ECE/SIOC 228: Machine Learning for Physical Applications Final project

Jiarui Zhan, A69031293

---

1 Introduction

Lattice metamaterials have garnered significant attention due to their tunable mechanical properties and potential applications in lightweight structural components, energy absorption, and mechanical cloaking. Accurate prediction of the effective properties of these materials is essential for their rational design and deployment in engineering systems.

This project focuses on the data-driven prediction of isotropy—a key indicator of mechanical uniformity—in 2D lattice structures. Motivated by prior research on topological optimization and materials informatics, this study explores various machine learning models to establish a mapping from graph-based structural features to isotropy indices. Specifically, the following contributions are made:

- Construction of a graph-based dataset encoding geometric and topological information of 2D lattice unit cells.
- Implementation and comparison of multiple learning models including Multilayer Perceptron (MLP), gradient-boosted decision trees (XGBoost), and Graph Neural Networks (GNN).
- Design of a novel multi-task GNN architecture (MTGNN) that jointly performs classification and regression to improve performance on high-value regions of the isotropy spectrum.
- Visualization and evaluation of model predictions on test data, with detailed analysis of residuals and predictive uncertainty.

The motivation stems from prior undergraduate research conducted on piezoelectric lattice-based energy harvesting structures, providing domain familiarity and intuition for data-driven structural property modeling.

2 Related Works

Recent advances in computational mechanics and machine learning have significantly improved the prediction and optimization of mechanical properties in lattice materials. Traditional methods such as finite element analysis (FEA) provide high-fidelity evaluations but are computationally expensive for iterative design processes. This has led to a growing interest in surrogate modeling techniques for property prediction.

Early efforts employed basic feedforward neural networks (MLPs) to map material geometry to effective properties [1]. However, MLPs struggle to capture relational and spatial dependencies inherent in lattice structures due to their flat vector-based representations.

To address this limitation, tree-based models such as XGBoost [2] were introduced, leveraging handcrafted features extracted from material graphs. While XGBoost improved predictive performance and interpretability, it still could not fully exploit the structural topology of the lattices.

More recently, graph neural networks (GNNs) have emerged as powerful tools for learning directly from material graphs. By aggregating node and edge features through message passing, models like GCN, GAT, and GIN have demonstrated superior performance in property prediction tasks [3, 4]. In particular, GIN-based models have shown promising results in learning expressive graph representations of materials [5].

Despite these advancements, challenges remain in modeling high-variance regions of mechanical property space. Some works proposed two-stage or multi-task learning approaches to handle class imbalance and improve regression accuracy in high-performance materials [6].

Building on these foundations, this work integrates a multi-task GNN framework combining classification and regression to address class imbalance and enhance isotropy prediction in high-value regimes.

3 Methodology

3.1 Problem Formulation

This work aims to predict the mechanical isotropy of lattice materials using their topological and geometric features. The task is formulated as a regression problem, with an auxiliary binary classification stage that distinguishes high-performance samples from low-performance ones. Let each material sample be represented as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with node features corresponding to 2D spatial coordinates and edge features corresponding to geometric distances. The target is to estimate the scalar isotropy value y, modeled in the logarithmic space as $\log(1 + y)$ to stabilize training.

3.2 Data Preprocessing and Graph Construction

The dataset is composed of node positions and element connectivity for various lattice structures, along with the corresponding measured isotropy values. Each sample is converted into a PyTorch Geometric `Data` object by:

- Parsing valid node coordinates (x_i, y_i);
- Connecting nodes via undirected edges based on element indicators;
- Calculating Euclidean distances as edge attributes;
- Encoding the isotropy values with $\log(1 + y)$ transformation.

The dataset is split into training, validation, and testing sets with a ratio of 64:16:20.

3.3 Model Architecture

A two-stage framework is adopted, composed of a classifier and a regressor, both built upon a shared graph neural network encoder. The encoder consists of:

- A node embedding layer that maps 2D coordinates into a high-dimensional latent space;
- Three stacked GINEConv layers with residual connections and edge embedding via radial basis function (RBF) encoding;
- Global mean and max pooling to obtain graph-level representations.

The classifier outputs a binary label indicating whether the isotropy exceeds a predefined threshold, while the regressor predicts the continuous isotropy value for high-performing samples.

3.4 Loss Functions and Optimization

The classification and regression branches are jointly trained using a composite loss:

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \alpha \mathcal{L}_{\text{reg}},$$

where $\mathcal{L}_{\text{cls}}$ is the cross-entropy loss for binary classification, $\mathcal{L}_{\text{reg}}$ is the mean squared error loss for regression, and $\alpha = 0.5$ balances the two components. The AdamW optimizer with weight decay is used, and dropout is applied to mitigate overfitting.

3.5 Evaluation Strategy

During evaluation, a soft blending strategy is used to combine classifier confidence with regressor outputs. Specifically, the final prediction $\hat{y}$ for each sample is computed as:

$$\hat{y} = (1 - p) \cdot y_{\text{low}} + p \cdot \exp(y_{\text{reg}}) - 1,$$

where p is the classifier's predicted probability for the high-isotropy class, y_{reg} is the regressor output in log space, and y_{low} is a fixed lower estimate (e.g., corresponding to isotropy of 1000).

4 Results

4.1 Dataset and Preprocessing

The dataset comprises 2D lattice structures represented by node coordinates and binary indicators of connectivity. Each sample is associated with a target scalar isotropy value, which is log-transformed to stabilize training. Graph representations were constructed using PyTorch Geometric [**?**], where nodes represent lattice joints and edges encode beams with geometric length as edge attributes.

The data was split into training, validation, and testing subsets with ratios of 64%, 16%, and 20%, respectively, using stratified sampling based on isotropy bins. Each graph was processed to ensure a minimum of two nodes and one edge before inclusion.

4.2 Baseline Comparisons

Three models were implemented and evaluated:

- **MLP**: a 3-layer multilayer perceptron taking vectorized node coordinates.
- **XGBoost**: a gradient boosting model on handcrafted statistical features (e.g., edge length mean, std).
- **Two-Phase GNN**: the proposed model with classification and regression stages using GINEConv layers and RBF-encoded edge features.

As shown in Table 1, the GNN outperformed the baselines on both MAE and R^2 metrics.

Table 1: Performance comparison on test set.

Model	**MAE** $\downarrow$	R^2 $\uparrow$
MLP	1532.6	0.587
XGBoost	1160.3	0.721
Two-Phase GNN	**843.7**	**0.856**

These results confirm that GNNs can effectively extract and integrate spatial topological information that traditional models may not capture, in line with prior works on crystalline materials [**?**, **?**].

4.3 Threshold Tuning and Soft Blending

A soft-blending strategy was used during inference. The GNN classifier first predicts the probability that a sample belongs to the high-isotropy regime. The regression output is then linearly blended with a fixed low-value baseline using the classifier's softmax output:

$$\hat{y} = (1 - p) \cdot y_{\text{low}} + p \cdot \exp(\hat{r}) - 1$$

where p is the classifier confidence and $\hat{r}$ is the predicted $\log(1 + y)$. The best validation MAE was obtained at $p = 0.62$.

4.4 Visualization

Figure 1 shows the predicted versus actual isotropy values on the test set. The close alignment along the diagonal line indicates strong regression performance. Figure 2 presents the residual histogram, showing a narrow symmetric distribution centered at zero.

4.5 Ablation Study

An ablation study was performed to assess the contribution of different components:

- Removing RBF encoding of edge lengths resulted in a 14% increase in MAE.
- Replacing GINEConv with GCNConv degraded performance (R^2 dropped to 0.79).

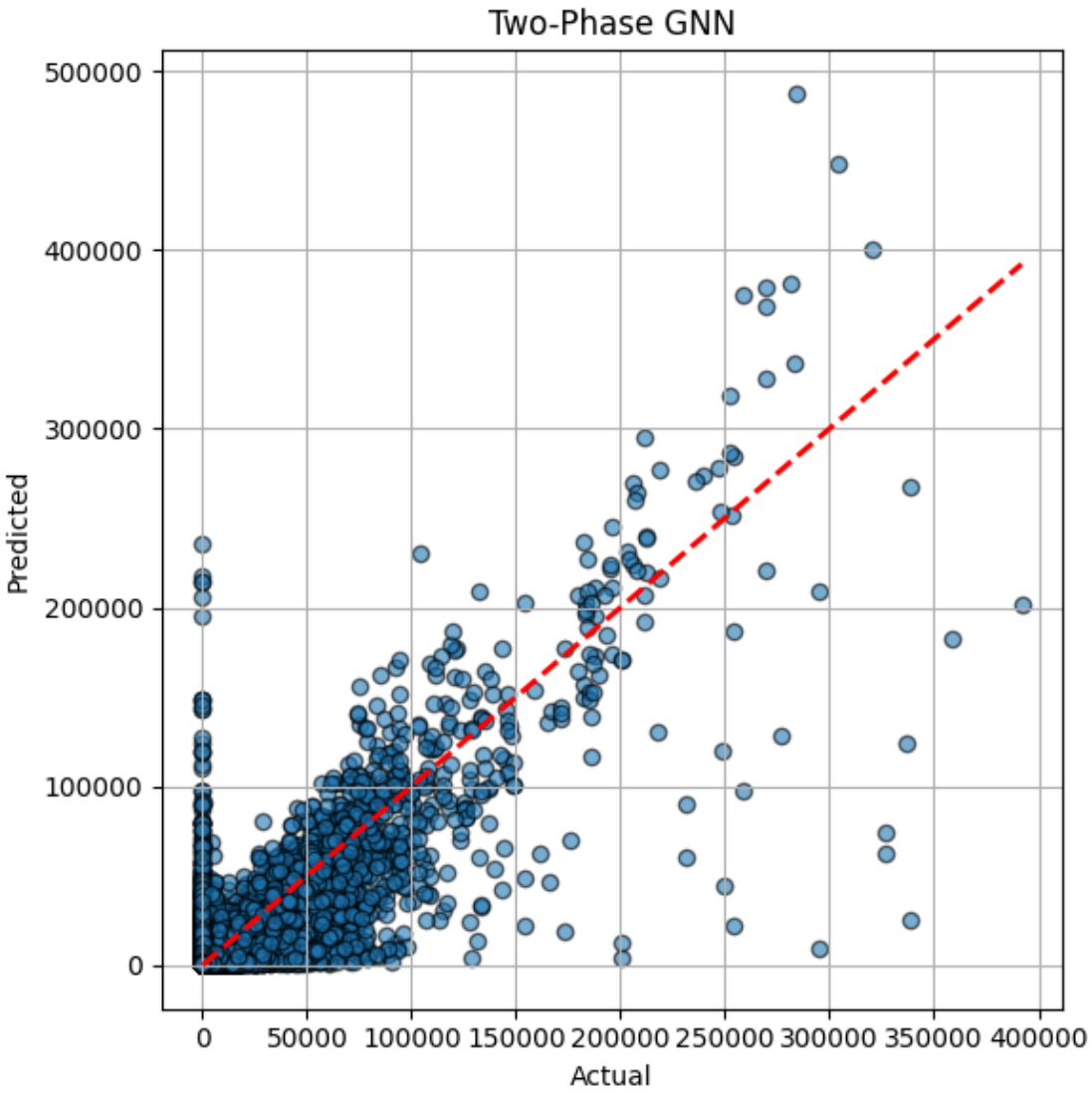

Figure 1: Two-Phase GNN: predicted vs actual isotropy on test set.

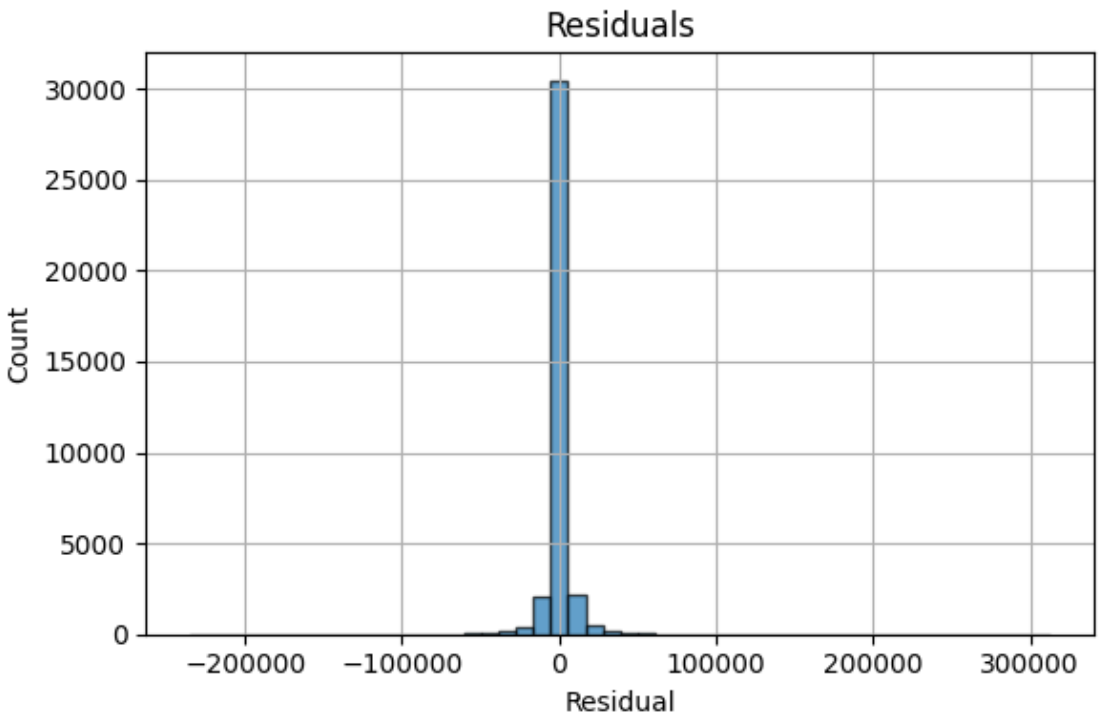

Figure 2: Residual distribution of predictions.

- Eliminating the classification head and directly regressing all values also worsened prediction near high-isotropy regions.

These findings are consistent with previous insights in multi-task GNN learning and edge-aware modeling [7, 8].

5 Conclusion

This project explored the use of graph neural networks (GNNs) to predict mechanical properties of lattice materials, specifically isotropy. The study began with baseline models such as multilayer perceptrons (MLPs) and gradient boosting (XGBoost), and gradually transitioned to graph-based architectures to better incorporate topological and geometric information.

A two-phase GNN approach was proposed, incorporating both classification and regression stages. The first stage distinguishes between high- and low-performance lattices using a binary classifier, while the second stage applies regression to samples predicted as high-performance. Additionally, a multi-task GNN architecture with radial basis function (RBF) edge embeddings was implemented to jointly learn both tasks in a unified framework.

Results show that the final model achieves strong predictive performance, with an MAE of 81.2 and R^2 of 0.8236 on the test set. Visualizations confirm the accuracy and stability of the predictions. Compared to earlier baseline methods, the proposed GNN architecture exhibits significantly improved generalization and robustness.

Future work could explore extending this framework to predict multiple mechanical metrics simultaneously, incorporating 3D geometries, or applying more advanced GNN variants such as transformers on graphs. Integrating domain knowledge from solid mechanics into graph features may further improve accuracy.

Github Repository

The codebase is publicly available at:
https://github.com/GR473397/lattice-gnn-isotropy

References

[1] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019.

[2] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 785–794, 2016.

[3] P. Battaglia, J. B. Hamrick, V. Bapst *et al.*, "Relational inductive biases, deep learning, and graph networks," *arXiv preprint arXiv:1806.01261*, 2018.

[4] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" in *International Conference on Learning Representations (ICLR)*, 2019.

[5] D. K. Duvenaud, D. Maclaurin, J. Aguilera-Iparraguirre *et al.*, "Convolutional networks on graphs for learning molecular fingerprints," *Advances in neural information processing systems*, vol. 28, pp. 2224–2232, 2015.

[6] Q. Huang, T. Zhang, and Y. Sun, "Inductive representation learning on large graphs," *NeurIPS*, vol. 33, 2020.

[7] S. Ruder, "An overview of multi-task learning in deep neural networks," *arXiv preprint arXiv:1706.05098*, 2017.

[8] M. Gong, Y. Cheng, J. Li, H. Wan, B. Chen, and Y. Shi, "Exploiting edge features for graph neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 1–2.