Project Proposal for CPSC 583

Gonghan Xu

1 Problem Definition

In the realm of Graph Neural Networks (GNNs), node classification remains one of the cornerstone tasks. Typically, GNN architectures focus on leveraging local graph structures, where nodes aggregate information from their neighboring nodes. Graph Attention Networks (GATs), for instance, employ an attention mechanism to weigh the importance of each neighbor during this aggregation. On the other hand, the Transformer architecture, originally designed for sequence data, have been useful for capturing complex relation within data through self-attention mechanisms.

The primary question driving this project is: Can the integration of GAT with a Transformer block lead to enhanced performance in node classification tasks, specifically in terms of accuracy?

This endeavor aims to explore and evaluate the potential benefits of such integration, combining the local aggregation power of GAT with the capability of Transformer in modeling intricate interactions. The two datasets chosen for this investigation are Cora and Citeseer, which offer substantial ground for assessing the performance of the proposed hybrid model.

2 Datasets

2.1 The Cora Dataset

The Cora dataset is a well-known benchmark dataset in the graph learning community. It comprises scientific publications, where each node represents a document and edges signify citations between these documents. The node features are based on word vectors, indicating the presence (or absence) of specific words in the document. Documents are classified into seven distinct classes based on their content.

Table 1: Statistics for the Cora dataset.

Statistic	Value
Number of Features	1433
Number of Classes	7
Number of Nodes	2708
Number of Edges	10556
Average Node Degree	3.90
Is Graph Directed	False

The t-SNE visualization for the Cora dataset (Figure 1) displays a diverse distribution of data points across different classes. There's noticeable clustering among certain classes, suggesting some similarity in feature space. However, significant overlap across the classes also highlights the intricacy of the dataset and the potential challenge of distinguishing between them based solely on node features.

2.2 The Citeseer Dataset

The Citeseer dataset is another standard benchmark dataset, similar in nature to the Cora dataset. It consists of scientific documents classified into six categories based on content. Nodes in this graph

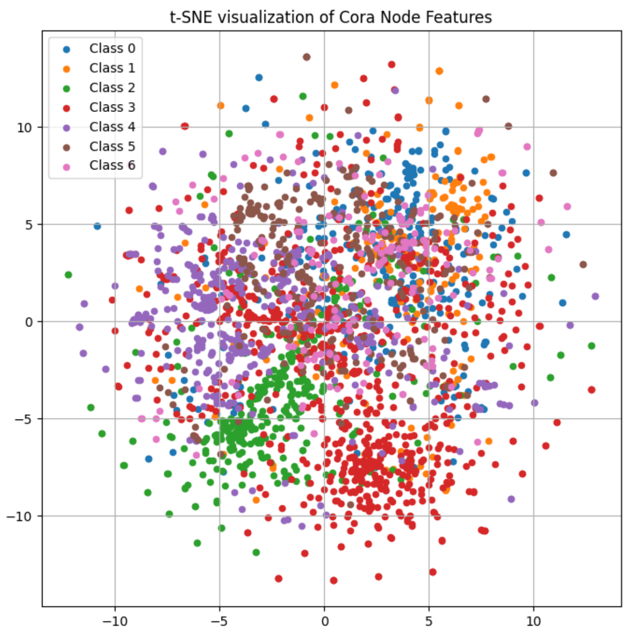

Figure 1: Visualization of the Cora dataset with t-SNE dimensionality reduction.

symbolize the documents, while edges denote citations. The node features, represented as word vectors, describe the presence or absence of particular terms in the document.

Table 2: Statistics for the Citeseer dataset.

Statistic	Value
Number of Graphs	1
Number of Features	3703
Number of Classes	6
Number of Nodes	3327
Number of Edges	9104
Average Node Degree	2.74
Is Graph Directed	False

From the t-SNE visualization for the Citeseer dataset (Figure 2), one can observe a more pronounced clustering of classes as compared to Cora. Despite this, some overlap still exists, particularly among a few class combinations. This shows the nuanced relationships and potential similarities among certain document groups in the dataset.

Overall, given the statistics and visualizations, the two datasets present an intricate structure with high-dimensional feature space. This complexity calls for sophisticated models, such as the proposed combination of GAT and Transformer layers, to achieve a high classification performance.

3 Description of Related Works

Graph-based learning has been a focal point of research with many significant contributions over recent years. Herein, we outline some of the pivotal works that have shaped the field and provide a contextual landscape for our proposed methodology.

Yang et al.[1] introduced the Planetoid, a semi-supervised learning method for graph-based classification. The model's primary strength lies in its capability to learn from both labeled and unlabeled data and it provides the first classification benchmark on the Cora and Citeseer datasets. The Planetoid model achieved an accuracy of 75.7% on the Cora dataset and 64.7% on Citeseer.

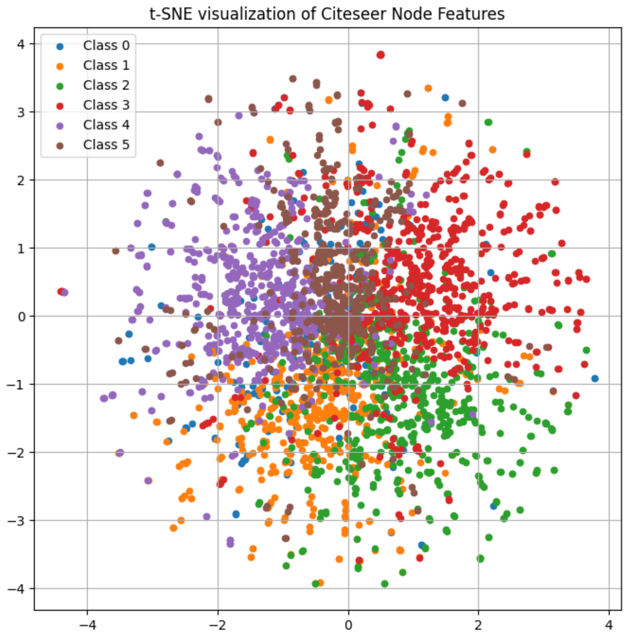

Figure 2: Visualization of the Citeseer dataset with t-SNE dimensionality reduction.

Kipf and Welling[2] presented the Graph Convolutional Network (GCN), a seminal work that simplifies the training process of deep networks on graph data. The GCN leverages a localized first-order approximation of spectral graph convolutions to operate directly on graph structures. It showcased an impressive accuracy of 81.5% on Cora and 70.3% on Citeseer.

Veličković et al.[3] introduced the Graph Attention Network (GAT), which incorporates attention mechanisms into graph neural networks. By assigning different attention scores to different nodes, GAT offers a more adaptive aggregation method compared to its predecessors. The model achieved notable accuracies of 83.0% on Cora and 72.5% on Citeseer.

Yun and collaborators[4] proposed the Graph Transformer Networks (GTN) that integrate the self-attention mechanism of Transformers directly within the GNN's message-passing process. This deep integration captures both local and global node dependencies. While this work exemplifies the potential of combining Transformers with graph neural networks, it's noteworthy that our proposed methodology distinguishes itself by incorporating the Transformer block post-aggregation, rather than as a part of the aggregation itself.

4 Proposed Approaches

Our core objective is to investigate whether the addition of Transformer blocks to GATs enhances the node classification accuracy on individual datasets like Cora and Citeseer.

So our baseline approach is Graph Attention Networks (GATs). GATs leverage attention mechanisms to weigh the importance of neighboring nodes during the aggregation phase. This ensures that more relevant neighbors have a greater influence on the central node's updated feature representation.

Our novel approach is to add a Transformer block post-aggregation. For our graph-based classification task, the Transformer block can be integrated after GAT layers, treating the aggregated node features from GATs as sequences. This can potentially capture higher-order dependencies among node features. So our novel model will be an integrated GAT-Transformer Model.

Each model will be trained from scratch on both the Cora and Citeseer datasets separately. We will use standard train-validation-test splits provided with the datasets and employ early stopping based on validation performance to find the appropriate hyper-parameters and to prevent overfitting. Given the potential complexity of the hybrid model, regularization techniques like dropout may be used to mitigate overfitting.

After evaluating the performances of the baseline approach and the novel approach on the two datasets, we will compare their performance to identify the benefits (or none) of the Transformer integration.

5 Evaluation Metrics

To evaluate the performance of our models on node classification tasks for datasets like Cora and Citeseer, we will use classfication accuracy on the test set as our primary evaluation metric. Accuracy is a direct reflection of how many nodes were correctly classified out of the total nodes. It provides a straightforward and intuitive measure of a model's performance, especially when the class distribution is relatively balanced, as is the case for our chosen datasets. Moreover, accuracy is a widely accepted and commonly used metric in the node classification literature, making our results easily comparable to prior works.

In addition, to better understand the learned embeddings, we can use dimensionality reduction techniques (e.g., t-SNE) to visualize the embeddings in 2D space, comparing the clustering quality of node embeddings from the hybrid model vs. the baseline GAT.

6 Timeline

Week 1 (the first week after fall break): Deep dive into primary papers and understand GATs and Transformer nuances.

Week 2: Implement and validate the baseline GAT model on Cora and Citeseer datasets.

Week 3: Develop and test the integrated GAT-Transformer model architecture.

Week 4: Train GAT-Transformer model on datasets and optimize hyperparameters.

Week 5: Evaluate model performance and analyze results.

Week 6: Compile findings and finalize the project report.

References

[1] Zhilin Yang, William Cohen, and Ruslan Salakhudinov. "Revisiting semi-supervised learning with graph embeddings". In: *International conference on machine learning*. PMLR. 2016, pp. 40–48.

[2] Thomas N Kipf and Max Welling. "Semi-supervised classification with graph convolutional networks". In: *arXiv preprint arXiv:1609.02907* (2016).

[3] Petar Velickovic et al. "Graph attention networks". In: *stat* 1050.20 (2017), pp. 10–48550.

[4] Seongjun Yun et al. "Graph transformer networks". In: *Advances in neural information processing systems* 32 (2019).