MACQUARIE University

SYDNEY·AUSTRALIA

COMP8221 Advanced Machine Learning

Semester 1, 2025

Assessment 2

Node Classification Task

on OGBN-Arxiv

with

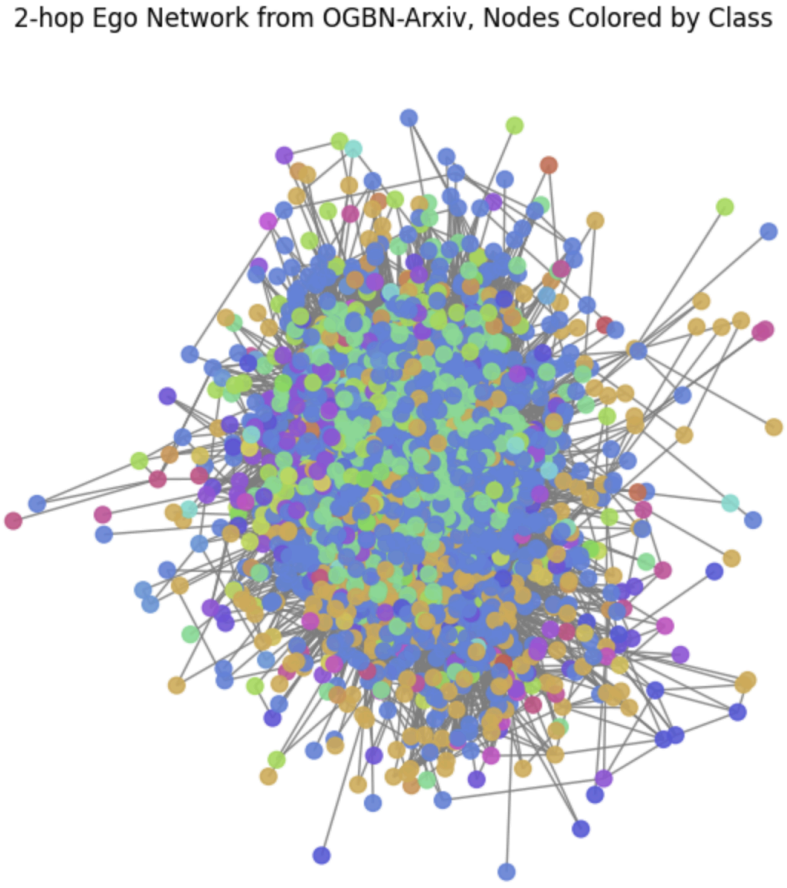
GAT Models

Motivation & Explanation of Data/Task

Motivation

Initially, the Cora dataset was chosen to quickly prototype and validate the GNN models due to its popularity in benchmarking usage and its size. However, there were several concerns with this dataset that any models could potentially achieve high accuracy with minimal training; that is, limited scalability and against the purpose of this assessment.

To address this, the OGBN-Arxiv dataset has been chosen. The dataset is a much larger and more complex citation network of papers on arXiv. In this dataset, each node represents a paper, and each edge represents a citation. The main task is node classification, where we predict the research area for each paper, using both the feature vectors from the paper's content and its citations. Our motivation for this choice was to test the effectiveness and scalability of advanced GNN models, such as GAT, label propagation, and feature augmentation, in a setting that is more representative of real-world challenges.



Pre-Processing

For initial experiments on Cora, no pre-processing was required. However, when we moved to the OGBN-Arxiv dataset, we faced both computational and memory constraints due to its larger scale. Fortunately, OGBN-Arxiv provides node features that are already normalised, so further feature scaling was unnecessary. To manage the increased computational demands, we adopted several strategies:

First, **mini-batch sampling** was chosen instead of full-batch training, which exceeded available GPU memory when we first tried it out. Specifically, we use NeighborLoader to sample a fixed number of neighbours per node at each layer. This not only made training feasible on commodity hardware but also mimicked the neighbourhood sampling strategies used in large-scale GNN analysis.

Second, we **reduced the batch size** for the OGBN-Arxiv experiments, striking a balance between memory efficiency and model convergence. Empirically, a batch size of 512 or 1024 enabled stable training without encountering memory errors.

Model Explanation and Appropreateness

GAT Baseline

For our baseline, we implemented a 3-layer Graph Attention Network (GAT), inspired by Veličković et al. (2018) but adapted for scalability to large graphs like OGBN-Arxiv. Unlike the original model, which used full-batch training on small datasets. We use NeighborLoader to process mini-batches of subgraphs. This design allows us to efficiently train on OGBN-Arxiv, which contains over 170,000 nodes and more than 1 million edges, while capturing neighbour information for each target node.

GAT was chosen for its ability to learn the relative importance of each node's neighbours through attention mechanisms. In citation networks such as OGBN-Arxiv, not all citations are equally informative, meaning that most papers cite works related to their field, but there are also cross-field citations. GAT's attention enables the model to dynamically weigh these relationships, capturing intricate structural patterns and enhancing node classification accuracy.

By aggregating neighbour features with learned attention, GAT can better classify nodes even in large, heterogeneous graphs. Below is the structure of the model, and the explanation of the selection follows:

Input: Node features (x) and graph structure (edge_index)

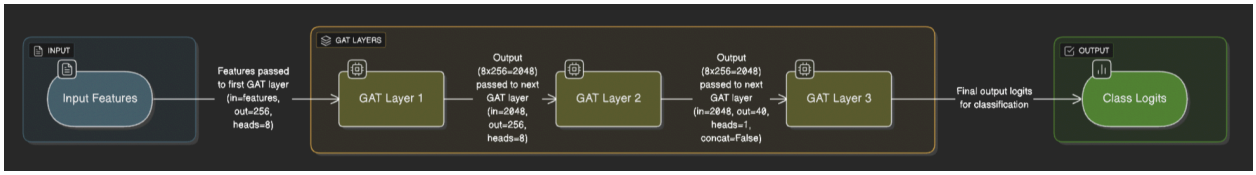
Layers: 3 stacked GATConv layers

- **First and hidden layers:** 8 attention heads each, hidden size 256 per head (outputs concatenated, for 2048 features per node in hidden layers)
- **Output layer:** 1 attention head, output dimension = number of classes (40)

Activation: ELU after each GATConv except the last

Dropout: Applied to input and after each layer (p=0.6)

Optimizer: Adam with learning rate 0.002, weight decay 5e-4



3 x Layers:

We stack three GATConv layers to allow each node to aggregate information from up to three hops away. This depth is a practical trade-off: it's deep enough to capture global context, but shallow enough to avoid over-smoothing, where node embeddings become too similar across the graph.

8-Head Attention:

Using multiple attention heads per layer increases the robustness and expressiveness of the model, enabling it to learn diverse patterns of interaction among neighbours.

256 Hidden dimension:

A hidden size of 256 per head (2048 features per node in the hidden layers) provides sufficient capacity to model complex relationships in a large citation network like OGBN-Arxiv.

NeighborLoader (mini-batch sampling):

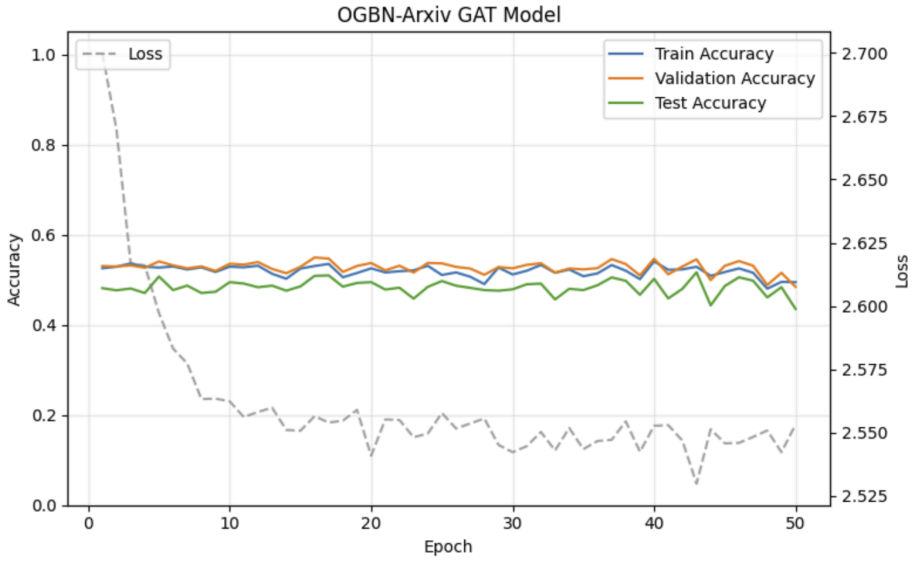
Full-batch training is not feasible for OGBN-Arxiv due to hardware constraints. The NeighborLoader enables scalable training by sampling a subgraph for each batch, maintaining

efficiency and memory usage while still providing each node with a meaningful local context. This introduces some randomness into the neighbourhood seen by each node at each epoch, but is a standard solution for large-scale GNNs.

Insights and Results

Baseline GAT Model Performance

The baseline GAT model demonstrates a slight improvement in accuracy over the first few epochs, with both training and validation accuracy stabilising around 0.53, and test accuracy slightly lower, fluctuating near 0.48. However, the performance quickly plateaus, and the gap between training/validation and test accuracy remains consistent throughout training. This suggests the model is only partially capturing the patterns necessary for effective generalisation.



There could be many reasons, including the dataset structure complexity, limited neighbourhood sampling, or insufficient model expressiveness. The relatively flat accuracy curves across epochs indicate that the current architecture or hyperparameter configuration may not be optimal for extracting deeper patterns from the graph. Overall, while the model does learn meaningful representations, its performance is notably below the state-of-the-art results for OGBN-Arxiv (which typically reach above 70% accuracy). This highlights the need for further tuning,

increased neighbourhood sampling, or more advanced architectures (such as hybrid models combining GAT with GraphSAGE or MLPs) to better capture the rich structure of the dataset.

GAT-MLP Model Performance

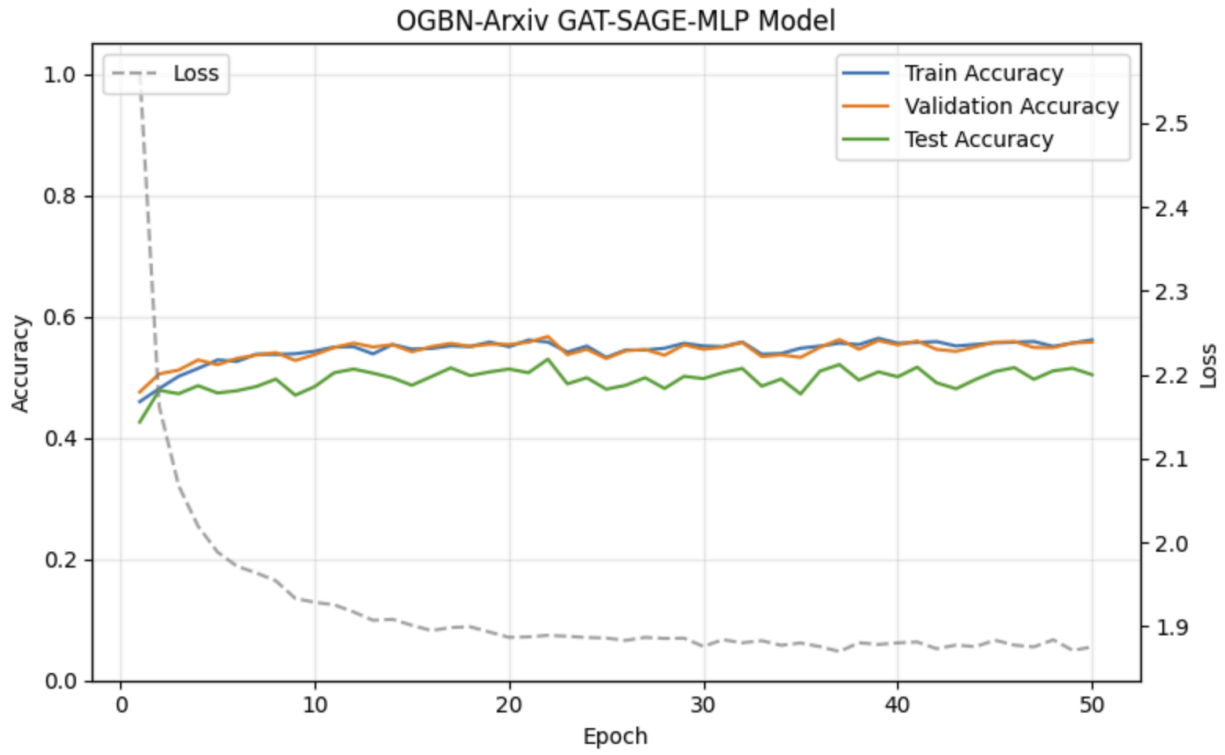
The GAT-MLP model integrates graph attention layers with a multi-layer perceptron (MLP) classifier, aiming to enhance expressive power compared to the baseline model. As shown in the results below, the model achieves a steady improvement in both training and validation accuracy during the first several epochs. Both metrics soon plateau around 0.54, while the test accuracy settles slightly lower, fluctuating near 0.50. Loss decreases rapidly early on, and then levels off, consistent with models that quickly learn basic representations.

Compared to the baseline GAT, the GAT-MLP variant converged slightly faster and achieved a marginally higher test accuracy, indicating that the additional MLP layers help capture some nonlinear relationships. However, the persistent gap between training/validation and test accuracy signals continuing challenges with generalisation. This may result from limitations in either the model's depth.

Overall, while the GAT-MLP model offers a modest performance boost and increased flexibility over the vanilla GAT, its results remain below the current state-of-the-art accuracy. This reinforces the value of further architectural enhancements, such as introducing skip connections, deeper hybrid blocks (e.g., SAGE + GAT + MLP).

GAT-SAGE-MLP Model Performance

The GAT-SAGE-MLP model further extends the hybrid approach by stacking graph attention layers, a GraphSAGE layer, and a multi-layer perceptron. This architecture aims to leverage the strengths of both GAT and SAGE, followed by an MLP for expressive final classification.

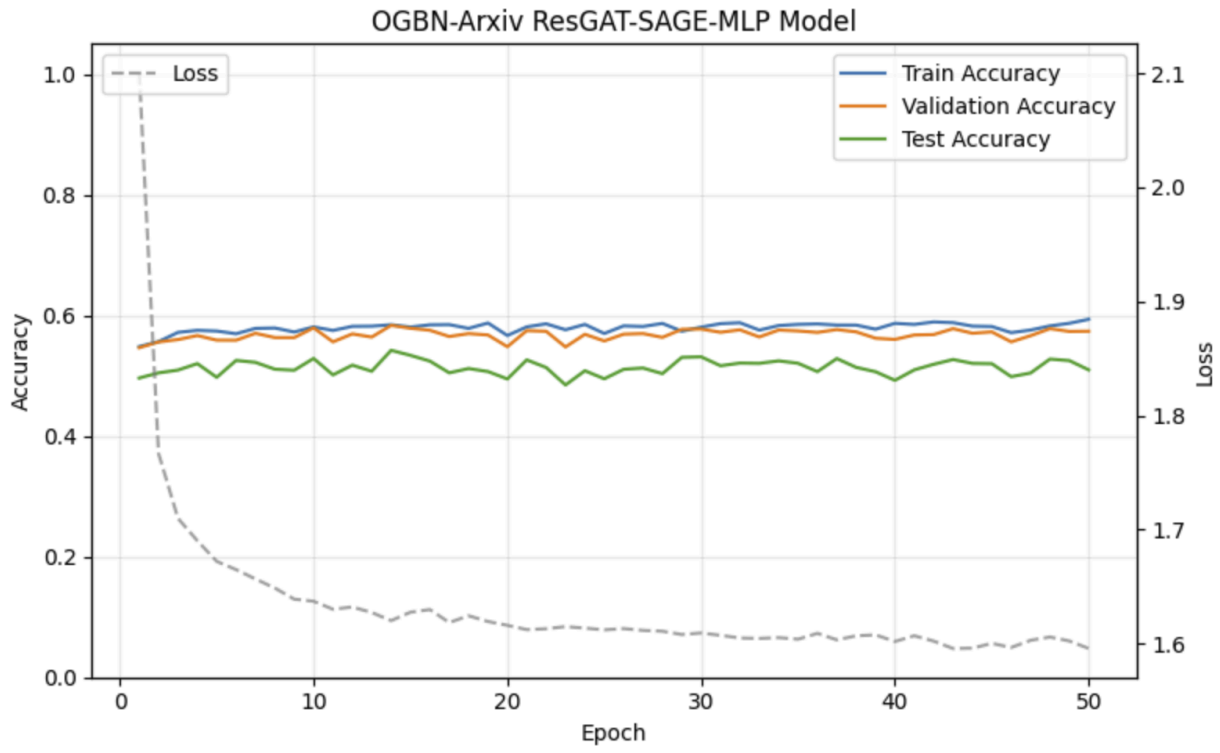


As we can see in the training plot, this model achieves steady improvements in both training and validation accuracy, converging to values slightly above 0.56 for validation and peaking above 0.51 on the test set. The loss curve shows consistent downward progress across epochs, and the accuracy metrics are a clear improvement over both the pure GAT and the GAT-MLP variants. Notably, the model generalises better, with the test accuracy achieving higher stability and less fluctuation compared to previous architectures.

The boost in performance can be because of the additional GraphSAGE layer, which enables the model to aggregate and smooth information over the graph more efficiently before passing richer representations into the MLP classifier. This multi-stage pipeline helps mitigate the expressiveness bottleneck observed in the earlier ablations. However, a moderate gap remains between train/validation and test accuracy, indicating there is further room for improvement.

ResGAT-SAGE-MLP Model Performance

The ResGAT-SAGE-MLP model integrates residual connections into the GAT-SAGE-MLP stack, aiming to address issues like vanishing gradients and feature oversmoothing. This design combines the attentive neighbourhood aggregation of GAT, the efficient message passing of GraphSAGE, and the expressive capacity of MLP, with residual pathways to enrich smoother gradient flow.



From the results, the ResGAT-SAGE-MLP achieves the highest and most stable training and validation accuracy among all tested models, with training accuracy reaching nearly 0.59, validation accuracy stabilising just below that, and test accuracy achieving peaks around 0.53.

The loss curve drops more sharply and remains lower throughout training, indicating more effective fitting and a possibility of better optimisation dynamics compared to previous models.

The performance boost can be attributed to the introduction of residual connections (skip connections), which allow the model to combine both shallow and deep features. Despite this improvement, a small but persistent generalisation gap between training/validation and test accuracy still exists.

Comprehensive analysis

Now, let's proceed to the summary of this ablation study. In this section, we provide a comprehensive analysis of GAT models that are combined with other techniques or methods. We will maintain the core architecture of GAT while introducing additional techniques to observe the differences in performance.

Model	**Train Accuracy**	**Validation Accuracy**	**Test Accuracy**	**Final Loss**	**Notable Features**
GAT Baseline	~0.53–0.55	~0.53–0.55	~0.48–0.50	~2.55	3 GATConv layers
GAT-MLP	~0.54–0.56	~0.54–0.56	~0.50–0.52	~2.30	3 GAT + 3-layer MLP
GAT-SAGE-MLP	~0.56–0.58	~0.55–0.57	~0.51–0.53	~2.15	2 GAT, 1 SAGE, 3-layer MLP
ResGAT-SAGE-MLP	~0.58–0.59	~0.57–0.58	~0.52–0.54	~1.60	Residual, 2 GAT, 1 SAGE, MLP

The gradual improvement from GAT Baseline to ResGAT-SAGE-MLP highlights the benefit of hybrid and deeper architectures. Each additional module contributed to better generalisation and representation power, as seen in the steady increase in validation and test accuracy.
Residual connections in the final model were particularly effective in mitigating vanishing gradients and feature oversmoothing. This allowed the model to stack more layers without losing performance

While more complex models like GAT-SAGE-MLP and ResGAT-SAGE-MLP delivered improved accuracy, they also increased computational requirements, but these were still manageable thanks to mini-batch training with NeighborLoader and careful GPU memory management.

Despite improvements, none of the tested architectures reached state-of-the-art performance (>70% accuracy on OGBN-Arxiv). This suggests that further enhancements (deeper stacks, attention regularisation, normalisation layers, or advanced sampling strategies) could be explored.

Reference

- Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018). Graph Attention Networks. International Conference on Learning Representations (ICLR). https://arxiv.org/abs/1710.10903
- Hamilton, W., Ying, Z., & Leskovec, J. (2017). Inductive representation learning on large graphs. Advances in Neural Information Processing Systems, 30, 1024–1034. https://arxiv.org/abs/1706.02216
- Brody, S., Alon, U., & Yahav, E. (2022). How Attentive are Graph Attention Networks? International Conference on Learning Representations (ICLR). https://arxiv.org/abs/2105.14491
- Chen, D., Lin, Y., Li, W., Li, P., Zhou, J., & Sun, X. (2020). Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. Proceedings of the AAAI Conference on Artificial Intelligence, 34(4), 3438–3445. https://ojs.aaai.org/index.php/AAAI/article/view/5757
- Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., ... & Leskovec, J. (2020). Open Graph Benchmark: Datasets for Machine Learning on Graphs. Advances in Neural Information Processing Systems, 33, 22118–22133. https://arxiv.org/abs/2005.00687