Application of Graph Neural Networks on Predicting Material Properties from Node Group Connectivity Matrices

Brennan Birn, Jared Guevara, Katrina Reyes, Yichen Tang, Riya Tiloda

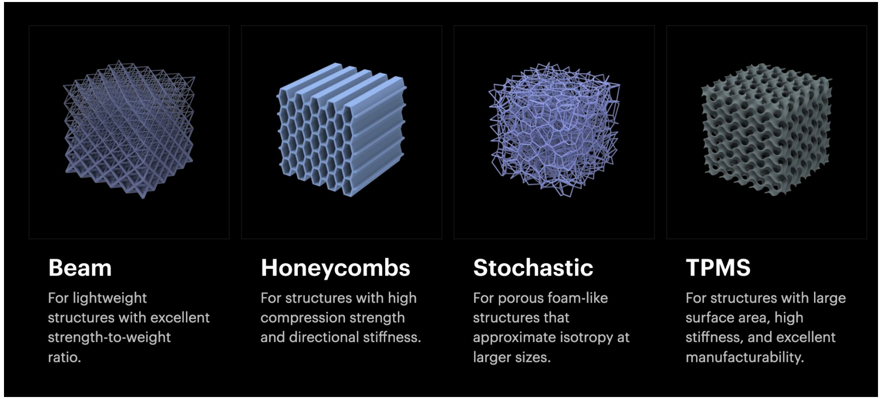
Chem 277B - Team 6

November 1, 2025

UC Berkeley

Project Summary

- Additively manufactured lattice structures can be mechanically superior to solid materials: lightweight, high strength, high tunability.
- Through cell topology and geometry optimization, lattices can exhibit enhanced mechanical properties relative to their volume, unachievable by other materials.

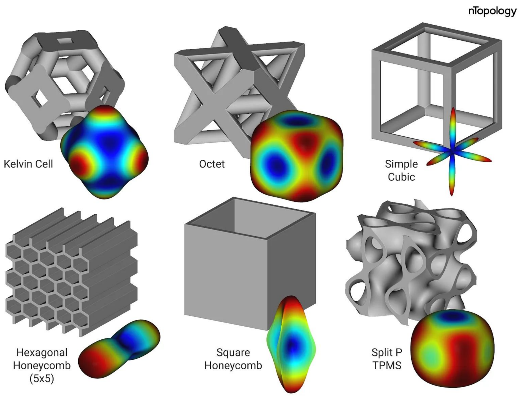


Lattice structures can be tuned to have properties mechanically superior to solid materials relative to their volume.

UC Berkeley

Challenge Addressed

- One tunable part of lattice structures is not directly controlled for at the structural level: Property surfaces in three dimensions
  - They approximate lattice response to external loading.
- Most literature focuses on tuning existing structures rather than discovering new ones.

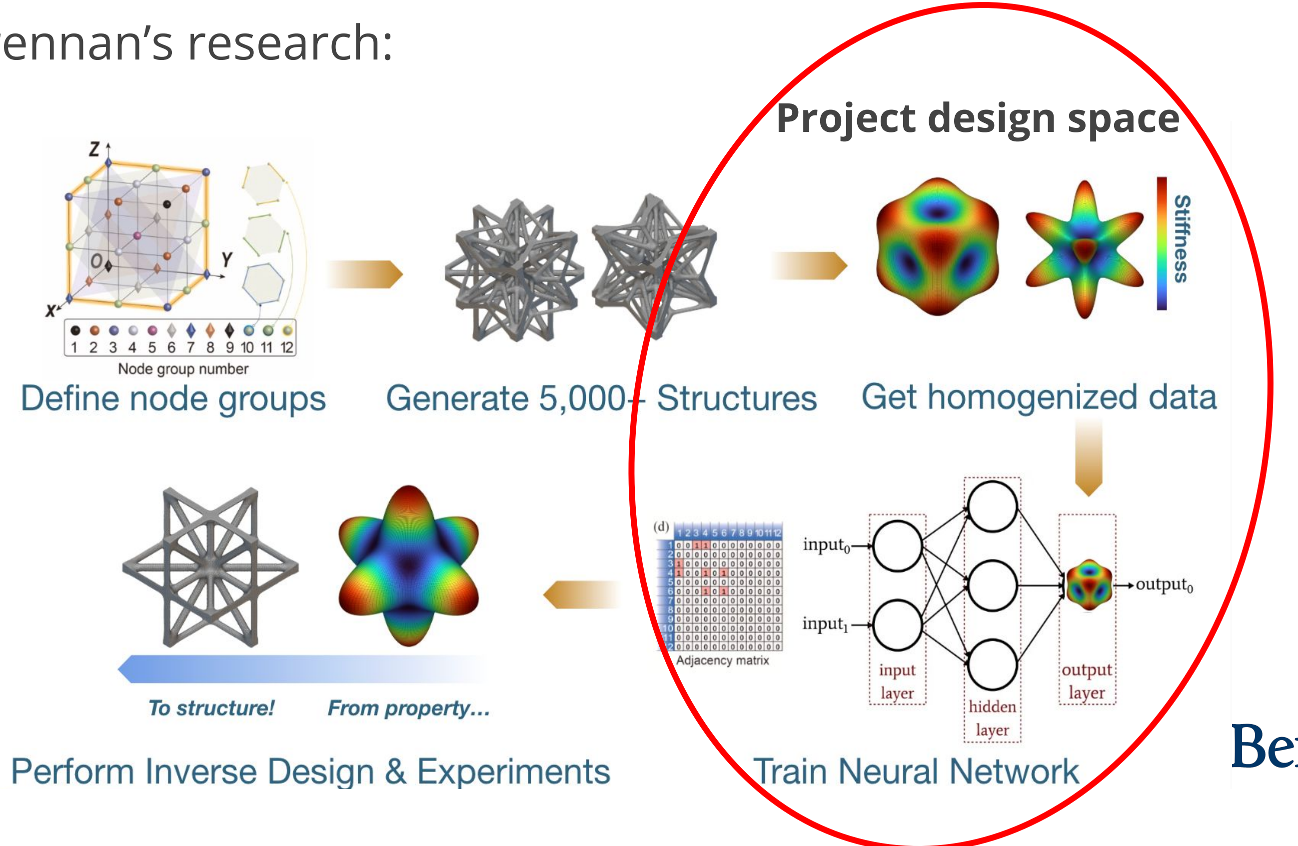


Current literature struggles to freely generate intricate structures while extracting key directional properties.

UC Berkeley

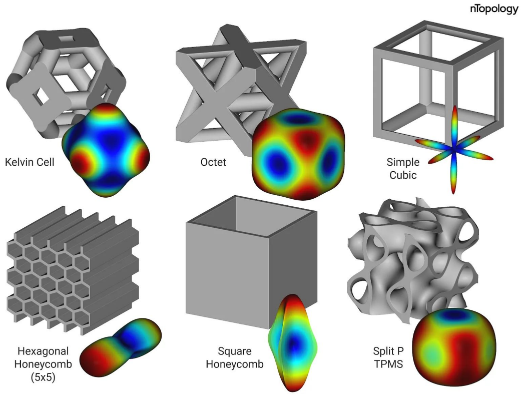
Background

Based on Brennan's research:



Berkeley

Quick lattice crash course



Orthotropic materials

Orthotropic materials have *3 mutually perpendicular symmetry planes*.

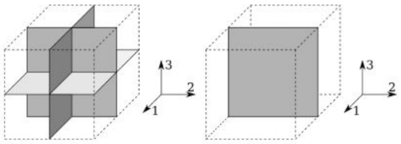


Figure-1: Orthotropic and transversely isotropic symmetry

Due to this symmetry, there are no coupling between normal stresses and shear strains, between shear stresses and normal strains, or between a shear stresses and a shear strains on different planes. Hence, the relation takes the form:

$$\begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \gamma_{23} \\ \gamma_{13} \\ \gamma_{12} \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} & S_{13} & 0 & 0 & 0 \\ S_{12} & S_{22} & S_{23} & 0 & 0 & 0 \\ S_{13} & S_{23} & S_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & S_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & S_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & S_{66} \end{bmatrix} \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \tau_{23} \\ \tau_{13} \\ \tau_{12} \end{bmatrix} \tag{9}$$

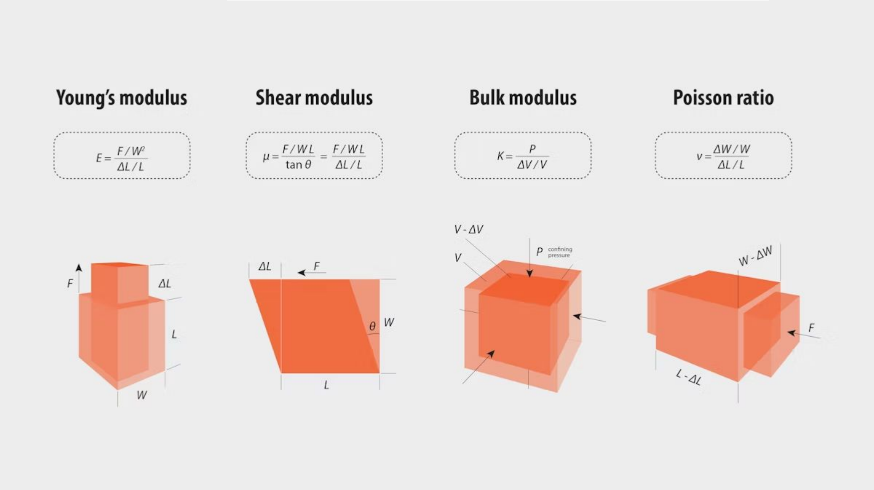
Orthotropic materials have *9 independent elastic constants* where the components of the compliance matrix expressed by the *engineering constants* are:

$$\begin{aligned} S_{11} &= \frac{1}{E_1} & S_{22} &= \frac{1}{E_2} & S_{33} &= \frac{1}{E_3} \\ S_{12} &= -\frac{\nu_{12}}{E_1} & S_{13} &= -\frac{\nu_{13}}{E_1} & S_{23} &= -\frac{\nu_{23}}{E_2} \\ S_{44} &= \frac{1}{G_{23}} & S_{55} &= \frac{1}{G_{13}} & S_{66} &= \frac{1}{G_{12}} \end{aligned} \tag{10}$$

where E_i are elastic moduli, ν_{ij} are Poisson's ratios, and G_{ij} are shear moduli.

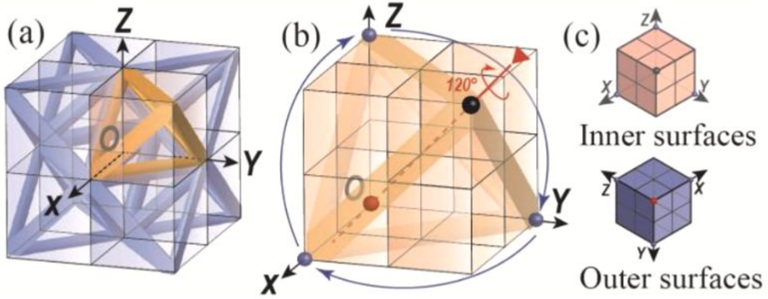
From the symmetry of the compliance matric:

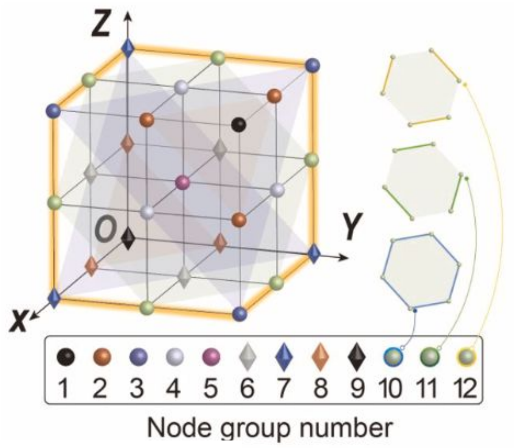
$$\frac{\nu_{ij}}{E_i} = \frac{\nu_{ji}}{E_j} \tag{11}$$



UC Berkeley

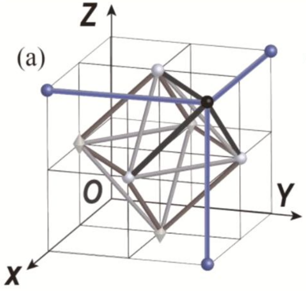
Solution (Prior inspiration)





Design space restrictions:

1. Origin located at center of cell.
2. Mirror planes located on axis planes
3. 3-fold rotation symmetries in 1 1 1 direction
4. Cell topology defined by 1/8 segment of cell



(d)

	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0	1	1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0
3	1	0	0	0	0	0	0	0	0	0	0	0
4	1	0	0	1	0	1	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	1	0	1	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0

Adjacency matrix

Cell topology is abstracted into a graph, with truss nodes considered as graph nodes and struts as graph edges.

UC Berkeley

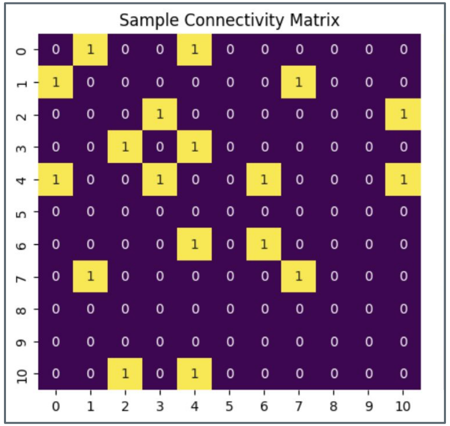
Dataset (~2500 structures)

Inputs:

- **Connectivity (adjacency) matrix**: 11x11 matrix describing connectivities between node groups
- **Rho (ρ*)**: relative density (either 0.15 or 0.30)

Outputs:

- **Compliance matrix (S)**: 6x6 matrix, inverse of stiffness matrix ($S = C^{-1}$)
- Varying material properties (**Young's moduli**, **shear moduli**, etc.)




```
Compliance matrix:
tensor([[16.2442, -4.3906, -4.3921,  0.0000,  0.0000,  0.0000],
        [-4.3906, 16.2539, -4.3922,  0.0000,  0.0000,  0.0000],
        [-4.3921, -4.3922, 16.2602,  0.0000,  0.0000,  0.0000],
        [ 0.0000,  0.0000,  0.0000, 38.9590,  0.0000,  0.0000],
        [ 0.0000,  0.0000,  0.0000,  0.0000, 39.0762,  0.0000],
        [ 0.0000,  0.0000,  0.0000,  0.0000,  0.0000, 39.0137]])
```

UC Berkeley

GNN configuration

Graph input:

- Edge indexes of Connectivity matrix (2 x [n_edges] tensor)

Node feature inputs:

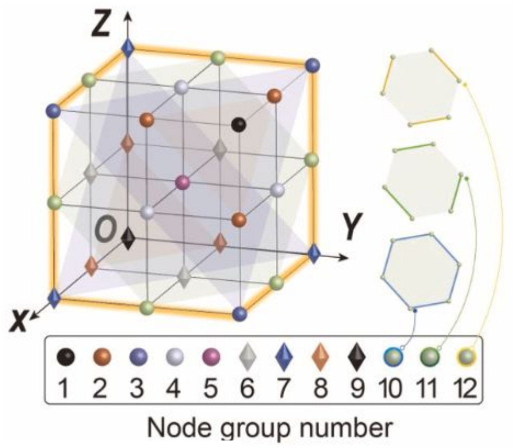
- Number of subnodes and connections
- Spatial location of up to three subnodes (normalized & flattened)

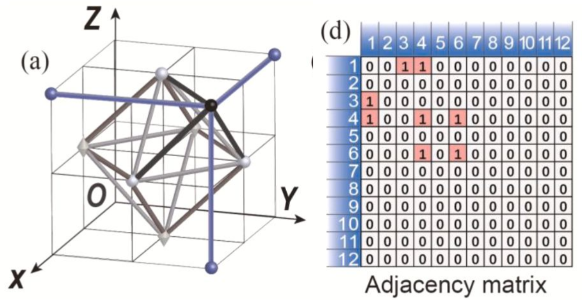
Edge feature input:

- Min distance between nodes

Post-GNN MLP input:

- Relative density ρ*
- Pooled node embeddings from the GNN





UC Berkeley

Methods and Implementation

Graph Convolutional Network (GCN)

- Updates made by averaging the features of neighboring nodes

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right)$$

Graph Attention Network (GAT)

- Using mask attention
- Computes attention between neighbors
- Relationship with neighbors and node state calculated based on weights and similarities

$$\mathrm{h}_i^{(l+1)} = \sigma\left(\sum_{j\in N(i)} \alpha_{ij} W^{(l)} h_j^{(l)}\right)$$

UC Berkeley

https://pharmanalytics.medium.com/understanding-network-diffusion-gcn-and-gat-how-do-they-relate-25c6eea7795a

Parameter sweep

Graph Convolutional Network (GCN)

```
SWEEP_GCN_EPOCHS        = (500,)          # reduced from 500
SWEEP_GCN_LR            = (5e-4,)         # fixed lr
SWEEP_GCN_HIDDEN_DIM    = (128, 256)      # width of GCN layers
SWEEP_GCN_NUM_CONVS     = (1, 2)          # depth of GCN (graph conv layers)
SWEEP_GCN_MLP_LAYERS    = (2, 3)          # depth of MLP head
SWEEP_GCN_HIDDEN_NEUR   = (128,)          # width of MLP head
SWEEP_GCN_DROPOUT       = (0.0, 0.1)      # dropout rate
```

Graph Attention Network (GAT)

```
SWEEP_ATTN_EPOCHS       = (500,)          # reduced from 500
SWEEP_ATTN_LR           = (5e-4,)         # fixed lr
SWEEP_ATTN_HIDDEN_DIM   = (64, 128)       # attention hidden dim (both divisible by 4 heads)
SWEEP_ATTN_NUM_LAYERS   = (3, 4)          # depth of attention stack
SWEEP_ATTN_NUM_HEADS    = (4,)            # number of attention heads
SWEEP_ATTN_MLP_LAYERS   = (2, 3)          # depth of MLP head
SWEEP_ATTN_HIDDEN_NEUR  = (128,)          # width of MLP head
SWEEP_ATTN_DROPOUT      = (0.0, 0.1)      # dropout rate
```

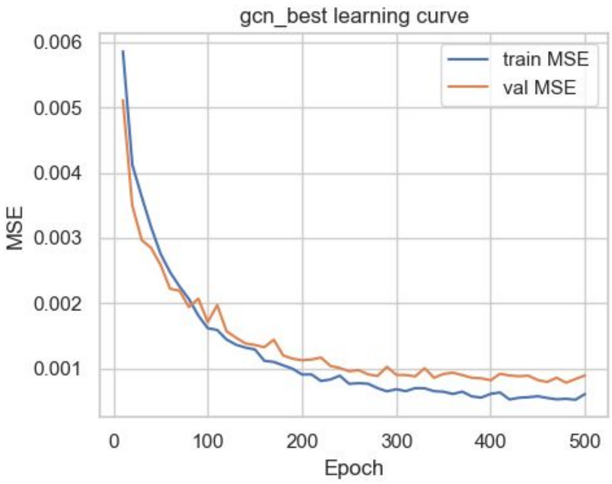
Best performing models results

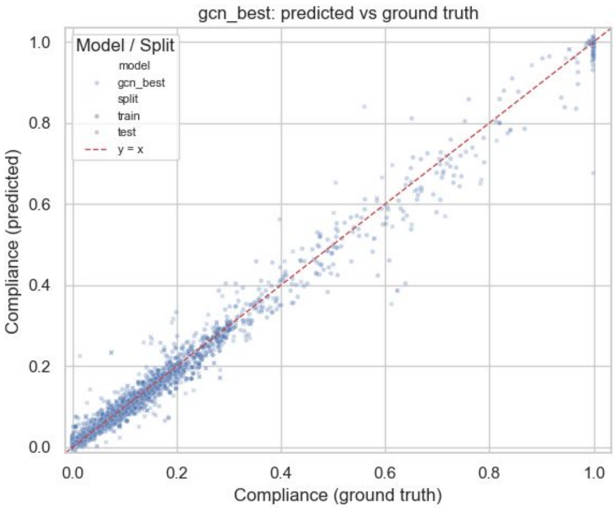
	model	epochs	lr	hidden_dim	num_convs	mlp_layers	hidden_neurons	dropout	test_mse	num_layers	num_heads
0	attn	500	0.0005	128	NaN	3	128	0.0	0.000747	4.0	4.0
1	attn	500	0.0005	128	NaN	2	128	0.1	0.000777	4.0	4.0
2	gcn	500	0.0005	256	2.0	2	128	0.1	0.000783	NaN	NaN
3	attn	500	0.0005	128	NaN	2	128	0.1	0.000785	3.0	4.0
4	gcn	500	0.0005	256	2.0	2	128	0.0	0.000796	NaN	NaN
5	attn	500	0.0005	128	NaN	2	128	0.0	0.000832	4.0	4.0
6	gcn	500	0.0005	128	2.0	2	128	0.0	0.000858	NaN	NaN
7	attn	500	0.0005	128	NaN	3	128	0.1	0.000861	3.0	4.0
8	attn	500	0.0005	128	NaN	3	128	0.0	0.000870	3.0	4.0
9	attn	500	0.0005	128	NaN	2	128	0.0	0.000883	3.0	4.0

Parameter sweep results

GCN

```
Best GCN config:
model                gcn
epochs               500
lr                0.0005
hidden_dim           256
num_convs            2.0
mlp_layers             2
hidden_neurons       128
dropout              0.1
test_mse        0.000783
```



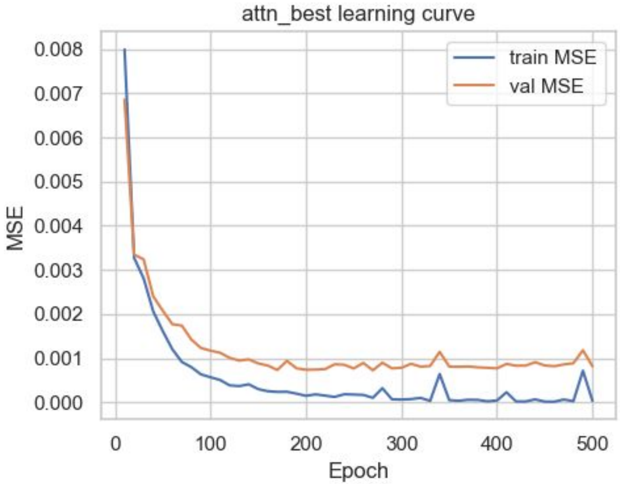


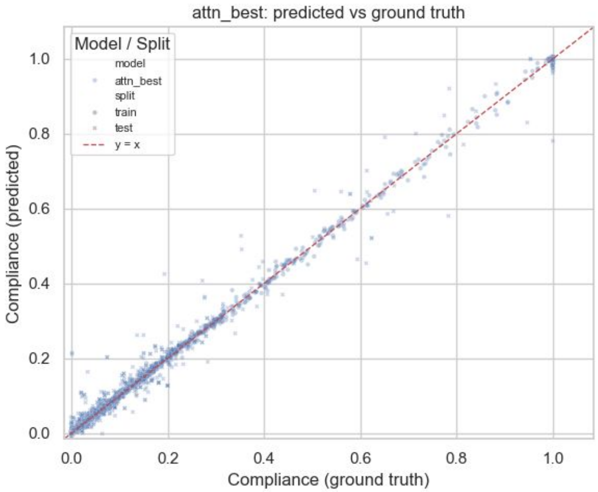
UC Berkeley

Parameter sweep results

GAT

Best Attention config:	
model	attn
epochs	500
lr	0.0005
hidden_dim	128
num_convs	NaN
mlp_layers	3
hidden_neurons	128
dropout	0.0
test_mse	0.000747
num_layers	4.0
num_heads	4.0





UC Berkeley

Parameter sweep results

UC Berkeley

2500 epoch evaluation

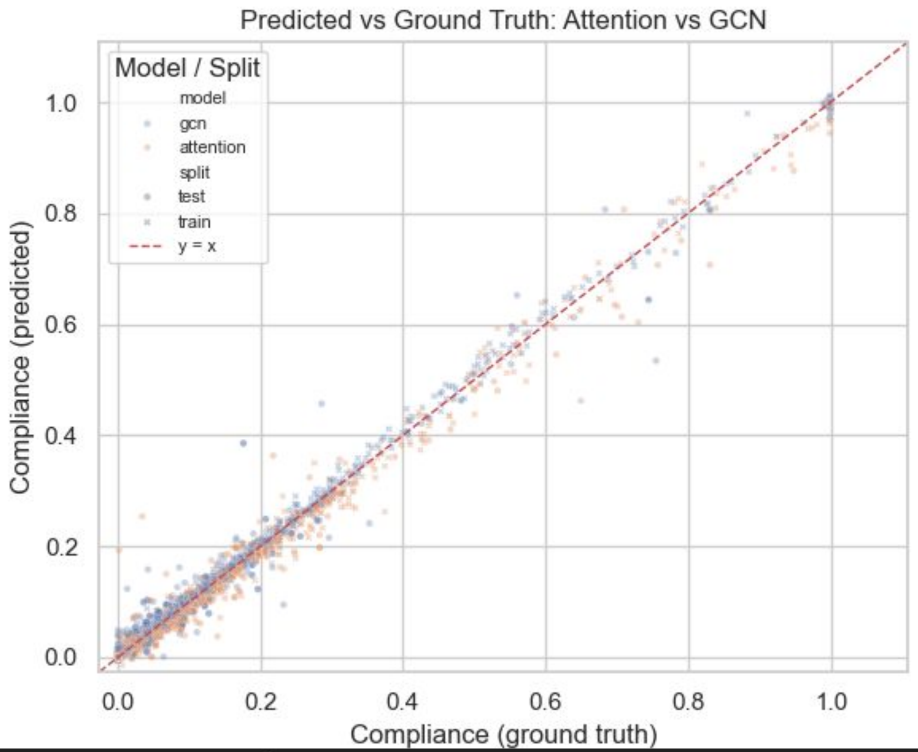
Graph Convolutional Network (GCN)

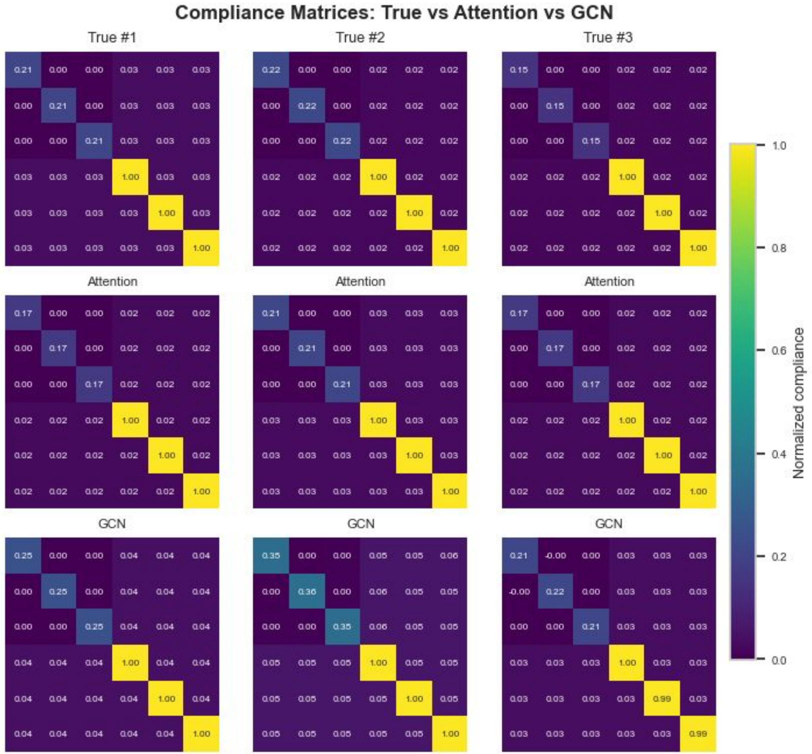
```python
# ---- Train GCN Model ----
model_gcn = GNN_v1(
    in_channels=num_node_features,
    embedding_size=128,
    out_dim=output_dim,
    dropout=0.0,
    mlp_layers=2,
    hidden_neurons=128,
    num_convs=1
).to(device)
```

Graph Attention Network (GAT)

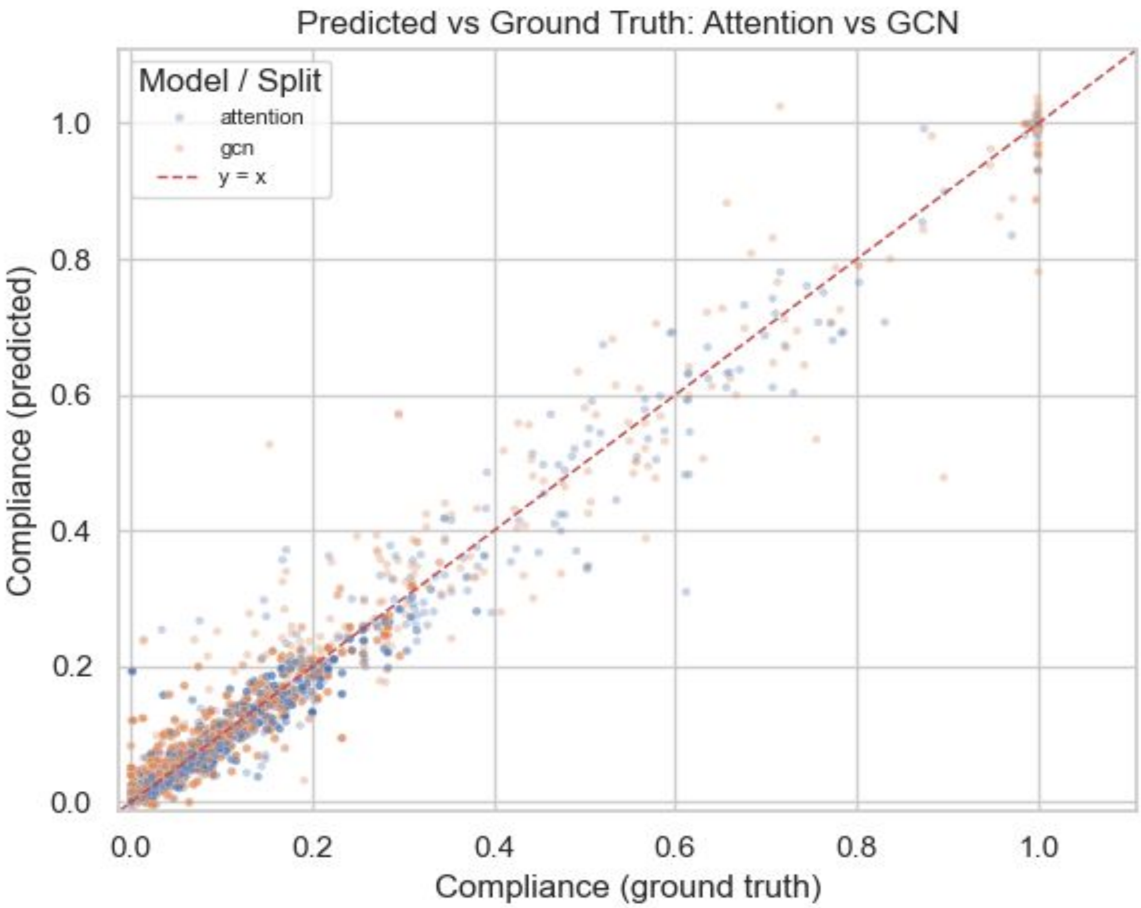
```python
# ---- Train Attention Model ----
model_attn = GNN_v2_Attention(
    in_channels=num_node_features,
    hidden_dim=64,
    out_dim=output_dim,
    num_heads=4,
    num_layers=3,
    mlp_layers=2,
    hidden_neurons=128,
    dropout=0.1,
).to(device)
```

2500 epoch evaluation





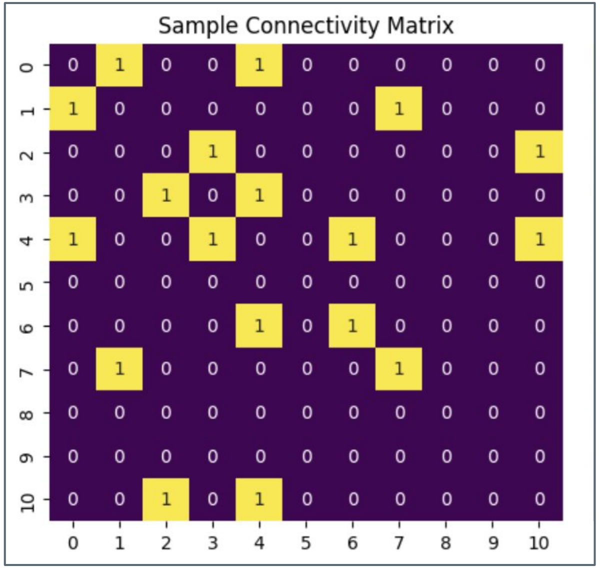

```
Attention metrics: {'MAE': 0.00823308527469635, 'MSE': 0.00024697312619537115, 'RMSE': 0.015715378652624668, 'R2': 0.996857659791567}
GCN metrics: {'MAE': 0.007370175328105688, 'MSE': 0.00027414553915150464, 'RMSE': 0.016557340944472473, 'R2': 0.9965119333812664}
```




```
Attention metrics: {'MAE': 0.014083466492593288, 'MSE': 0.0007193599012680352, 'RMSE': 0.026820885542204517, 'R2': 0.9908068264344685}
GCN metrics: {'MAE': 0.01820702664554119, 'MSE': 0.0011527605820447206, 'RMSE': 0.03395232807989344, 'R2': 0.9852681147873581}
```

Demo

Using the previously shown connectivity matrix as our graph:



rho	mean_E	mean_G
0.3	0.061528	0.025630

```
Compliance Matrix max: 39.08, min: -4.39
```

Discussion

Current limitations:

- Only ~2500 datapoints.
- Constant learning rate for each model sweep.
- Couldn’t run K-Fold cross-validation due to time constraints.

Future work:

- Inverse design, coupling regression model with Genetic Algorithms.
- Demo with 3D structures.
- Create exportable material models to be used in simulations.
- Other evaluation metrics so that weighting is assessed fairly.

UC Berkeley

Q&A

References:

- Birn, Brennan, et al. “Freeform Lattice Structure Optimization via Generative Design: Enhanced Isotropy and Stress Delocalization.” *Advanced Engineering Materials*, 7 Nov. 2025, advanced.onlinelibrary.wiley.com/doi/abs/10.1002/adem.202501559, https://doi.org/10.1002/adem.202501559. Accessed 1 Dec. 2025.
- Gongora, Aldair E, et al. “Accelerating the Design of Lattice Structures Using Machine Learning.” *Scientific Reports*, vol. 14, no. 1, 14 June 2024, pp. 13703–13703, www.nature.com/articles/s41598-024-63204-7, https://doi.org/10.1038/s41598-024-63204-7. Accessed 1 Dec. 2025.
- Jiang, Bingyue, et al. “GNNs for Mechanical Properties Prediction of Strut-Based Lattice Structures.” *International Journal of Mechanical Sciences*, vol. 269, May 2024, p. 109082, www.sciencedirect.com/science/article/pii/S0020740324001255, https://doi.org/10.1016/j.ijmecsci.2024.109082. Accessed 1 Dec. 2025.
- Kim, Seungjin, et al. “Design of Cubic-Symmetric Lattice Structures for Target Mechanical Properties Using Machine-Learned Models of Property Prediction and Structure Reconstruction.” *Ssrn.com*, 2025, papers.ssrn.com/sol3/papers.cfm?abstract_id=5199373. Accessed 1 Dec. 2025.
- Maurizi, Marco, et al. “Inverse Design of Truss Lattice Materials with Superior Buckling Resistance.” *Npj Computational Materials*, vol. 8, no. 1, 29 Nov. 2022, www.nature.com/articles/s41524-022-00938-w, https://doi.org/10.1038/s41524-022-00938-w. Accessed 1 Dec. 2025.
- “NTop | Computational Design Software | Formerly NTopology.” *NTop*, 2025, www.ntop.com/. Accessed 1 Dec. 2025.
- Rubi, Leire Roma, et al. *Tensegrity Structures for Energy Absorption in Aerospace Landing and Reusable Rocket Systems*. 1 Jan. 2025, arc.aiaa.org/doi/epdf/10.2514/6.2025-97105, https://doi.org/10.2514/6.2025-97105. Accessed 1 Dec. 2025.

UC Berkeley