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

Fast Prediction of Stress Distribution

A GNN-based surrogate model for unstructured mesh FEA

Jiaqian WU*, Chaoyu DU, Benjamin DILLENBURGER, Michael A. KRAUS[a, b]

*Department of Architecture, ETH Zurich
Stefano-Franscini-Platz 1, 8093 Zurich, Switzerland
jiaqian.wu@arch.ethz.ch

[a] Department of Civil, Environmental and Geomatic Engineering, ETH Zurich, Switzerland
[b] Department of Civil Engineering and Geodesy, TU Darmstadt, Germany

Abstract

Structural design is underpinned by stress distribution analysis, typically performed using Finite Element Analysis (FEA) to simulate complex physical phenomena. However, the computational demands of high-fidelity simulations can be prohibitive, hindering iterative design optimization workflows. Recent advances in deep learning provide a promising alternative through the training of surrogate models to approximate FEA responses, thus bypassing numerical solution methods. While grid-based surrogate models have shown efficacy in this area, their dependence on fixed resolutions limits their ability to handle irregular geometries. To overcome this challenge, we propose a novel framework utilizing graph neural networks (GNNs), allowing for a flexible representation of non-uniform geometries and mechanical conditions. Our approach encompasses the entire pipeline, from data synthesis to model training. It contributes to an extendable dataset with variations in geometries, loads, and boundary conditions of steel plates with holes. The surrogate model, employing a specialized MESHGRAPHNETS architecture, adeptly captures the complex underlying physics, enabling rapid prediction of stress distribution for unseen cases. The results demonstrate the model's efficiency and accuracy in accelerating unstructured mesh FEA, with generalizability across shapes, mesh resolutions, and mechanical settings.

Keywords: stress distribution, graph neural network, surrogate model, unstructured mesh, finite element analysis, deep learning

1. Introduction

Accurate prediction of physical responses underpins engineering applications across various domains. Finite Element Analysis (FEA), a well-established physics-based modeling method, has proven to be a reliable tool for numerically solving underlying partial differential equations (PDEs) and remains the industry standard for simulating complex physical phenomena, including solid and fluid mechanics, material science, and aerodynamics. However, high-fidelity FEA simulations can be computationally intensive (Bolandi et al. [1]) and reliant on expert participation (Nath et al. [2]), specific software or platforms, posing challenges to real-world applications, particularly iterative design optimization workflows (Cao et al. [3]) that require numerous simulations with varying parameters in real-time.

To overcome these limitations, deep learning (DL) techniques offer a promising avenue through surrogate models for the regression of FEA solutions. Since FEA-based simulations can be conducted on either regular grids or unstructured meshes, DL-based models such as convolutional neural networks (CNNs) and graph neural networks (GNNs) have gained traction for their ability to handle these two types of FEA representations seamlessly. Compared with fixed-resolution grids, mesh-based

representation excels in modeling angled or curved geometries, where vertices in a mesh can be allocated unevenly, thus budgeting in important regions that contain complex shapes or require higher precision. Within the context of GNNs, finite elements and their physical properties can be naturally represented as nodes in the graph, with connectivity as edges.

This study aims to explore the feasibility and challenges of training and utilizing a GNN-based surrogate model for accelerating unstructured mesh FEA. Specifically, we focus on predicting the stress distribution in 2D steel plates with holes under various load conditions – a ubiquitous structural component across various industries.

Our work makes three key contributions:

- We integrate parametric modeling with FEA to generate a large-scale dataset, which accommodates variations in geometries, mesh resolutions, and mechanical configurations, allowing for the training of robust and generalizable models.
- We propose a tailored MESHGRAPHNETS architecture optimized for accelerating FEA, particularly for stress distribution in thin plates with holes.
- We demonstrate the effectiveness of our method by applying it to predict stress distribution in unseen cases and employing evaluation metrics to assess the model's predictive accuracy, providing insights into its potential for real-world applications.

2. Related work

Deep learning presents significant advantages for surrogate models in FEA, offering their ability to approximate high-dimensional non-linear relationships (Tang et al. [4]) and achieve computational efficiency. In stress analysis, DL-based surrogate models can be broadly categorized into two prominent approaches depending on how input data are represented: grid-based and mesh-based approaches.

2.1. Grid-based surrogate models

Grid-based models excel at handling Euclidean input, such as images, utilizing DL techniques like generative adversarial networks (GANs) and CNNs. These models have been successful on data with an underlying grid-like structure (Bronstein et al. [5]) due to their inherent inductive biases, like translation invariance, weight sharing, and locality. These properties align with the demands of FEA using regular grids and continuum-based models (Gulakala et al. [6]).

Grid-based models have proven effective in various applications, including stress field prediction using CNNs (Rezasefat and Hogan [7]) and conditional GANs in solid structures (Hoq et al. [8], Nie et al. [9], Jiang et al. [10]), U-Net for mechanical responses in microstructures (Khorrami et al. [11]). However, grid-based surrogate models have inevitable limitations. Their dependence on a fixed resolution requires identical input sizes, posing challenges for domains with variable lengths. Additionally, grid-based models are restricted to structured rectangular domains (typically 2D quadrilateral or 3D hexahedron meshes with uniform side lengths). Such regularity of grids lacks flexibility in conforming to complex geometries with curved boundaries. These limitations make grid-based models less suitable for scenarios involving unstructured meshes, which are commonly adopted across various engineering disciplines. Consequently, this necessitates the exploration of alternative approaches like mesh-based surrogate models.

2.2. Mesh-based surrogate models

Modeling mesh-based physical processes encompasses structured (Fu et al. [12]) and unstructured (Pfaff et al. [13], Sanchez-Gonzalez et al. [14]) meshes. Unstructured meshes feature irregular arrangements of non-uniform cells, enabling adaptive resolutions within the computational domain. This flexibility allows FEA simulations using unstructured meshes to achieve superior accuracy, particularly in regions with sharp gradients, and achieve more cost-efficient performance, thus mitigating the limitations of grid-based models.

Harnessing intuitive mesh-to-graph mapping, GNNs are commonly employed in mesh-based surrogate models owing to their capability to process non-Euclidean data structures, inheriting all the advantages

of unstructured meshes (Maurizi et al. [15]). Unlike traditional neural networks, GNNs are inherently independent of resolutions without imposing a predetermined tensor size as input. Instead, data structures are typically represented as a set of nodes and edges, where nodes denote entities, and edges describe relationships or connectivity between them. This intrinsic ability to handle unstructured meshes positions GNNs particularly well-suited for FEA applications. The crux of GNN operation lies in Message Passing (MP), a process where information is propagated and aggregated throughout the graph. This allows GNNs to effectively capture the intricate relationships and dynamic changes within the mesh.

GNNs have found success across a myriad of applications, serving as the cornerstone for surrogate models in various domains. These applications range from urban wind field prediction (Shao et al. [16]), soft-tissue mechanics (Dalton et al. [17]), fracture and stress evolution modeling (Perera et al. [18]), and physical dynamics simulations (Lino et al. [19], Cao et al. [3]). These examples showcase the adaptability and efficacy of GNNs in diverse scenarios, reinforcing their role as a potent tool for processing complex, unstructured data representations. In the realm of stress analysis, graph-based frameworks have emerged to predict physical fields based on loading and boundary conditions (Gulakala et al. [6]), or to establish connections between material properties and physical responses (Maurizi et al. [15]). Nevertheless, current models often exhibit limitations in generalizability (Gulakala et al. [6]). The development of models capable of handling a broader spectrum of geometries, loads, and boundary conditions remains an active area of research. This necessitates the creation of more diverse datasets and the continued exploration of more generalizable surrogate models.

3. Methodology

This section details our proposed pipeline from synthesizing datasets to building surrogate models for predicting stress distribution in FEA simulations. Specifically, our focus lies on two-dimensional plates with holes exhibiting linear elastic behavior typical of common steel alloys.

3.1. Data generation

The publicly available SimuStruct dataset (Ribeiro et al. [20]) offers a valuable starting point, which consists of 1000 samples of plates with six circular holes under consistent loading and boundary conditions. However, this dataset is insufficient for training a GNN capable of predicting the structural behavior of plates across diverse geometric and mechanical configurations. To address this limitation, our pipeline starts by expanding upon the SimuStruct dataset significantly by creating a diverse collection of steel plates containing holes. These samples exhibit variations in the plate's dimensions, holes' number, sizes, and positions, as well as the external loads and boundary conditions applied. This parametric model is the foundation for generating mesh representations and training adaptable surrogate models. By adjusting the predefined values or generative ranges of parameters, the model can be seamlessly extended to encompass a broader range of geometries and conditions for future applications.

3.1.1. Parametric modeling

To diversify the dataset and capture a wide range of scenarios, we employed parametric modeling to generate a large number of synthetic two-dimensional plates with varying geometries, loading conditions, and boundary constraints. This approach allows us to systematically explore the design space and create a comprehensive dataset for training and evaluating our GNN-based surrogate model.

Figure 1 illustrates the process of creating this synthetic dataset in Rhino and Grasshopper, with all steps being parametrically controlled:

1) Starting from the shape of the rectangular plate, we incorporate holes of different sizes and randomly distributed placements.
2) The input surface is discretized into an unstructured, triangular mesh with varying resolutions, with options for adaptive remeshing based on curvature and proximity to holes, ensuring an accurate representation of complex geometries.
3) Boundary conditions are set as free or pinned, providing versatility in constraining the plate's edges.

4) Loading conditions include axial and uniaxial loads, with magnitudes varying from 5 to 30 N/m.
5) The material properties remain constant, representing a linear-elastic, isotropic material with a Young's modulus of 210 GPa and a Poisson's ratio of 0.3. While this simplification enables efficient exploration of geometric and mechanical variations, future work could incorporate different material properties to expand the dataset further.
6) Steps 2–5 allow us to construct the FEA model using Karamba3D (Preisinger [21]) as the finite element solver. The model yields numerical solutions of von Mises stress fields within the plates.
7) Finally, we exported geometric information, model details, and numerical results as CSV files for training and evaluation.

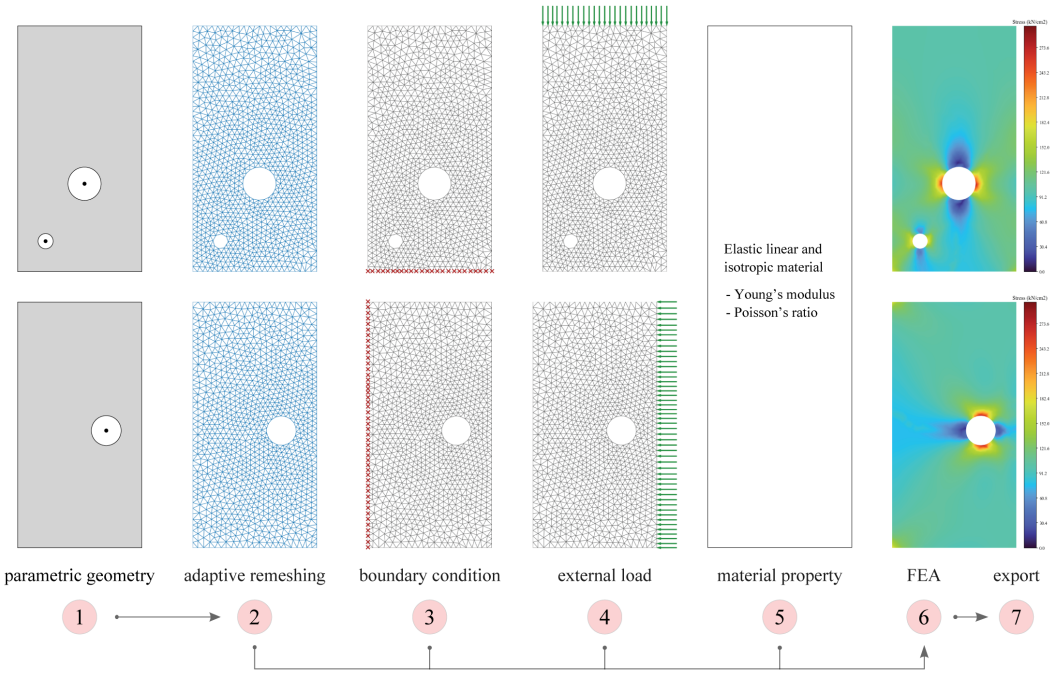

Figure 1: Parametric pipeline for the synthetic dataset

3.1.2. Data structure

Our dataset comprises 29,000 samples of unique mesh configurations. For each configuration, we define:

- *H*: Number of holes in the plate.
- *N*: Number of vertices in the mesh.
- *C*: Number of faces (namely, cells or elements) in the mesh.

Each sample within the dataset corresponds to a specific mesh and mechanical configuration, and includes six key pieces of information, as shown in Figure 2:

- *Circular holes* (shape: [*H*, 3]): This data describes the geometry of the holes in the plate using three values per hole (the 2D coordinates of the center and its radius).
- *Mesh geometry* (shape: [*N*, 2]): This describes the 2D coordinates of each vertex in the mesh.
- *Mesh topology* (shape: [*C*, 3]): This information defines the connectivity of the mesh by specifying the indices of vertices associated with each face. It can be used to derive the edges between vertices.
- *Boundary conditions* (shape [*N*, 1]): This specifies the type of boundary conditions applied to each vertex. A value of 0 indicates a free vertex, and 1 signifies a pinned vertex.
- *External loads* (shape: [*N*, 2]): This represents the external forces applied to each vertex in the x and y directions. Zero entries indicate no load.
- *von Mises stress* (shape: [*N*, 1]): This represents the von Mises stress value calculated at each vertex in the mesh.

This comprehensive data structure enables the reconstruction of the mesh geometry and the visualization of the corresponding FEA simulation results.

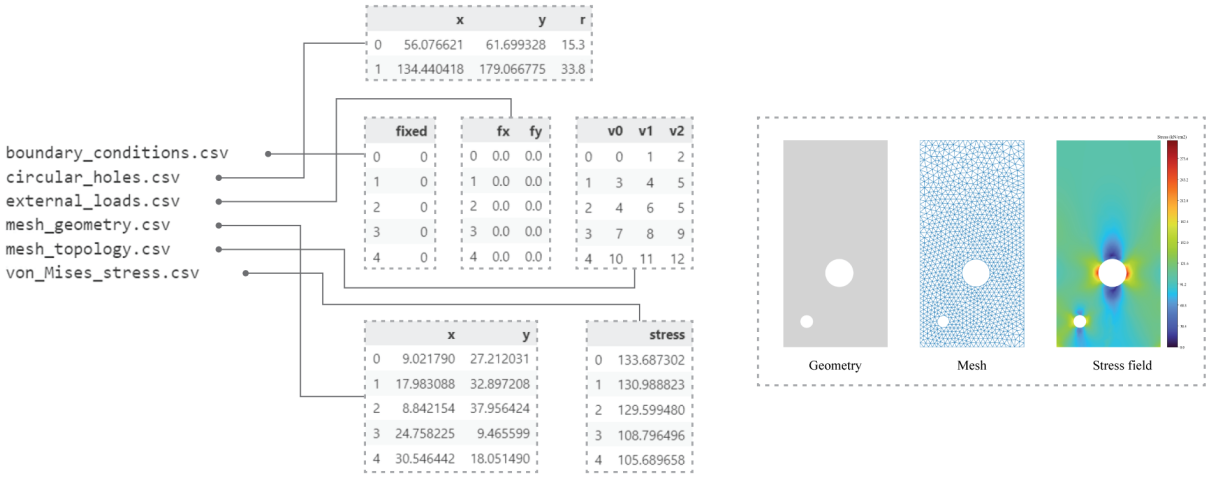

Figure 2: Data structure for each sample in the dataset

3.2. Surrogate model

Unstructured meshes can be naturally converted into graph representation, making graph neural networks a compelling solution for tackling FEA problems. The mesh data, including vertices and faces, can be effectively mapped onto a graph structure, with relevant attributes assigned to nodes, edges, and the entire graph itself. Our proposed surrogate model leverages MESHGRAPHNETS (Pfaff et al. [13], Kanfar [22]), a versatile GNN architecture specifically designed to handle unstructured mesh data. This architecture has demonstrated success in various physical systems simulated on meshes, ranging from structural mechanics over cloth simulations to fluid dynamics.

3.2.1. Mesh-to-graph representation

We represent the mesh domain as a computational graph denoted by $G = (V, E)$, where V is a set of N nodes (vertices) connected through M edges (E). The raw mesh data is transformed into node attributes, edge attributes, node outputs (target labels for prediction), and visualization data. The features extracted from the dataset are as follows:

- *Node features* (shape: [N, 5]): This combines geometric and mechanical information for each node, including its 2D coordinates, boundary condition, and the external load applied to it.
- *Edge index* (shape: [2, M]): This captures the graph's connectivity by storing the indices of bi-directional connected nodes for each edge within a matrix.
- *Edge attributes* (shape: [M, 3]): This assigns a 2D vector for each edge, representing the direction from the start node to the end node, along with its vector norm.
- *Node output* (shape: [N, 1]): This represents the von Mises stress value at each node, serving as the target label for the prediction model.

While additional data, like information on cells and holes, is stored for visualization purposes, it is not directly involved in the embedding process or the training itself.

3.2.2. GNN architecture

As depicted in Figure 3, the surrogate model adopts the encode–process–decode paradigm used in MESHGRAPHNETS (Battaglia et al. [23], Pfaff et al. [13]). The encoding stage consists of separate node and edge encoders. These encoders employ a multilayer perceptron (MLP) to transform the raw input data (node features and edge attributes) into a format conducive to processing by GNNs. The processing stage is the model's core and involves message passing, aggregation, and updating phases, wherein the encoded information is iteratively propagated through stacked MP layers. This process captures the intricate relationships and dependencies within the mesh data. Finally, the decoder takes the processed

information and translates it into the desired output – the predicted von Mises stress distribution across all mesh nodes.

The building blocks of the encoder, processor, and decoder are ReLU-activated MLPs with two hidden layers. Both the number of processor layers and output size for these MLPs are set to 32, except for the decoder's output layer, which is adjusted to match the dimension of prediction. All MLPs incorporate residual connections to mitigate the vanishing gradient problem, while all outputs (except the decoder's). are normalized using LayerNorm.

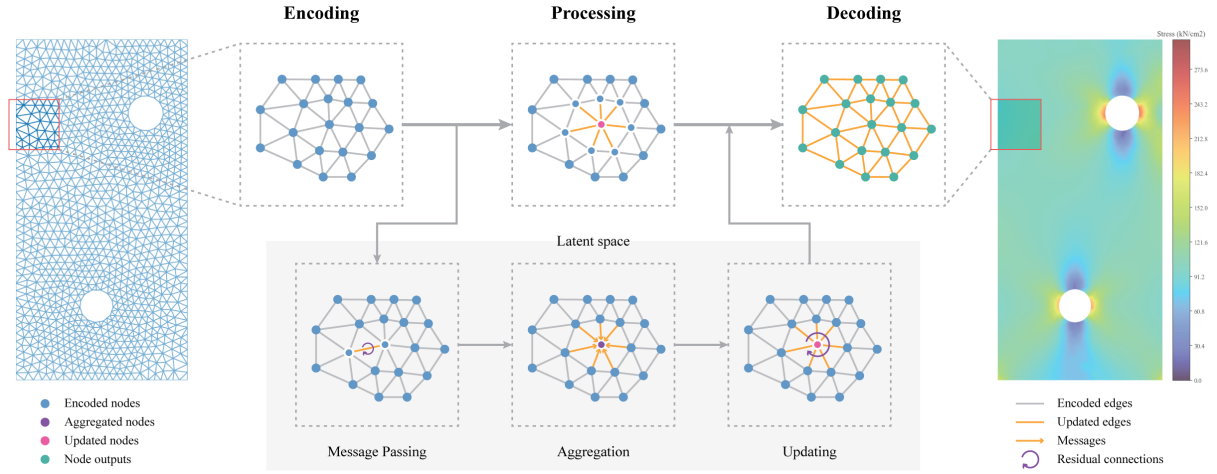

Figure 3: Schematics of the GNN-based surrogate model

3.2.3. Performance evaluation

We employ a comprehensive suite of metrics to evaluate the surrogate model's effectiveness. During training, the root mean squared error (RMSE) serves as the loss function. This metric quantifies the average difference between the model's predictions and the actual stress values obtained from the FEA simulations considered the ground truth. In our case, RMSE is defined as:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(s_i - \hat{s}_i)^2}\,, \tag{1}$$

where s_i denotes the actual stress value at node i (ground truth) computed by the FEA solver, $\hat{s}_i$ is the corresponding predicted stress by the surrogate model, and n is the total number of nodes across all samples.

For the test process, we utilize the R-square (R^2) metric to assess the proportion of variance in the actual stress values that can be explained by the model's predictions. R^2 ranges from 0 to 1, with values closer to 1 indicating a better fit and stronger explanatory power. It is defined as:

$$\mathrm{R}^2 = 1 - \frac{\sum_{i=1}^{n}(s_i - \hat{s}_i)^2}{\sum_{i=1}^{n}(s_i - \bar{s})^2}\,, \tag{2}$$

where $\bar{s}$ is the average of the actual stress values (s_i) across all nodes.

Furthermore, we leverage the cumulative distribution function (CDF) of prediction error percentage to gain deeper insights into the distribution of prediction errors across the test set. The error percentage ε_i at each node is calculated as:

$$\varepsilon_i = \frac{|s_i - \hat{s}_i|}{s_i} \times 100\% \tag{3}$$

By analyzing this set of metrics (RMSE, R^2, and CDF of error percentage), we obtain a clearer picture of the model's accuracy and its ability to capture the underlying relationships within the stress distribution data.

4. Results

The GNN experiments were implemented using PyTorch and PyG libraries on a Windows system equipped with an AMD Threadripper PRO 5975WX 32-Core processor, NVIDIA RTX A5000 GPU, and 256GB of RAM.

To expedite the experimentation process, we partitioned a portion of the synthetic dataset into the training, validation, and test sets. The training set contained 23,000 samples, while the validation and test sets held 3,000 samples each. We employed the Adam optimizer to accelerate model convergence, with a cosine annealing scheduler for the learning rate (2.5×10^{-4}). Hyperparameter tuning was performed but the model exhibited minimal sensitivity, culminating in the best validation RMSE of 0.04615 (Figure 4). Subsequently, the trained model was evaluated on unseen data from the test set to assess its generalization capability.

Figure 5 presents a selection of predicted stress fields alongside their corresponding ground truth values from the test set. The error at each node is visualized using a color scheme: red indicates positive errors (predicted higher than the actual stress), and blue signifies negative errors. This allows for a quick visual assessment of the model's prediction accuracy across different mesh configurations.

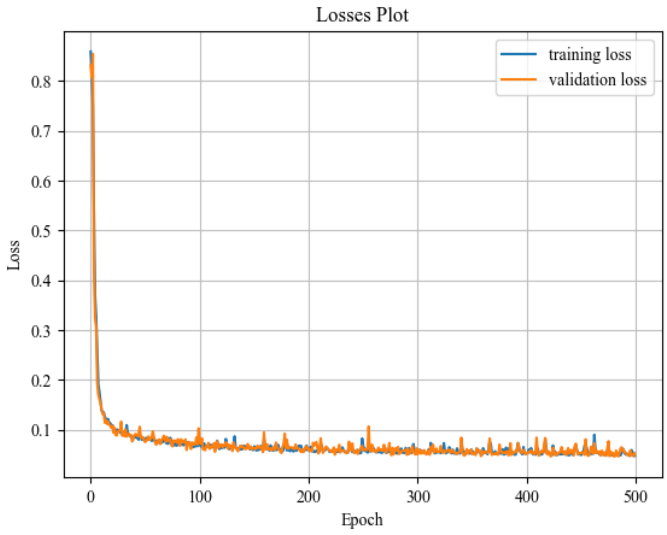

Figure 4: Losses plot of the training process

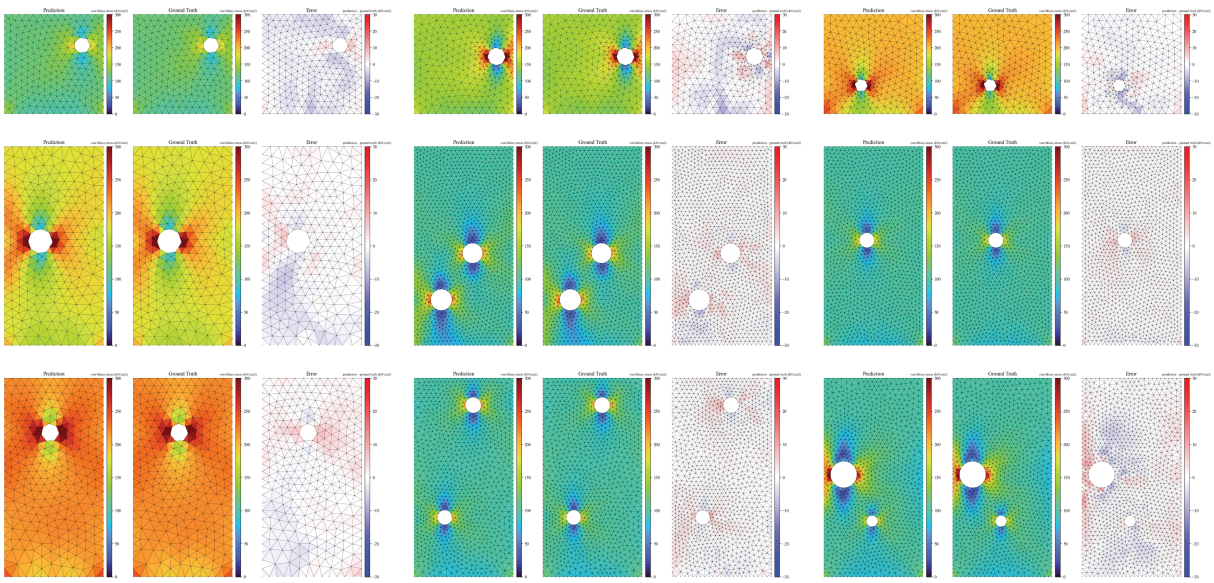

Figure 5: Predicted stress fields, ground truths, and errors of test samples

As further evidence of the model's performance, Figure 6 combines a 45° regression plot (indicating a strong correlation between predicted and actual stress values) with the distribution of stress values from the test set. This plot reveals an R-squared score of 0.997 and an RMSE of 0.05683. Furthermore, Figure 7 showcases the CDF of the error percentage, highlighting that 95% of the predictions fall within a narrow band of 4.5% error.

To assess computational efficiency, we compared the execution times of both methods on a Windows system with an Intel Xeon Gold 5412U processor and 32GB RAM. For a fair benchmark, both the surrogate model and FEA solver relied solely on CPU computation for all test cases. The surrogate model demonstrably outperformed the FEA solver in speed, averaging only 48 milliseconds per prediction, while Karamba3D required an average of 84 milliseconds. In practice, this efficiency gap is expected to widen even further when leveraging the power of GPUs for computation.

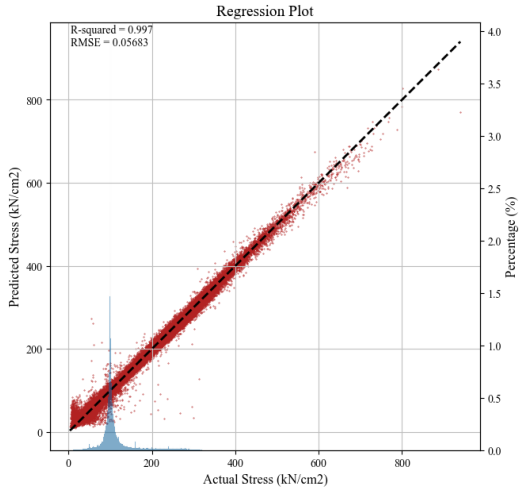

Figure 6: 45° regression plot of stress prediction

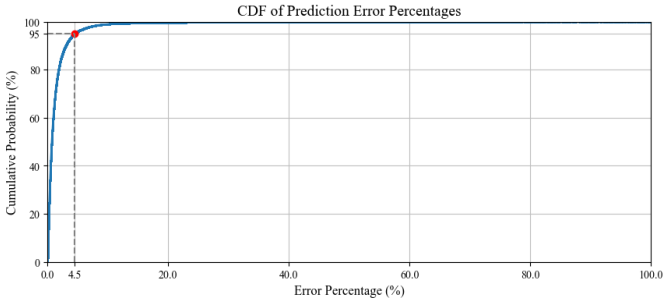

Figure 7: CDF of prediction error percentage

5. Discussion

The proposed GNN-based surrogate model achieves commendable performance, exhibiting a robust fit and remarkable generalizability across diverse mesh resolutions, plate dimensions, hole arrangements, and mechanical conditions. The high R-squared score and low RMSE (Figure 6) collectively indicate its high accuracy and low average error. The CDF of prediction error percentage (Figure 7) underscores the model's reliability in approximating stress fields. These results showcase the efficacy of our model in capturing the complexities of stress distribution within thin plates with holes. The observed speed advantage of the model stems from its generic architecture and reduced computational requirements compared to traditional FEA solvers, which need to solve complex partial differential equations. Notably, this superiority is likely to become even more pronounced when dealing with increasingly intricate conditions.

While the model exhibits strong overall performance, a closer examination of Figure 6 reveals a concentration of outliers with larger errors at both ends of the stress spectrum. Specifically, the model's accuracy decreases in regions experiencing exceptionally high or low stress values compared to the intermediate range. This phenomenon can be attributed to two key factors. Firstly, the inherent bias within the synthetic dataset skews towards extreme values, leading the model to prioritize learning mid-range patterns during the training process. Secondly, the application of graph convolution via stacked MPs might contribute to over-smoothing (Cao et al. [3]). Consequently, the model's predictive accuracy in regions with either very low or very high stress is compromised. It is crucial to acknowledge that these areas with extreme values often hold heightened significance in practical engineering applications and are thus expected to be predicted with higher accuracy. For example, pinpointing areas of peak stress concentration is paramount for evaluating structural integrity.

6. Conclusion

This study demonstrates the effectiveness of using a mesh-to-graph representation coupled with a graph neural network for predicting stress distribution in thin plates. The GNN architecture, leveraging the power of graph convolution and message passing, can effectively capture the relationship among structural and geometric information encoded within unstructured meshes, enabling accurate stress prediction.

Our approach begins with an extendable pipeline for dataset generation. This framework allows for the creation of comprehensive datasets featuring diverse geometries, boundary conditions, and external loads, all derived from parametric modeling and FEA. Additionally, we introduce a customized MESHGRAPHNETS model tailored for stress distribution prediction and validate its generalizability in accommodating various shapes, mesh resolutions, and mechanical configurations. Notably, the superior speed of our model makes it suitable for real-time applications demanding rapid responses.

However, this research also identifies a key limitation concerning stress prediction in regions with extreme values. Larger errors were observed in these areas, impeding the model's ability to predict stress in critical locations reliably. While the surrogate model performs well for moderate values, future efforts should focus on enhancing its sensitivity to extreme stress conditions.

Potential avenues for improvement include implementing data augmentation techniques such as oversampling or undersampling, engineering cost functions to prioritize accurate predictions in extreme stress regions, and exploring alternative GNN architectures. Utilizing graph attention networks or multiscale GNNs shows promise in facilitating more accurate and informative message passing within the graph. By addressing these nuances, the surrogate model can be refined to provide more dependable predictions across the entire stress spectrum, ultimately leading to its wider adoption in real-world engineering scenarios.

Acknowledgments

We would like to thank Dr. Danielle Griego, Vera Balmer, Hang Zhang, and our reviewers for their valuable discussions, support, and feedback on this work and manuscript.

References

[1] H. Bolandi, X. Li, T. Salem, V. N. Boddeti, and N. Lajnef, "Bridging finite element and deep learning: High-resolution stress distribution prediction in structural components," *Front. Struct. Civ. Eng.*, vol. 16, no. 11, pp. 1365–1377, Nov. 2022, doi: 10.1007/s11709-022-0882-5.

[2] D. Nath, Ankit, D. R. Neog, and S. S. Gautam, "Application of Machine Learning and Deep Learning in Finite Element Analysis: A Comprehensive Review," *Arch. Comput. Methods Eng.*, Mar. 2024, doi: 10.1007/s11831-024-10063-0.

[3] Y. Cao, M. Chai, M. Li, and C. Jiang, "Efficient Learning of Mesh-Based Physical Simulation with BSMS-GNN." arXiv, Jun. 18, 2023. doi: 10.48550/arXiv.2210.02573.

[4] M. Tang, Y. Liu, and L. J. Durlofsky, "A deep-learning-based surrogate model for data assimilation in dynamic subsurface flow problems," *J. Comput. Phys.*, vol. 413, p. 109456, Jul. 2020, doi: 10.1016/j.jcp.2020.109456.

[5] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: going beyond Euclidean data," *IEEE Signal Process. Mag.*, vol. 34, no. 4, pp. 18–42, Jul. 2017, doi: 10.1109/MSP.2017.2693418.

[6] R. Gulakala, B. Markert, and M. Stoffel, "Graph Neural Network enhanced Finite Element modelling," *PAMM*, vol. 22, no. 1, p. e202200306, 2023, doi: 10.1002/pamm.202200306.

[7] M. Rezasefat and J. D. Hogan, "A finite element-convolutional neural network model (FE-CNN) for stress field analysis around arbitrary inclusions," *Mach. Learn. Sci. Technol.*, vol. 4, no. 4, p. 045052, Dec. 2023, doi: 10.1088/2632-2153/ad134a.

[8] E. Hoq, O. Aljarrah, J. Li, J. Bi, A. Heryudono, and W. Huang, "Data-driven methods for stress field predictions in random heterogeneous materials," *Eng. Appl. Artif. Intell.*, vol. 123, p. 106267, Aug. 2023, doi: 10.1016/j.engappai.2023.106267.

[9] Z. Nie, H. Jiang, and L. B. Kara, "Stress Field Prediction in Cantilevered Structures Using Convolutional Neural Networks," *J. Comput. Inf. Sci. Eng.*, vol. 20, no. 1, p. 011002, Feb. 2020, doi: 10.1115/1.4044097.

[10] H. Jiang, Z. Nie, R. Yeo, A. B. Farimani, and L. B. Kara, "StressGAN: A Generative Deep Learning Model for 2D Stress Distribution Prediction," *J. Appl. Mech.*, vol. 88, no. 5, p. 051005, May 2021, doi: 10.1115/1.4049805.

[11] M. S. Khorrami *et al.*, "An artificial neural network for surrogate modeling of stress fields in viscoplastic polycrystalline materials," *Npj Comput. Mater.*, vol. 9, no. 1, Art. no. 1, Mar. 2023, doi: 10.1038/s41524-023-00991-z.

[12] X. Fu, F. Zhou, D. Peddireddy, Z. Kang, M. B.-G. Jun, and V. Aggarwal, "An FEA surrogate model with Boundary Oriented Graph Embedding approach." arXiv, Aug. 30, 2021. doi: 10.48550/arXiv.2108.13509.

[13] T. Pfaff, M. Fortunato, A. Sanchez-Gonzalez, and P. W. Battaglia, "Learning Mesh-Based Simulation with Graph Networks." arXiv, Jun. 18, 2021. doi: 10.48550/arXiv.2010.03409.

[14] A. Sanchez-Gonzalez, J. Godwin, T. Pfaff, R. Ying, J. Leskovec, and P. Battaglia, "Learning to Simulate Complex Physics with Graph Networks," in *Proceedings of the 37th International Conference on Machine Learning*, PMLR, Nov. 2020, pp. 8459–8468. Accessed: Mar. 21, 2024. [Online]. Available: https://proceedings.mlr.press/v119/sanchez-gonzalez20a.html

[15] M. Maurizi, C. Gao, and F. Berto, "Predicting stress, strain and deformation fields in materials and structures with graph neural networks," *Sci. Rep.*, vol. 12, no. 1, p. 21834, Dec. 2022, doi: 10.1038/s41598-022-26424-3.

[16] X. Shao, Z. Liu, S. Zhang, Z. Zhao, and C. Hu, "PIGNN-CFD: A physics-informed graph neural network for rapid predicting urban wind field defined on unstructured mesh," *Build. Environ.*, vol. 232, p. 110056, Mar. 2023, doi: 10.1016/j.buildenv.2023.110056.

[17] D. Dalton, D. Husmeier, and H. Gao, "Physics-informed graph neural network emulation of soft-tissue mechanics," *Comput. Methods Appl. Mech. Eng.*, vol. 417, p. 116351, Dec. 2023, doi: 10.1016/j.cma.2023.116351.

[18] R. Perera, D. Guzzetti, and V. Agrawal, "Graph neural networks for simulating crack coalescence and propagation in brittle materials," *Comput. Methods Appl. Mech. Eng.*, vol. 395, p. 115021, May 2022, doi: 10.1016/j.cma.2022.115021.

[19] M. Lino, C. Cantwell, A. A. Bharath, and S. Fotiadis, "Simulating Continuum Mechanics with Multi-Scale Graph Neural Networks." arXiv, Jun. 09, 2021. doi: 10.48550/arXiv.2106.04900.

[20] B. A. Ribeiro *et al.*, "SimuStruct: Simulated Structural Plate with Holes Dataset with Machine Learning Applications," presented at the Workshop on "'Machine Learning for Materials'" ICLR 2023, Apr. 2023. Accessed: Dec. 07, 2023. [Online]. Available: https://openreview.net/forum?id=s3tOuyR1vM7

[21] C. Preisinger, "Linking Structure and Parametric Geometry," *Archit. Des.*, vol. 83, no. 2, pp. 110–113, 2013, doi: 10.1002/ad.1564.

[22] R. Kanfar, "Learning Mesh-Based Flow Simulations on Graph Networks," Stanford CS224W GraphML Tutorials. Accessed: Dec. 07, 2023. [Online]. Available: https://medium.com/stanford-cs224w/learning-mesh-based-flow-simulations-on-graph-networks-44983679cf2d

[23] P. W. Battaglia *et al.*, "Relational inductive biases, deep learning, and graph networks." arXiv, Oct. 17, 2018. doi: 10.48550/arXiv.1806.01261.