

## Abstract

In the field of additive manufacturing engineering, the design and optimization of micro-structures subjected to various loading conditions has been increasingly represented using graph and constraint-based models. These representations facilitate the manipulation and exploration of structural topologies, enabling the optimization of micro-structures for desired performance characteristics. This research work investigates the application of Graph Neural Networks (GNNs) to learn the mapping from initial undirected graph representations, augmented with edge attributes, to their optimized topological solutions. Deep learning techniques are increasingly being applied in problems arising in scientific computing, and have shown promise in learning data to solution operators in important applications such as uncertainty quantification[1]. By employing the GNNs for the graph based data, this work aims to capture the intrinsic relationships and constraints inherent in structural designs. This is ongoing research work, and more results are in progress.

## Graph Neural Networks and Machine Learning

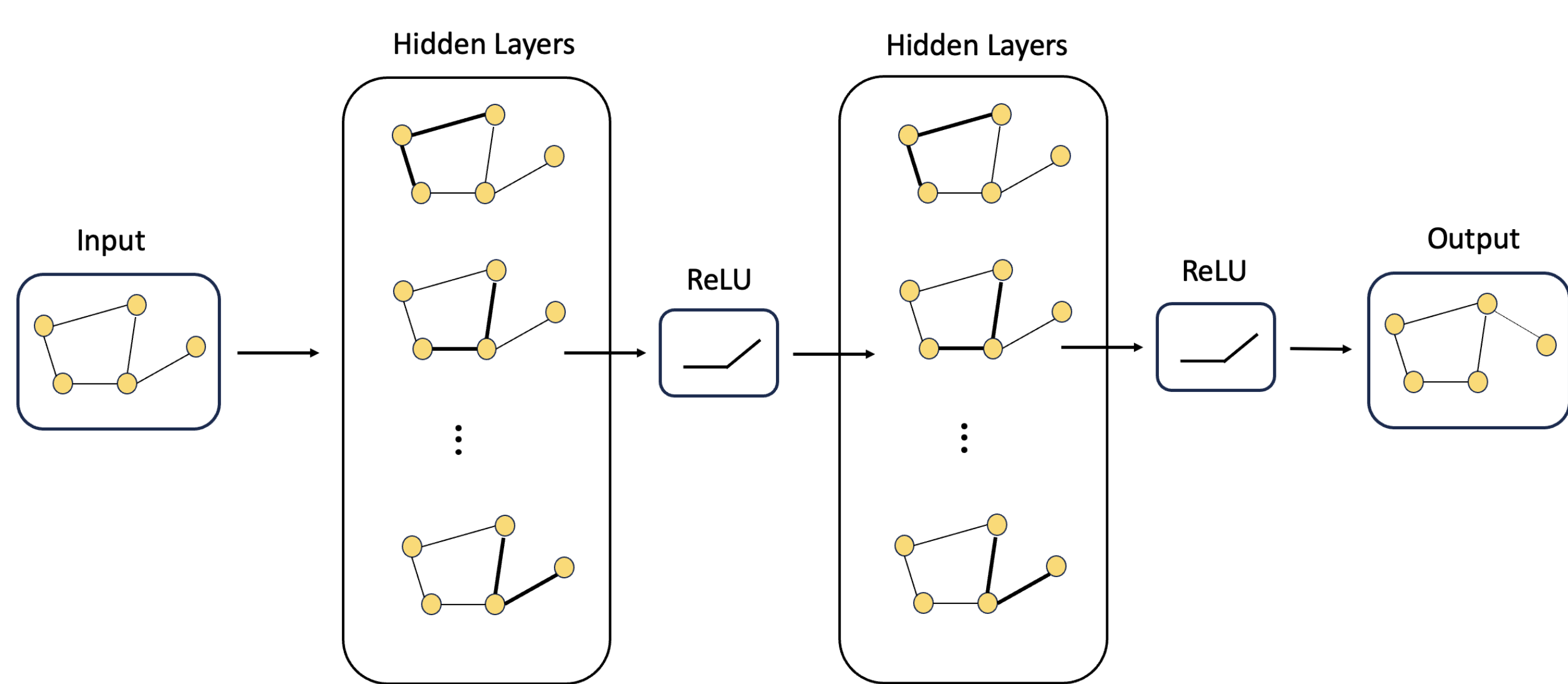


Figure 1: General Structure of Graph neural network

Let the initial structure be represented as a graph  $G_i = (V, E, F, R)$ , where:  $V$  represents the set of vertices in the graph,  $E$  edges between vertices and  $F, R \in \mathbb{R}^{|E|}$  represent the edge attributes. In this work we consider training a GNN as a surrogate for an expensive black box optimization routine mapping an initial Graph  $G$  [2] to an optimized graph  $G'_i = (V', E', F', R')$ , with minimal stress. A GNN transforms the initial graph to the optimized graph, represented as:

$$f_{\text{GNN}} : G \rightarrow G'$$

The optimization process is guided by an objective function  $J(G, G')$  that quantifies the optimality of  $G'$  based on specific criteria represented by constraints. The GNN model is trained to minimize  $J$  by adjusting its parameters  $\theta$ :

$$\min_{\theta} J(G, f_{\text{GNN}}(G; \theta))$$

## Challenges

- The design space for structural optimization is often high-dimensional.
- Scaling optimization to larger problems with traditional methods has many computational and memory constraints.
- The design space being discrete adds complexity to the process, making heuristic-based black box optimization even more challenging due to its slow convergence

## Schematics

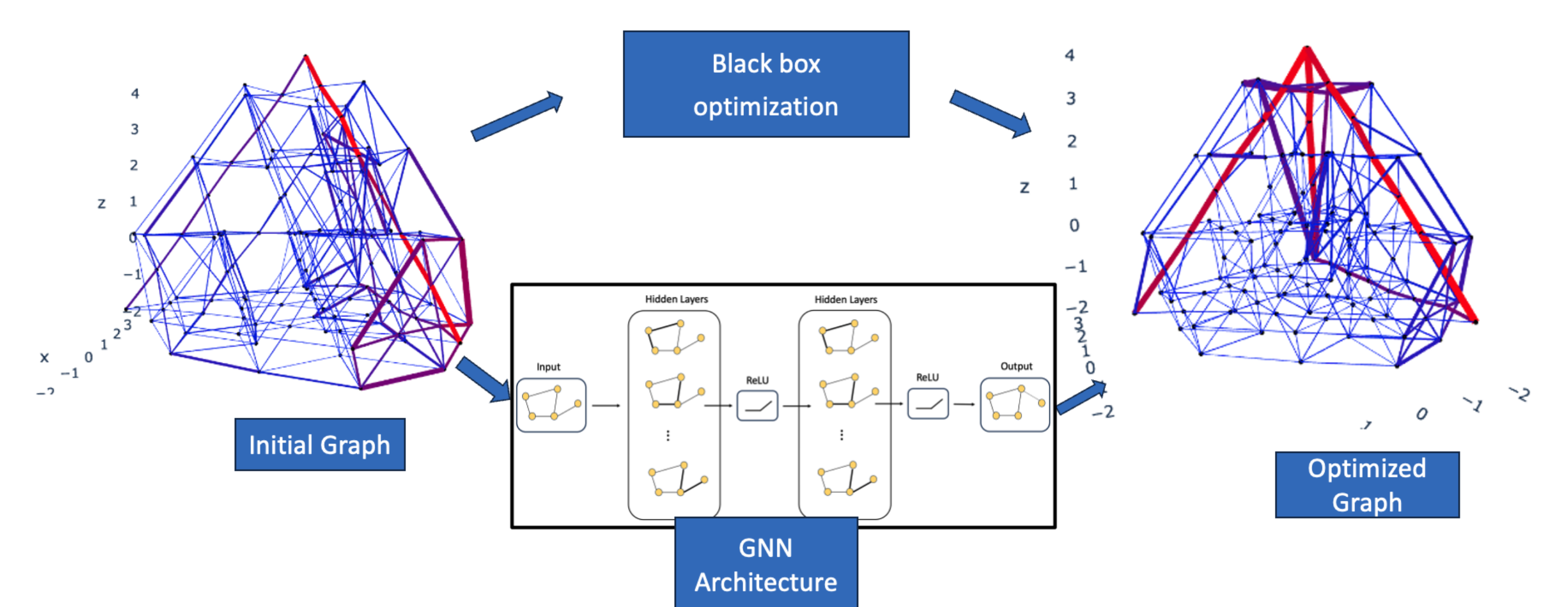


Figure 2: Schematic explaining the methodology for the GNN based approximation

## Discussion

For this work, we aim to employ graph neural networks based on convolution and attention[3]. In Graph Neural Networks (GNNs) that incorporate edge attributes, the learning process can be summarized into three pivotal steps: feature transformation through convolution, dynamic importance weighting via attention, and the aggregation of features for node updates. The mathematical representations of these steps are as follows:

**1. Convolution** The Node Features can be computed as  $\mathbf{h}'_i = \mathbf{W}\mathbf{h}_i$  and Edge Attributes:  $\mathbf{e}'_{ij} = \mathbf{V}\mathbf{e}_{ij}$  where  $\mathbf{W}$  and  $\mathbf{V}$  are learnable weight matrices for node features  $\mathbf{h}_i$  and edge attributes  $\mathbf{e}_{ij}$ , respectively.

### 2. Attention: Computing Attention Coefficients

■ Attention Coefficients:

$$\alpha_{ij} = \text{softmax}_j (\text{LeakyReLU} (a ([\mathbf{h}'_i || \mathbf{h}'_j || \mathbf{e}'_{ij}])))$$

This formulation dynamically assigns importance to neighbors' features and incorporates edge attributes into the computation of attention coefficients  $\alpha_{ij}$ .

### 3. Aggregation: Weighted Feature Aggregation

■ Updated Features:  $\mathbf{h}_i^{\text{new}} = \sigma \left( \sum_{j \in \mathcal{N}(i)} \alpha_{ij} (\mathbf{h}'_j + \mathbf{e}'_{ij}) \right)$

Aggregates transformed neighbor features weighted by attention coefficients, updating the node feature vector  $\mathbf{h}_i^{\text{new}}$ .

## References

- [1] Adcock, B., Brugiapaglia, S., Dexter, N. and Moraga, S., 2022. Near-optimal learning of Banach-valued, high-dimensional functions via deep neural networks. arXiv preprint arXiv:2211.12633.
- [2] Sitharam, M., Youngquist, J., Nolan, M. and Peters, J., 2019. Corner-sharing tetrahedra for modeling micro-structure. Computer-Aided Design, 114, pp.164-178.
- [3] Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P. and Bengio, Y., 2017. Graph attention networks. stat, 1050(20), pp.10-48550.