Boosting ResGAT for Node Classification of Citation Network with Multi-hop & Motif-Based Attention

Yixiao Chen
Department of Statistics & Data Science
Yale University
New Haven, CT 06511
yixiao.chen@yale.edu

1 Introduction

In our project, we aim to solve the node classification problem of citation network using deep learning with graph-based methods. Accurate classification of academic papers is fundamental for efficient information retrieval and bibliometric analysis (Debackere, 2022; Ye, 2022). Recent studies have highlighted the advantages of leveraging inter-paper relationships, and many researchers have applied a combination of graph-based and attention-based learning methods for improved paper classification performance (Veličković et al., 2017; Zhang et al., 2022; Huang et al., 2024).

The graph-based paper classification problem can be formally defined as follows.

- **Citation Graph** - The citation graph is defined as $G = (V, E)$, where $V = \{v_1, v_2, \cdots, v_n\}$ represents a list of papers and $E \subset V \times V$ is the edge set. The nodes in the graph represent papers, and the edges represent relationships between the papers.
- **Node Features** - Given a paper $v_i \in V$, its features are defined as X_i, which may include information such as the paper's title, abstract, keywords, or other metadata.
- **Node Labels** - The label y_i for each paper, which indicates the classification of paper v_i. The label is derived based on specific categories or topics assigned to the paper.
- **Edges** - The edge set E represents relationships between papers, such as citation links or co-authorship, and is typically static for this problem.

The existing ResGAT model proposed by Huang et al. (2024) integrates a multi-head attention mechanism and a residual network structure to improve academic paper classification by effectively handling multi-level citation relationships. However, ResGAT's ability to capture rich, contextual information from multiple levels of the graph is limited. This project aims to develop a more accurate and generalized graph neural network model for paper classification by incorporating a multi-hop attention mechanism (Wang et al., 2020) and explicitly separates and processes relationships across different levels of aggregation. To further enhance the model's performance in learning complex graph structures, we incorporate a motif-based hybrid information matrix (Sheng et al., 2024) that take into account both first-order and higher-order structural information to compute the multi-hop attention computation. Ultimately, our results validate that the proposed approaches improve classification accuracy and generalization across multiple datasets.

Preprint. Under review.

2 Related Works

- **Graph Neural Networks:** Graph Neural Networks (GNNs) have recently achieved significant advancements in graph representation learning. Significant efforts on designing the architecture of GNN have been made to propagate messages differently. Graph Convolutional Networks (GCN) (Kipf and Welling, 2016) aggregate feature information from a node's neighbors using convolution operations for semi-supervised learning on graphs. GraphSAGE (Hamilton et al., 2017) extends GCN by sampling a fixed-size neighborhood and applying learnable aggregation functions to generate node embeddings, enabling inductive learning on large-scale graphs. Unlike these GNNs, which assume equal influence from neighboring nodes, the Graph Attention Network (GAT) (Veličković et al., 2017) utilizes an attention mechanism to learn the relative importance between connected nodes. Recent works (Li and Ouyang, 2024; Zhang et al., 2024) attempt to incroporate LLMs into the process of graph representation learning to further foster the performance and the generalized ability of GNNs.
- **GAT in Paper Classification & ResGAT:** Graph Attention Networks (GAT) are particular useful for paper classification with applying attention mechanisms to weigh the importance of neighboring nodes during feature aggregation. This mechanisms allow the model to focus more on influential citations while dynamically adjusting these weights. The effectiveness of GAT in paper classification is evidenced by numerous studies (Kim and Oh, 2019; Verma et al., 2023; Huang et al., 2024). Specifically, Huang et al. (2024) proposed ResGAT incorporates a multi-head attention mechanism along with a residual network structure to enhance academic paper classification by efficiently managing multi-level citation relationships.
- **Multi-hop Attention:** Wang et al. (2020) propose Multi-hop Attention Graph Neural Network (MAGNA), which incorporates multi-hop context information into each layer of GNN attention computation. MAGNA uses a diffusion-based approach to capture large-scale structural information effectively, achieving state-of-the-art performance on node classification and knowledge graph completion tasks. This technique are widely adopted in recent works (Wang et al., 2023; Deng and Huang, 2024; Gong et al., 2024) to improve the performance of GAT for various tasks.
- **Motif-based Attention:** After Sankar et al. (2017) proposed to use motif to create novel spatial GCN to capture high-order structural information, Peng et al. (2018) exploit motifs to capture local stationary and spatial structures of graphs to improve the performance of GNNs in graph classification tasks. Furthermore, Sheng et al. (2024) introduce demonstrate that incorporating motifs also improves the expressive power and robustness of graph neural networks in various node classification tasks.

3 Methodology

Our research methodology focuses on evaluating two distinct approaches for enhancing Residual Graph Attention Networks (ResGAT) by integrating motif-based adjacency matrices and multi-hop attention mechanisms. These advanced architectures are compared against traditional graph neural network baselines to assess their effectiveness in node classification tasks.

3.1 Proposed Architectures

3.1.1 Motif-based Residual GAT (M-ResGAT)

The first proposed architecture, M-ResGAT, incorporates higher-order structural patterns through motif-based adjacency matrices while leveraging residual connections for stable training. The core innovation lies in the hybrid attention mechanism that combines first-order neighbor relationships with motif information through a weighted matrix:

$$H = \beta \cdot A + (1 - \beta) \cdot A_M, \tag{1}$$

where:

- A represents the standard adjacency matrix capturing direct connections,
- $A_M = A^3$ is the motif-based adjacency matrix capturing triangle patterns,

- β is a hyperparameter balancing the contribution of each component.

This hybrid matrix H is then used to compute attention coefficients:

$$\alpha_{ij} = \text{softmax}(H_{ij} \cdot \text{LeakyReLU}(\mathbf{a}^\top [\mathbf{W}\mathbf{h}_i \,|\, \mathbf{W}\mathbf{h}_j])), \tag{2}$$

where $|$ denotes concatenation.

3.1.2 Multi-hop Residual GAT (MultiHop-ResGAT)

The second architecture focuses on capturing broader structural context through explicit multi-hop attention computation. For each layer l, the model computes attention scores across multiple hop distances:

$$\mathbf{h}_i^{(l)} = \sum_{k=1}^{K} \gamma_k \sum_{j \in \mathcal{N}_k(i)} \alpha_{ij}^k \mathbf{W}\mathbf{h}_j^{(l-1)}, \tag{3}$$

where:

- $\mathcal{N}_k(i)$ represents the k-hop neighborhood of node i,
- α_{ij}^k is the attention coefficient for k-hop neighbors,
- γ_k are learnable weights for different hop distances,
- K is the maximum hop distance considered.

3.2 Baseline Models

We compare our proposed architectures against four established baseline models: Graph Convolutional Network (GCN), Graph Attention Network (GAT), GraphSAGE, Residual GAT (ResGAT).

3.3 Evaluation Framework

To systematically evaluate the effectiveness of our proposed architectures, we assess their performance on node classification tasks using the following metrics:

- Classification Accuracy,
- Macro F1-Score,
- Area Under the ROC Curve (AUC).

All models are trained in a transductive setting, where the complete graph structure is available during training, but only a portion of node labels is used for supervision. This allows us to evaluate how effectively each architecture leverages both labeled and unlabeled data in the graph structure.

4 Experiments

4.1 Datasets

In this project, we will implement our novel approach with the three citation benchmark datasets used by Bojchevski and Günnemann (2017): CORA ML, CiteSeer, and CORA. The summary statistics of both datasets are shown in Table 1. The degree distribution and class distribution can be visualized in the Appendix (Figures 1 and 2).

4.2 Training Details

All experiments were conducted with the configurations shown in Table 2 in the Appendix.

4.3 Results

All experiments results were presents with the configurations shown in Table 3, Table 4, and Table 5 in the Appendix.

4.4 Analysis

Training Dynamics: Figure 3 shows the training and validation accuracy curves for all models for CORA dataset. Both M-ResGAT and MultiHop-ResGAT demonstrate faster convergence compared to baseline models, suggesting that incorporating higher-order structural information accelerates learning.

Model Comparison: Our experimental results in Table 3, Table 4, and Table 5 highlight distinct performance patterns across different evaluation metrics. On Cora-ML, M-ResGAT achieves competitive test accuracy (87.00%) and the highest AUC score (98.68%), while matching GraphSAGE's strong F1 performance (86.42%). For CiteSeer, MultiHop-ResGAT demonstrates superior performance with the highest test accuracy (96.11%) and F1 score (96.31%), though GCN achieves the best AUC score (99.71%). On the more challenging Cora dataset, MultiHop-ResGAT leads in both test accuracy (69.07%) and F1 score (64.00%), while M-ResGAT achieves the highest AUC (97.31%). These results suggest that MultiHop-ResGAT's explicit multi-hop attention mechanism is particularly effective for capturing complex node relationships in denser networks like CiteSeer, whereas M-ResGAT's motif-based approach excels at identifying broader class-specific patterns, as evidenced by its strong AUC performance across all datasets.

5 Conclusion

In this project, we introduced two novel architectures, M-ResGAT and MultiHop-ResGAT, which enhance traditional GAT models by incorporating different approaches to capturing higher-order structural information. Our comprehensive evaluation across three citation networks demonstrates the effectiveness of both approaches, with M-ResGAT achieving superior AUC scores on Cora-ML (98.68%) and MultiHop-ResGAT excelling in accuracy on CiteSeer (96.11%). These results validate our hypothesis that incorporating structural information beyond immediate neighborhoods can improve node classification performance. The complementary strengths of these architectures suggest that the choice between motif-based and multi-hop approaches may depend on specific dataset characteristics and performance priorities. The consistent performance across multiple runs, evidenced by low standard deviations in our metrics, demonstrates the reliability and stability of both approaches.

6 Limitations and Future Directions

Current Limitations:

- **Computational Efficiency:** M-ResGAT exhibits significantly longer training times, particularly on the Cora dataset (5247.53 $\pm$ 6.12 seconds) compared to baseline models (30–150 seconds).
- **Task Complexity:** The evaluation focused solely on node classification in citation networks, which may not fully demonstrate the models' capabilities on more complex tasks.

Future Directions:

- **Parameter Sensitivity Analysis:** Conduct comprehensive studies on the sensitivity of β parameter in M-ResGAT to optimize the balance between local and motif-based information.
- **Alternative Motif Structures:** Explore the effectiveness of different motif patterns beyond triangles, such as squares or directed patterns, which might better capture domain-specific structural information.
- **Scalability Enhancements:** Develop optimized implementations to reduce training time while maintaining performance, potentially through sparse matrix operations or selective motif computation.

7 Reproducibility

To ensure reproducibility of the results, we provide a complete GitHub repository containing all the code, configurations, and instructions required to replicate the experiments. You can access the repository at the following link: GitHub Repository for M-ResGAT.

Repository Contents:

- **README.md**: Includes detailed instructions on how to set up the environment, run the experiments, and interpret the results.
- **Requirements.txt**: Specifies all the dependencies required to run the project.
- **Main script**: The file train.py contains the code to produce the main results. Running this script will replicate the results reported in this paper.

References

Bojchevski, A. and Günnemann, S. (2017). Deep Gaussian Embedding of Graphs: Unsupervised Inductive Learning via Ranking. *arXiv*.

Debackere, W. G. &. K. (2022). Various aspects of interdisciplinarity in research and how to quantify and measure those. *Scientometrics*, 127(9):5551–5569.

Deng, P. and Huang, Y. (2024). Edge-featured multi-hop attention graph neural network for intrusion detection system. *Computers & Security*, 148:104132.

Gong, C., Wei, Z., Wang, X., Wang, R., Li, Y., and Zhu, P. (2024). Subgraph-based attention network for multi-hop question answering. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9.

Hamilton, W. L., Ying, R., and Leskovec, J. (2017). Inductive Representation Learning on Large Graphs. *arXiv*.

Huang, X., Wu, Z., Wang, G., Li, Z., Luo, Y., and Wu, X. (2024). ResGAT: an improved graph neural network based on multi-head attention mechanism and residual network for paper classification. *Scientometrics*, 129(2):1015–1036.

Kim, D. and Oh, A. (2019). Supervised Graph Attention Network for Semi-Supervised Node Classification. [Online; accessed 4. Oct. 2024].

Kipf, T. N. and Welling, M. (2016). Semi-Supervised Classification with Graph Convolutional Networks. *arXiv*.

Li, X. and Ouyang, J. (2024). Related Work and Citation Text Generation: A Survey. *arXiv*.

Peng, H., Li, J., Gong, Q., Wang, S., Ning, Y., and Yu, P. S. (2018). Graph Convolutional Neural Networks via Motif-based Attention. *arXiv*.

Sankar, A., Zhang, X., and Chang, K. C.-C. (2017). Motif-based Convolutional Neural Network on Graphs. *arXiv*.

Sheng, J., Zhang, Y., Wang, B., and Chang, Y. (2024). MGATs: Motif-Based Graph Attention Networks. *Mathematics*, 12(2):293.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. (2017). Graph Attention Networks. *arXiv*.

Verma, A. K., Saxena, R., Jadeja, M., Bhateja, V., and Lin, J. C.-W. (2023). Bet-GAT: An Efficient Centrality-Based Graph Attention Model for Semi-Supervised Node Classification. *Appl. Sci.*, 13(2):847.

Wang, G., Ying, R., Huang, J., and Leskovec, J. (2020). Multi-hop Attention Graph Neural Network. *arXiv*.

Wang, L., Li, Z.-W., You, Z.-H., Huang, D.-S., and Wong, L. (2023). MAGCDA: A Multi-Hop Attention Graph Neural Networks Method for CircRNA-Disease Association Prediction. *IEEE J. Biomed. Health Inf.*, 28(3):1752–1761.

Ye, H. Z. &. M. J. &. Y. Y. &. H. J. &. T. (2022). Recommendation method for academic journal submission based on doc2vec and XGBoost. *Scientometrics*, 127(5):2381–2394.

Zhang, J., Chen, J., Maatouk, A., Bui, N., Xie, Q., Tassiulas, L., Shao, J., Xu, H., and Ying, R. (2024). LitFM: A Retrieval Augmented Structure-aware Foundation Model For Citation Graphs. *arXiv*.

Zhang, L., Sun, B., Shu, F., and Huang, Y. (2022). Comparing paper level classifications across different methods and systems: an investigation of Nature publications. *Scientometrics*, 127(12):7633–7651.

A Appendix

A.1 Figures

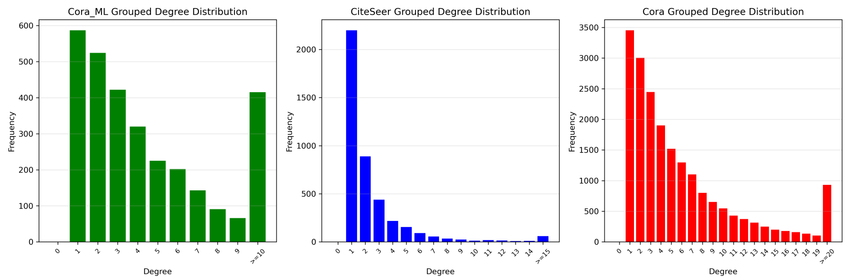

Figure 1: Grouped Degree Distribution Comparison for CORA ML Dataset, CiteSeer Dataset, and CORA dataset.

Figure 2: Class Distribution Comparison for CORA ML Dataset, CiteSeer Dataset, and CORA dataset.

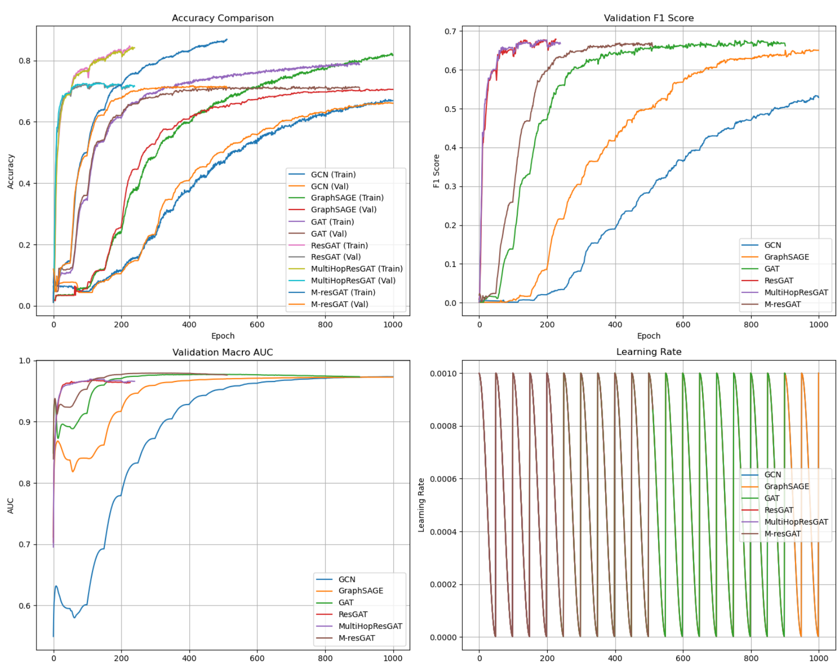

Figure 3: Training Dynamics: Training and validation accuracy for different models.

A.2 Tables

Dataset Statistics:

Dataset Statistics	CORA ML Dataset	CiteSeer Dataset	CORA Dataset
Graph Size			
Total number of nodes	2,995	4,230	19,793
Total number of edges	8,158	5,337	63,421
Number of isolated nodes	0	0	0
Node Classification			
Number of labeled nodes	2,995	4,230	19,793
Number of non-labeled nodes	0	0	0
Number of node categories	7	6	70

Table 1: Statistics of CORA ML Dataset, CiteSeer Dataset, and CORA dataset.

Training Details:

Aspect	Details
Model Architecture	
Hidden Dimensions	256
Number of Layers	3
Attention Heads	8
Dropout Rate	0.3
Training Parameters	
Optimizer	AdamW (learning rate = 0.001, weight decay = 1×10^{-4})
Learning Rate Scheduler	Cosine annealing with warm restarts
Early Stopping	Patience of 100 epochs
Maximum Epochs	1,000
Random Seeds	42, 43, 44
Data Splits	
Training	80%
Validation	10%
Test	10%

Table 2: Training Configurations

Node Classification Performance on Cora-ML:

Model	Test Acc (%)	Test F1	Test AUC	Time (s)
GCN	86.12 ± 0.19	84.72 ± 0.31	98.59 ± 0.01	6.57 ± 1.46
GraphSAGE	87.21 ± 0.00	86.30 ± 0.13	98.57 ± 0.03	4.97 ± 0.93
GAT	87.32 ± 0.68	86.67 ± 0.62	98.58 ± 0.04	6.47 ± 0.40
ResGAT	86.45 ± 0.51	85.44 ± 0.79	98.26 ± 0.06	2.30 ± 0.46
MultiHopResGAT	84.37 ± 0.68	82.97 ± 0.76	97.65 ± 0.21	3.63 ± 1.17
M-ResGAT	87.00 ± 0.18	86.42 ± 0.12	98.68 ± 0.01	14.00 ± 2.13

Table 3: Performance comparison on Cora-ML dataset (mean ± standard deviation over 3 runs)

Node Classification Performance on CiteSeer:

Model	Test Acc (%)	Test F1	Test AUC	Time (s)
GCN	95.25 ± 0.27	95.41 ± 0.27	99.71 ± 0.01	6.87 ± 1.37
GraphSAGE	95.95 ± 0.14	96.13 ± 0.15	99.59 ± 0.02	4.23 ± 0.25
GAT	95.33 ± 0.23	95.50 ± 0.23	99.58 ± 0.05	6.30 ± 1.87
ResGAT	95.41 ± 0.13	95.58 ± 0.14	99.60 ± 0.04	3.93 ± 0.95
MultiHopResGAT	96.11 ± 0.13	96.31 ± 0.12	99.51 ± 0.03	5.83 ± 2.17
M-ResGAT	95.33 ± 0.23	95.51 ± 0.25	99.49 ± 0.03	47.30 ± 9.97

Table 4: Performance comparison on CiteSeer dataset (mean ± standard deviation over 3 runs)

Node Classification Performance on Cora:

Model	Test Acc (%)	Test F1	Test AUC	Time (s)
GCN	63.86 ± 0.09	51.30 ± 0.14	96.90 ± 0.00	120.27 ± 2.22
GraphSAGE	68.32 ± 0.03	63.33 ± 0.07	97.10 ± 0.01	164.10 ± 1.48
GAT	68.86 ± 0.12	63.69 ± 0.21	97.22 ± 0.03	139.07 ± 8.05
ResGAT	68.49 ± 0.26	63.16 ± 0.20	95.78 ± 0.10	30.63 ± 1.15
MultiHopResGAT	69.07 ± 0.33	64.00 ± 0.16	96.10 ± 0.19	37.30 ± 4.10
M-ResGAT	68.31 ± 0.32	63.16 ± 0.29	97.31 ± 0.04	5247.53 ± 6.12

Table 5: Performance comparison on Cora dataset (mean ± standard deviation over 3 runs)