PushNet to Attentive PushNet

Harsh Pundhir

VU Amsterdam

Abstract. This study introduces Attentive PushNET, an enhancement of the PushNET framework for neural message passing in graph neural networks (GNNs) through the integration of attention mechanisms. By dynamically weighting the importance of neighboring nodes' information, Attentive PushNET aims to improve the efficiency and adaptability of information propagation across complex graph structures. Experimental evaluations on widely recognized datasets, CORA and CiteSeer, demonstrate the superior performance of Attentive PushNET compared to its predecessor, PushNET. These findings suggest that incorporating attention into neural message passing frameworks can significantly advance the field of graph-based learning.
Keywords: Graph Neural Networks, Neural Message Passing, Attention Mechanisms, PushNET, Attentive PushNET, Graph Learning

1 Introduction

Graphs are a fundamental way to represent entities and their relationships in various domains like social networks, citation networks, and biological networks. The recent advent of graph neural networks (GNNs) has revolutionized learning on these graph-structured data. Traditional GNN approaches, such as Graph Convolutional Networks (GCNs), aggregate information from a node's neighborhood, often leading to limitations like fixed locality and ineffective modeling of long-range dependencies [2].

PushNet, an innovative variant in the field of neural message passing, addresses these challenges. It employs an asynchronous, push-based message-passing algorithm, allowing information to propagate on demand rather than indiscriminately pulling from all neighbors. This method not only enhances the efficiency but also reduces noise in learned node features. By leveraging Approximate Personalized PageRank for neighborhood determination, PushNet adapts to the specific source node, enabling it to capture multi-scale node representations effectively [1].

This report explores the integration of attention mechanisms into PushNet, envisioning a system where the information propagation is not only adaptive but also contextually aware. The aim is to combine the strengths of PushNet with the dynamic focusing ability of attention mechanisms, potentially overcoming the limitations of existing GNNs in handling complex graph structures and data variances.

2 Related Works

Graph Neural Networks (GNNs) have become a focal point of research for learning on graph-structured data, enabling applications in various domains including social network analysis, recommender systems, and biological network interpretation. The success of GNNs is largely attributed to their ability to capture complex dependencies within graph data through neural message passing mechanisms. Among these, the PushNet model presents a novel asynchronous message passing approach that significantly improves efficiency and adaptability over traditional synchronous message passing methods.

2.1 Neural Message Passing in Graphs

Traditional GNN models, such as Graph Convolutional Networks (GCN) by Kipf and Welling [2], rely on synchronous message passing where nodes aggregate information from their immediate neighbors. This process, although effective, limits the receptive field to immediate neighbors and requires stacking multiple layers to reach distant nodes, potentially leading to oversmoothing issues. PushNet, introduced by Busch et al. [1], addresses these limitations by implementing an asynchronous message passing mechanism, allowing for adaptive and efficient information propagation across the graph.

2.2 Attention Mechanisms in GNNs

The integration of attention mechanisms into GNNs has been explored to enhance model performance further by allowing nodes to dynamically weigh the importance of their neighbors' information. The Graph Attention Network (GAT) by Veličković et al. [3] represents a significant advancement in this area, demonstrating the utility of attention in learning node representations that are contextually informed by the importance of neighboring nodes. This approach has inspired subsequent research into more sophisticated attention-based models that adaptively capture node interactions within graphs.

2.3 PushNet

The foundational principles of PushNet provide a robust framework for adaptive message passing. However, the integration of attention mechanisms, as demonstrated in GAT and other works, suggests a promising avenue for enhancing PushNet's capabilities. Attentive PushNet, by incorporating attention, aims to leverage the strengths of PushNet's efficient propagation mechanism while further refining the adaptiveness and selectivity of message passing based on the relevance of the information being propagated. This hybrid approach seeks to marry the efficiency and adaptability of PushNet with the dynamic importance weighting afforded by attention mechanisms, potentially setting a new standard for performance in tasks involving complex graph-structured data.

3 Background

Graph Neural Networks are designed to process data represented in graph form. A graph $G = (V, E)$ consists of a set of vertices V and a set of edges E where each edge connects a pair of vertices. GNNs leverage the graph structure to learn node representations by aggregating features from neighboring nodes. The aggregation function, often parameterized by neural networks, allows GNNs to capture the local graph topology around each node. Mathematically, the feature vector of a node v_i k-th layer of a GNN can be expressed as:

$$h_i^{(k)} = \sigma\left(W^{(k)} \cdot AGG\left(\{h_j^{(k-1)} : j \in N(i)\}\right) + b^{(k)}\right)$$

(1)

where $h\imath^{(k)}$ denotes the feature vector of node vi at layer k, $W(k)$ and $b(k)$ are the weight matrix and bias vector of the k-th layer, respectively, AGG is an aggregation function that combines features from the neighbors N(i) of node v_i and σ is a non-linear activation function.

3.1 PushNet Architecture

PushNET introduces a novel approach to neural message passing by emphasizing asynchronous, demand-driven propagation of information across the graph. Unlike traditional GNNs that aggregate information synchronously from immediate neighbors, PushNET selectively "pushes" information through the most relevant paths in the graph until convergence.

The key innovation of PushNET lies in its formulation of the message-passing process. For a given node $v_i$the propagation of its message to a neighboring node $v_j$is determined by the relevance of v_j to v_i's message, enabling adaptive learning of node representations that are sensitive to the global graph structure. The message update rule in PushNET can be abstracted as follows:

$$m_{ij}^{(k)} = \phi\left(h_i^{(k-1)}, h_j^{(k-1)}; \theta^{(k)}\right)$$

(2)

$$h_i^{(k)} = h_i^{(k-1)} + \sum_{j \in N(i)} m_{ij}^{(k)}$$

(3)

where $m_{ij}^{(k)}$ is the message sent from node v_i to node v_j at layer k, $h_i^{(k)}$ is the feature vector of node v_i at layer k, and ϕ is a learnable function parameterized by $\theta^{(k)}$ that computes the message based on the features of v_i and v_j.

3.2 Attention Mechanism

Attention mechanisms allow a model to dynamically focus on the most relevant parts of the input data. In the context of GNNs, attention can be used to weigh the contributions of neighboring nodes differently during the aggregation process, enhancing the model's ability to capture nuanced patterns in the graph data. The attention coefficient between two connected nodes v_i and v_j can be computed as:

$$\alpha_{ij} = \frac{\exp\left(LeakyReLU\left(a^T[Wh_i||Wh_j]\right)\right)}{\sum_{k\in N(i)}\exp\left(LeakyReLU\left(a^T[Wh_i||Wh_k]\right)\right)}$$

(4)

where α_{ij} represents the attention coefficient, indicating the importance of node v_j's features to node v_i, W is a shared linear transformation applied to each node's features, $\mathbf{a}$ is a learnable vector that computes the attention coefficient, and $\|\cdot\|$ denotes concatenation. Integrating attention into PushNET, dubbed Attentive PushNET, involves modifying the message-passing mechanism to incorporate these attention coefficients, allowing the model to adaptively focus on the most informative edges in the graph during information propagation.

4 PushNET vs. Attentive PushNET

4.1 PushNET

PushNET introduces an innovative approach to neural message passing in graph neural networks (GNNs). Unlike traditional GNNs, which rely on synchronous aggregation of features from a node's immediate neighbors, PushNET employs an asynchronous message passing strategy that allows for selective and demand-driven propagation of information across the graph. This method enhances both the efficiency and adaptability of the learning process, particularly in handling large-scale and complex graph structures.

The architecture of PushNET is centered around the *Local Push Message Passing (LPMP)* algorithm, which dynamically adjusts the flow of information by pushing messages along the most relevant edges based on a set of convergence criteria. Mathematically, the update rule for a node's feature vector in PushNET can be expressed as follows:

$$H^{(k+1)} = H^{(k)} + \alpha\Phi^{(k)}F^{(k)} - \Phi^{(k+1)}F^{(k+1)}$$

where $H^{(k)}$ represents the aggregated feature matrix at iteration k, $\Phi^{(k)}$ is the matrix of unprocessed information, $F^{(k)}$ denotes the feature matrix, and α is the restart probability.

4.2 Attentive PushNET

Building upon the foundational principles of PushNET, Attentive PushNET integrates attention mechanisms into the message passing strategy to further refine the model's ability to discern and prioritize the most informative messages during propagation. By leveraging attention, Attentive PushNET aims to enhance the model's performance by providing a more nuanced understanding of the graph's structure and the relationships between nodes.

The key component of Attentive PushNET is the *AttentionLayer*, which computes attention scores for each node based on the relevance of its neighbors' information. The attention scores are then used to weight the messages being propagated, allowing the model to focus on the most significant messages at each step of the information flow as also described in the equation 4 above.

4.3 Comparative Analysis

The integration of attention mechanisms in Attentive PushNET represents a significant enhancement over the original PushNET model. While PushNET's asynchronous message passing strategy already offers a level of adaptability and efficiency superior to traditional synchronous GNNs, Attentive PushNET takes these advantages further by adding a layer of selectivity in the propagation process. This selectivity allows Attentive PushNET to focus computational resources on processing and propagating the most relevant information, potentially leading to improvements in learning accuracy and model performance on complex tasks. However, the inclusion of attention mechanisms also introduces additional computational complexity to the model. The calculation of attention scores requires extra computations which, depending on the size and density of the graph, could impact the overall efficiency of the model. Despite this, the potential benefits in terms of enhanced model performance and adaptability may outweigh the costs, especially in applications where the precision of the learned representations is critical.

5 Methodology

5.1 Datasets Used for Evaluation

For the evaluation of PushNET and Attentive PushNET, two well-established datasets, CORA and CiteSeer, were selected. These datasets are benchmarks in the domain of graph neural networks, particularly in the context of semi-supervised node classification tasks.

CORA: Consists of scientific publications categorized into one of seven classes. The graph is constructed by treating each publication as a node and citations between publications as edges. This dataset is particularly challenging due to its sparse citation network and the high dimensionality of node features.

CiteSeer: Similar to CORA, CiteSeer is a citation network with publications classified into six categories. The graph's nodes represent documents, and edges represent citation links. CiteSeer poses unique challenges due to its diverse topics and citation patterns.

Table 1: Summary of Datasets Used for Evaluation

Dataset	Number of Nodes	Number of Edges
CORA	2,708	5,429
CiteSeer	3,327	4,732

5.2 Experimental Setup

The experimental setup involves the implementation of PushNET and Attentive PushNET models, followed by their training and evaluation on the CORA and CiteSeer datasets. The process is facilitated by the PyTorch Geometric library, however the LPMP message massing was made from scratch. Other details can be seen in table 2.

Note: **Even though small alpha and epsilon values are recommended, I picked larger values for a little quick computation. Actual research should focus on small values of the aforementioned parameters.

Table 2: Parameter Ranges Used in Experiments

Parameter	Range
Alpha (α)	0.5 – 0.9
Epsilon (ϵ)	0.5 – 0.9
Epochs	200
Learning Rate	0.01

Training Procedures: Both models are trained using the Local Push Message Passing (LPMP) function to aggregate features from neighboring nodes. The primary difference is the incorporation of an attention mechanism in Attentive PushNET, which dynamically weighs the contributions of neighboring nodes during feature aggregation. The attention architecture implemented in Attentive PushNET utilizes an *AttentionLayer* that dynamically computes attention scores for each node's features. This is achieved through a softmax operation on linear transformations of the features, allowing the model to weigh the importance of neighbor nodes' information selectively.

Evaluation Metrics: The performance of both models is assessed using precision, recall, F1 score, and accuracy. These metrics provide a comprehensive understanding of the models' effectiveness in classifying nodes correctly within the graph. To facilitate reproducibility and tracking of experimental results, the

WandB platform is utilized. WandB allows for logging hyperparameters, training progress, and evaluation metrics, thereby enabling a systematic comparison between PushNET and Attentive PushNET under various configurations.

Code Implementation: The methodology incorporates code snippets for data loading, model initialization, training, and evaluation. For instance, the LPMP function is critical for feature aggregation in both models, while the `AttentionLayer` class is specific to Attentive PushNET, enabling the computation of attention scores for feature weighting. The evaluation process involves running the models on the test set of each dataset and computing the aforementioned metrics to gauge performance.

6 Results

For both Cora and Citeseer datasets, the models employing attention mechanisms (Attention = 1) significantly outperform those without (Attention = 0) across all metrics. This indicates the effectiveness of attention mechanisms in improving model accuracy and overall performance as seen in table 3. Moreover, the consistency in performance improvement with the addition of attention mechanisms (as evidenced by the increase in metrics from Attention = 0 to Attention = 1) suggests that attention mechanisms contribute to better model generalization by effectively capturing and utilizing relevant information. When comparing across datasets, the models applied to the Citeseer dataset generally achieve higher Accuracy, Precision, Recall, and F1_Score than those applied to the Cora dataset.

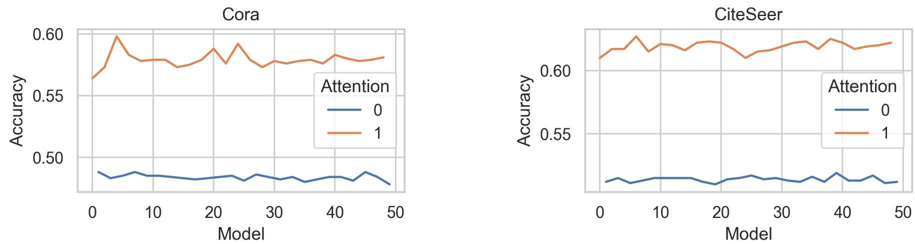
An instance of change in accuracy is evident from the figure 1. There is a clear demarcation between the classes of attention = 1 (with attention) and Attention = 0 (without attention).

Table 3: Averaged Metrics for Cora and Citeseer datasets

Dataset	Attention	Accuracy	Precision	Recall	F1_Score
Cora	0	0.48 (0.00)	0.48 (0.00)	0.52 (0.00)	0.48 (0.00)
Cora	**1**	**0.58 (0.01)**	**0.56 (0.00)**	**0.60 (0.00)**	**0.56 (0.01)**
Citeseer	0	0.51 (0.00)	0.51 (0.00)	0.51 (0.00)	0.50 (0.00)
Citeseer	**1**	**0.62 (0.00)**	**0.59 (0.00)**	**0.59 (0.00)**	**0.59 (0.00)**

6.1 Effect of alpha and epsilon

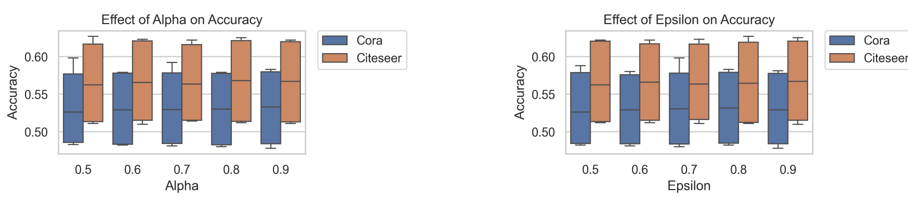
Increasing the values of Epsilon and Alpha does not correlate well with an improvement in accuracy for both datasets as shown in figure 2. The Cora dataset consistently demonstrates higher accuracy compared to Citeseer across all parameter values. While the accuracy's variability remains fairly consistent across



(a) Cora dataset (b) Citeseer dataset

Fig. 1: Visualizations of the Cora and Citeseer datasets.

different values of Epsilon and Alpha, there is a slight indication that higher Alpha values may introduce more variability in the Cora dataset. Overall, the positive trend in accuracy with increasing parameter values suggests that these parameters are influential in the performance of the models assessed.



(a) alpha (b) epsilon

Fig. 2: Effect of Alpha and Epsilon

7 Conclusion

The results indicate that the Attentive PushNET model, which incorporates attention mechanisms, consistently outperformed the baseline PushNET model across both the CORA and CiteSeer datasets. This enhancement underscores the effectiveness of attention mechanisms in improving the adaptability and precision of message passing in graph neural networks. Interestingly, variations in the parameters α and ϵ did not significantly impact the overall performance of the models, suggesting that the architecture's robustness and the attention layer's capability to prioritize relevant information play a more crucial role in determining model performance than these parameters. Consequently, the introduction of attention mechanisms in PushNET presents a promising avenue for future research and application in graph-based learning tasks.

References

1. Julian Busch, Jiaxing Pi, and Thomas Seidl. Pushnet: Efficient and adaptive neural message passing. *arXiv preprint arXiv:2003.02228*, 2020.
2. Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.
3. Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.