to generate immediate structural feedback supports interactive use cases such as parametric design exploration, rapid screening of concepts, and educational experimentation. Although training the model requires an initial computational investment, this is a one-time process that enables high-throughput prediction for subsequent tasks.

Table 3: Comparison of Inference Time.

Comparison Item	SAP2000 (FEM Simulation)	GNN Surrogate Model
Total time (100 cases)	35 min 24.75 sec	7.23 sec
Average time per case	21.25 sec	0.072 sec
User input complexity	High	Low
Suitability for iteration	Low	High

3.3 Interactive feedback for learning

In addition to its predictive accuracy, the proposed GNN-based model offers significant potential as an educational tool for conceptual structural design. The current implementation allows students to generate building alternatives using Grasshopper and immediately view predicted displacements within Rhino, providing live, visual feedback during early design. This integration enables learners to explore diverse structural configurations without the time-consuming setup and analysis required in traditional FEM software.

The intended users are senior undergraduate students in architecture and engineering, especially those in project-based integrated design courses. These learners typically possess foundational structural knowledge but often face challenges in evaluating design decisions efficiently and confidently. By enabling immediate feedback, the surrogate model supports the development of structural intuition and encourages exploration during the conceptual phase, when design flexibility and openness to iteration are most valuable.

Unlike traditional tools such as SAP2000, which require detailed modeling and analysis for each alternative, this framework allows students to rapidly test and compare options in a design-oriented environment. While hand calculations can be useful for validating individual cases, they are impractical for exploring multiple design variations. Moreover, trial-and-error methods often lack reliable feedback, leaving students uncertain about the structural soundness of their proposals.

The proposed approach addresses these limitations by embedding structural feedback directly into a visual and interactive workflow. Students can engage with “what if” scenarios, such as adjusting building height, span length, or cross-sectional properties, and receive immediate predictions of structural behavior. This promotes faster decision-making, strengthens early-stage reasoning, and helps bridge the gap between structural analysis and creative design thinking.

Figures 5 and 6 illustrate the current capabilities of the interactive interface. Designed with accessibility and transparency in mind, the interface lowers the barrier to entry for students and novice designers. Instead of requiring users to define complex structural models or scripts, it offers an intuitive visual environment where alternatives are automatically generated, analyzed, and visualized.

As shown in Figure 5, a wide range of building alternatives can be generated based on user-defined design specifications. Each configuration is annotated with key parameters such as total floor area, L/W ratio, story count, and estimated member cost. While the models are generated and explored in 3D, the alternatives are shown here in a consistent top-down view to facilitate comparison of structural layouts and visualization of the associated parameter plot. This layout allows students to clearly identify plan geometry variations and relate them to changes in the input values.

Figure 6 presents a perspective view of the predicted displacement results mapped onto the deformed building geometries. Each configuration is shown with its displaced shape under static wind loading, and nodal displacements are color-coded using a spectral gradient. Nodes with larger displacements are rendered in warmer colors such as red and orange, while smaller movements appear in cooler tones such as green and blue.

By combining geometric deformation with a visual spectrum of displacement magnitudes, the interface allows students to intuitively compare the relative performance of different design alternatives. This enables learners to assess how design decisions influence overall stability and serviceability. In particular, lateral displacements are closely related to the drift ratio, a key metric for evaluating occupant comfort and structural safety. Excessive drift can result in serviceability issues or violations of design standards such as those in Eurocode or ASCE 7. The color overlay helps students identify areas of concern and develop reasoning about which configurations are more structurally effective.

Design 1
Total Floor Area: 8179.80m²
Story: 16
L/W Ratio: 1.7
Height: 64 m
Member Cost: 431470.36 €

Design 2
Total Floor Area: 8068.06m²
Story: 16
L/W Ratio: 1.1
Height: 64 m
Member Cost: 443061.09 €

Design 3
Total Floor Area: 8288.14m²
Story: 17
L/W Ratio: 1.7
Height: 68 m
Member Cost: 452924.71 €

Design 4
Total Floor Area: 8185.77m²
Story: 17
L/W Ratio: 1.1
Height: 68 m
Member Cost: 465323.55 €

Design 5
Total Floor Area: 8275.65m²
Story: 18
L/W Ratio: 1.6
Height: 72 m
Member Cost: 472203.38 €

Design 6
Total Floor Area: 8367.47m²
Story: 18
L/W Ratio: 1.6
Height: 72 m
Member Cost: 473507.14 €

Design 7
Total Floor Area: 8399.83m²
Story: 18
L/W Ratio: 1.8
Height: 72 m
Member Cost: 474828.05 €

Design 8
Total Floor Area: 8370.67m²
Story: 18
L/W Ratio: 1.9
Height: 72 m
Member Cost: 475085.68 €

Design 9
Total Floor Area: 8179.21m²
Story: 16
L/W Ratio: 2.0
Height: 64 m
Member Cost: 484293.69 €

Design 10
Total Floor Area: 8098.77m²
Story: 16
L/W Ratio: 2.3
Height: 64 m
Member Cost: 484805.29 €

Design 11
Total Floor Area: 8191.04m²
Story: 16
L/W Ratio: 2.1
Height: 64 m
Member Cost: 484926.63 €

Design 12
Total Floor Area: 8083.13m²
Story: 18
L/W Ratio: 1.2
Height: 72 m
Member Cost: 484968.82 €

Design 13
Total Floor Area: 8292.73m²
Story: 16
L/W Ratio: 1.8
Height: 64 m
Member Cost: 485434.93 €

Design 14
Total Floor Area: 8120.14m²
Story: 16
L/W Ratio: 2.4
Height: 64 m
Member Cost: 485868.47 €

Design 15
Total Floor Area: 8304.53m²
Story: 16
L/W Ratio: 2.1
Height: 64 m
Member Cost: 486580.97 €

Figure 5: Design alternatives generated from parametric input.

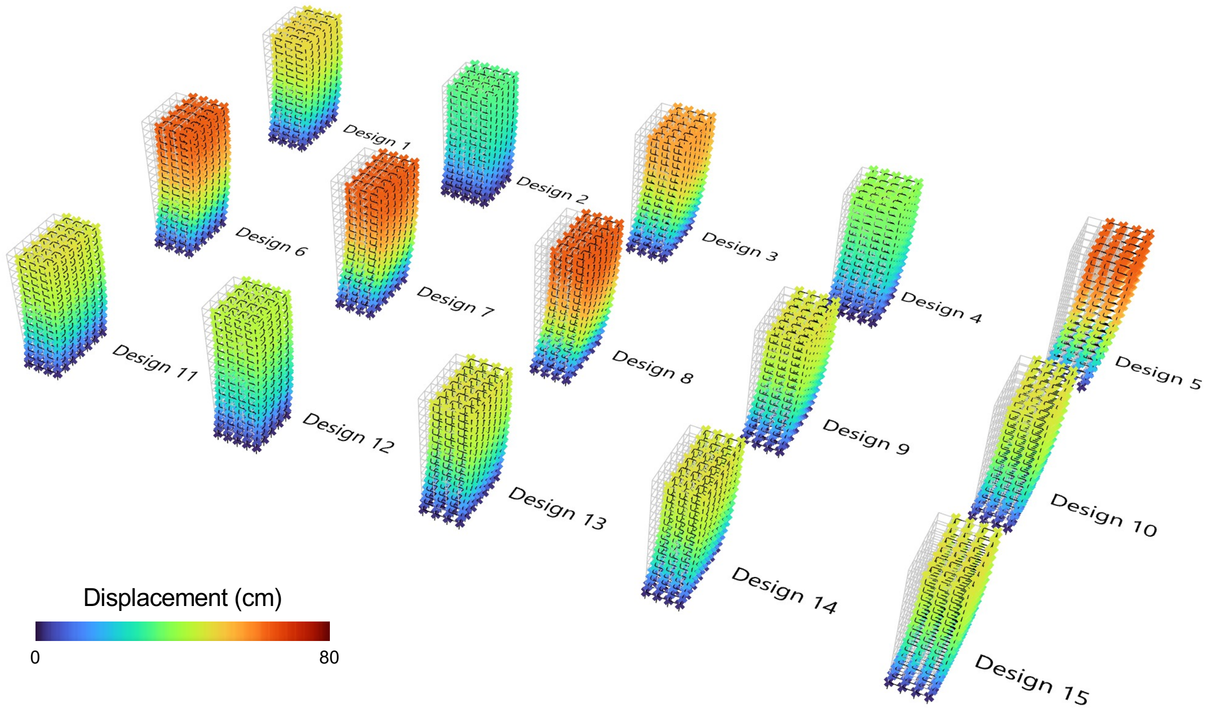


Figure 6: Predicted displacement visualization for building alternatives.

Together, these figures demonstrate the potential of the GNN-powered interface as both a learning tool and a generative design assistant. It supports real-time exploration, encourages structural reasoning, and helps bridge the gap between intuitive design processes and structural performance evaluation, all within an interactive and visually integrated environment.

4. DISCUSSION

This section reflects on the model's performance and its broader implications. It discusses the strengths of the GNN-based surrogate in learning structural behavior and generalizing to unseen configurations, responds to questions around the role of interpolation, and considers potential limitations related to dataset structure and model fit. Together, these insights help position the proposed approach within the context of both practical design workflows and future research directions.

4.1 Learning structural behavior

The GNN model developed in this study learns structural behavior by simulating how information propagates through connected components. Unlike tabular models, which treat each input feature independently, the GNN leverages its message-passing architecture to aggregate information from neighboring nodes and edge attributes. This structure-aware mechanism allows the model to capture force flow and relational dependencies that define structural performance.

In conventional ML systems, such relationships are typically lost or need to be encoded manually. In contrast, the graph-based representation enables the model to reason directly about geometry, connectivity, and load paths. This encoding aligns naturally with physical principles and provides a robust foundation for generalization. The trained model is not constrained by fixed feature formats or layout assumptions, making it flexible across various design scenarios. This capability is essential when evaluating early-stage building designs where geometry and topology can vary widely. The model offers a reusable and scalable surrogate for structural simulation that operates without relying on time-intensive FEM analyses, especially useful when rapid evaluation is needed.

4.2 Beyond simple interpolation

While it is reasonable to ask whether a simple interpolation approach could replace the need for a trained model, particularly given the size of the synthetic dataset, this strategy is limited in important ways. First, the design space for building structures is inherently complex, involving nonlinear interactions between geometry, material properties, and load conditions. These relationships cannot be reliably captured by local interpolation alone.

Second, structural design practices vary significantly across countries due to differing building codes and safety standards. A model that relies purely on interpolation would be constrained by the scope of its training data and unlikely to adapt to varied regulatory environments. In contrast, the GNN is trained to learn the underlying structural behavior, rather than just filling gaps between known samples. This allows it to generalize beyond the specific configurations seen during training and make physically plausible predictions even when extrapolating to previously unseen building types.

That said, it is important to acknowledge that the current model has been trained on data generated under a specific set of assumptions aligned with Eurocode. While the GNN architecture is well-suited for adaptation, further evaluation will be necessary to assess its performance under differing national regulations and design conventions.

By embedding structural logic into the model architecture, this approach offers a transferable, data-efficient alternative to traditional simulation methods. It supports scalable integration into diverse design contexts where predefined data coverage cannot be guaranteed. As part of future work, extending the training data to include varied code requirements or applying domain adaptation techniques could help validate and broaden the model's cross-regional applicability.

4.3 Model fit and limitations

The GNN model demonstrates consistent and stable learning behavior across all predicted outputs, as indicated by the close alignment of training and in-distribution test error curves for displacement, shear force, and bending moment. This pattern suggests that the model effectively learns generalized structural patterns from the training data without overfitting to specific configurations. The smooth and gradual convergence observed across these tasks also indicates that the chosen architecture and graph-based feature representation are well-suited to capturing a wide range of structural response behaviors across varied design scenarios.

However, it is important to note that both the training and in-distribution test datasets are generated through the same automated design pipeline. While this ensures consistency and control over input variability, it may also limit the diversity of structural forms encountered during training. As a result, the performance of in-distribution test set may not fully capture the model's robustness when applied to more irregular or unconventional building configurations. Although out-of-distribution tests demonstrate strong generalization, further work is needed to investigate performance under extreme parameter combinations. Applying techniques such as transfer learning or curriculum learning may help adapt the model to new domains or progressively more complex design scenarios.

Another consideration involves the interpretability of the model. While the GNN architecture captures structural relationships, it functions as a black-box model from the user's perspective. This may limit adoption in educational or regulatory contexts. Exploring explainable AI (XAI) techniques, such as attention visualization or sensitivity analysis, could enhance transparency and provide users with interpretable justifications for predicted structural responses.

5. CONCLUSION

This research presents a graph-based surrogate modeling framework for predicting nodal structural responses in multi-story building structures under static wind loads. By encoding buildings as graphs and applying a Graph Neural Network (GNN) architecture, the model captures key structural relationships and load propagation behaviors across a wide range of configurations. The results show that the model performs well not only on training and in-distribution test data but also when extrapolated to unseen structural variants, demonstrating its potential as a scalable and generalizable surrogate for linear static analysis.

A key strength of this approach lies in its ability to provide accurate predictions with minimal computational cost. Once trained, the GNN enables rapid structural evaluation across large datasets, making it especially useful in early-stage design workflows where speed and flexibility are essential. Compared with traditional FEM simulations and rule-based estimations, the surrogate model offers a more data-driven and general solution.

Beyond its technical performance, the model has been integrated into a Rhino-Grasshopper workflow to support live feedback during conceptual design. This allows novices to explore structural alternatives interactively, helping to build structural intuition and support more informed early-stage decisions. While the current implementation focuses on static wind loading, the framework offers a foundation for future extensions to more complex loading scenarios and differing national regulations. These extensions will further broaden the model's educational and practical relevance.

ACKNOWLEDGEMENTS

The research presented is supported by the TUM Georg Nemetschek Institute Artificial Intelligence for the Built World.

REFERENCES

Arciszewski, T., & Lakmazaheri, S., (2001). Structural design education for the 21st century, *International Journal of Engineering Education* 17(4), 446–454.

Boulanger, S., & Smith, I., (2001). Multi-strategy workspace navigation for design education, *Design Studies* 22(2), 111–140.

Horikx, M. P., (2022). Conceptual structural design: Understanding structural performance.

D'Acunto, P., Ohlbrock, P.O. & Pawlitschko, R., (2024). Conceptual design of structures. Birkhäuser.

Kazemi, P., Turrin, M., Andriotis, C., Entezami, A., Ghisi, A., & Mariani S., (2023). AI-assisted generative workflow for the early-stage design of sustainable tall buildings based on their structural behaviour, in *Advanced Building Skins 2023.*

Thai, H.-T., (2022). Machine learning for structural engineering: A state-of-the-art review, Structures, 38, p. 448–491.

Kazemi, P., Turrin, M., Andriotis, C., Entezami, A., Ghisi, A., & Mariani S., (2024). Deep learning approach to structural performance prediction of tall buildings in seismic areas, in *Proceedings of the IASS 2024 Symposium Redefining the Art of Structural Design.*

Battaglia, P.W., Hamrick, J.B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., et al., (2018). Relational inductive biases, deep learning, and graph networks. arXiv:1806.01261

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., & Sun, M., (2020). Graph neural networks: A review of methods and applications, *AI Open*, 1, p. 57-81.

Chang, K.-H., & Cheng, C.-Y., (2020). Learning to simulate and design for structural engineering, in *Proceedings of the 37th International Conference on Machine Learning*.

Bleker, L., Pastrana, R., Ohlbrock, P.O., & D'Acunto, P., (2023). Structural form-finding enhanced by graph neural networks, in Gengnagel, C., Baverel, O., Betti, G., Popescu, M., Thomsen, M.R., & Wurm, J. (eds) Towards Radical Regeneration. DMS 2022. Springer, Cham.

Bleker, L., Tam, K.M.M., & D'Acunto, P., (2024). Logic-informed graph neural networks for structural form-finding. *Advanced Engineering Informatics*, 61.

Whalen, E., & Mueller, C., (2022). Towards reusable surrogate models: Graph-based transfer learning on trusses. *Journal of Mechanical Design*, 144(2).

Fey, M., & Lenssen, J. E., (2019). Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*.

Kingma, D. P., & Ba, J., (2017). Adam: A Method for Stochastic Optimization. arXiv:1412.6980.

CSI, (2002). SAP 2000 analysis reference manual. Computers and Structures Inc., Berkeley.

British Standards Institution (BSI), (2005). Eurocode 1: Actions on structures - Part 1-4: General actions - Wind actions.

German Institute for Standardization (DIN), (2010). German version EN 1991-1-4:2005 + A1:2010 + AC:2010.

McNeel, R., & others, (2024). Rhino 8. Robert McNeel & Associates, Seattle, WA.