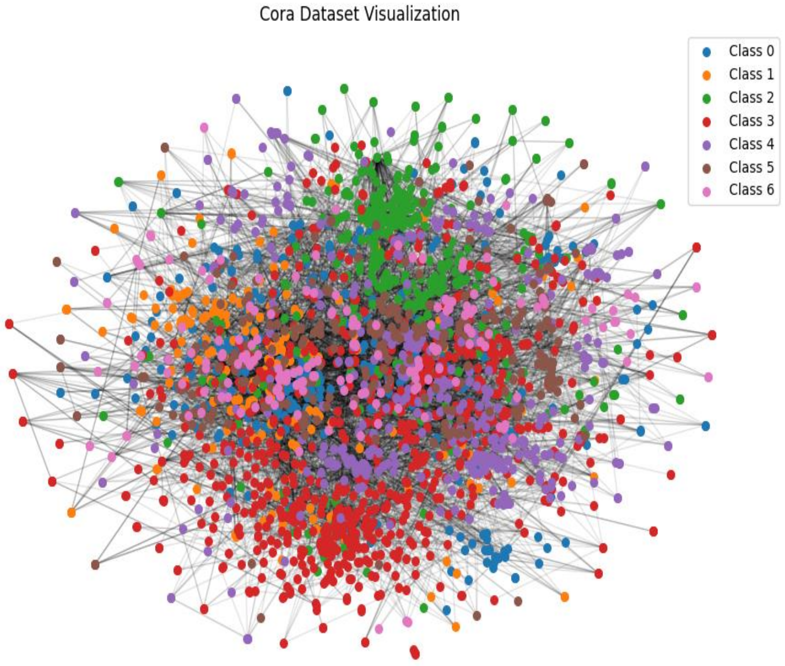
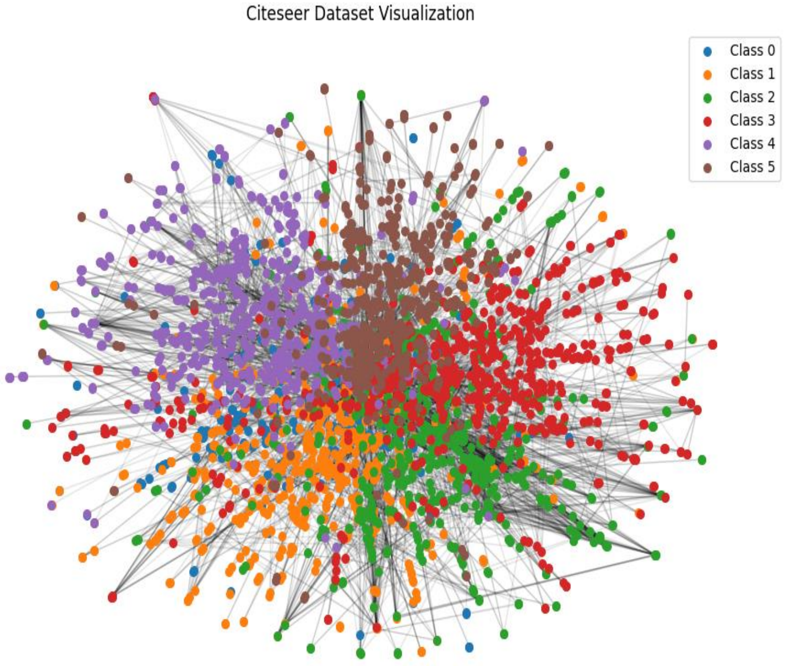
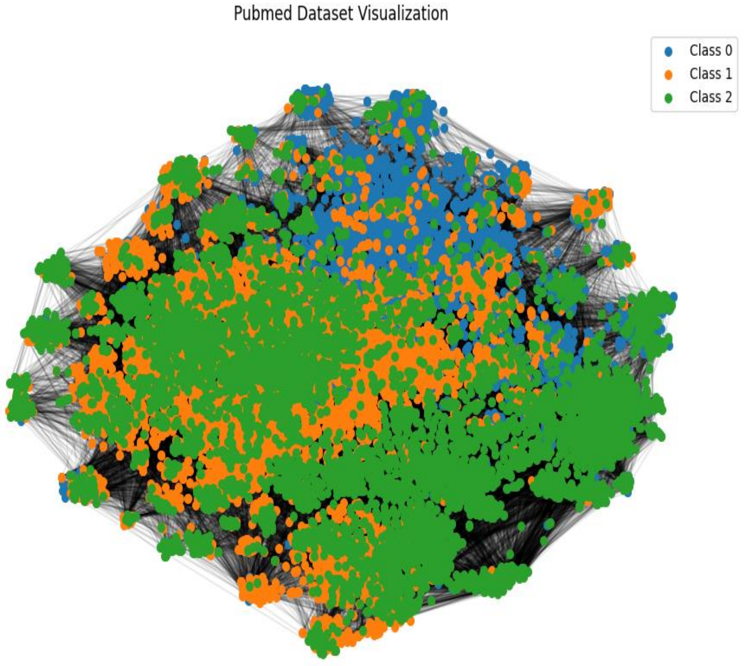
GNN Model Architecture

By: Arghodeep Nandi
2024EEZ8395

Datasets








```
--- Properties for Cora ---
Number of Nodes: 2708
Number of Edges: 5278
Average Degree: 3.8981
Density: 0.0014
Transitivity: 0.0935
Average Clustering Coefficient: 0.2407
Diameter: 19
Average Degree Centrality: 0.0014
Assortativity Coefficient: -0.0659
Homophily Score: 0.8100
```

```
--- Properties for CiteSeer ---
Number of Nodes: 3279
Number of Edges: 4552
Average Degree: 2.7765
Density: 0.0008
Transitivity: 0.1301
Average Clustering Coefficient: 0.1435
Diameter: 28
Average Degree Centrality: 0.0008
Assortativity Coefficient: 0.0484
Homophily Score: 0.7355
```

```
--- Properties for PubMed ---
Number of Nodes: 19717
Number of Edges: 44324
Average Degree: 4.4960
Density: 0.0002
Transitivity: 0.0537
Average Clustering Coefficient: 0.0602
Diameter: 18
Average Degree Centrality: 0.0002
Assortativity Coefficient: -0.0436
Homophily Score: 0.8024
```

GCN

$$\mathbf{X}' = \hat{\mathbf{D}}^{-1/2}\hat{\mathbf{A}}\hat{\mathbf{D}}^{-1/2}\mathbf{X}\mathbf{\Theta},$$

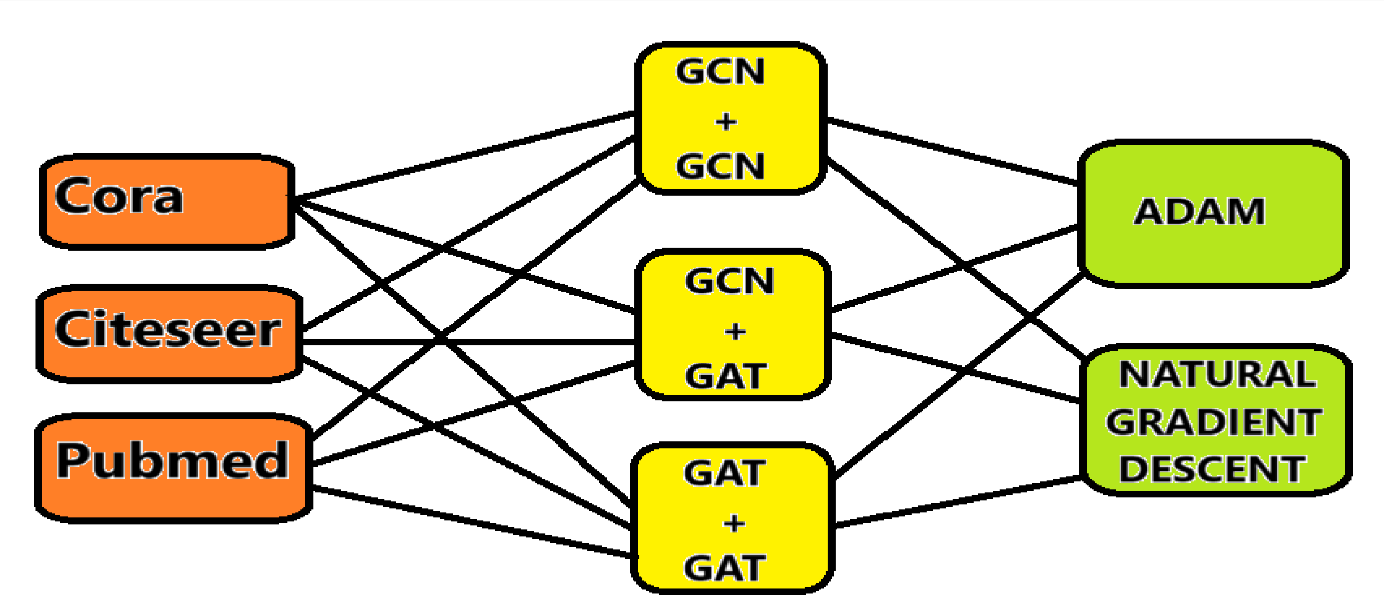
$$\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$$

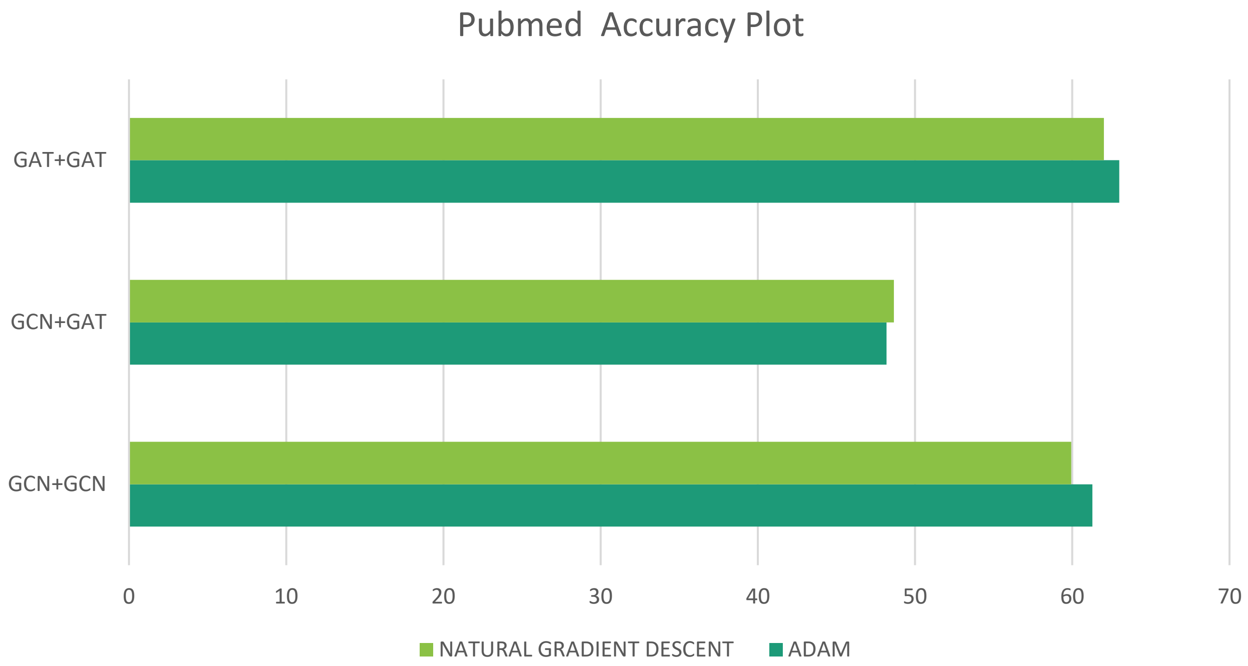
$$\hat{D}_{ii} = \sum_{j=0} \hat{A}_{ij}$$

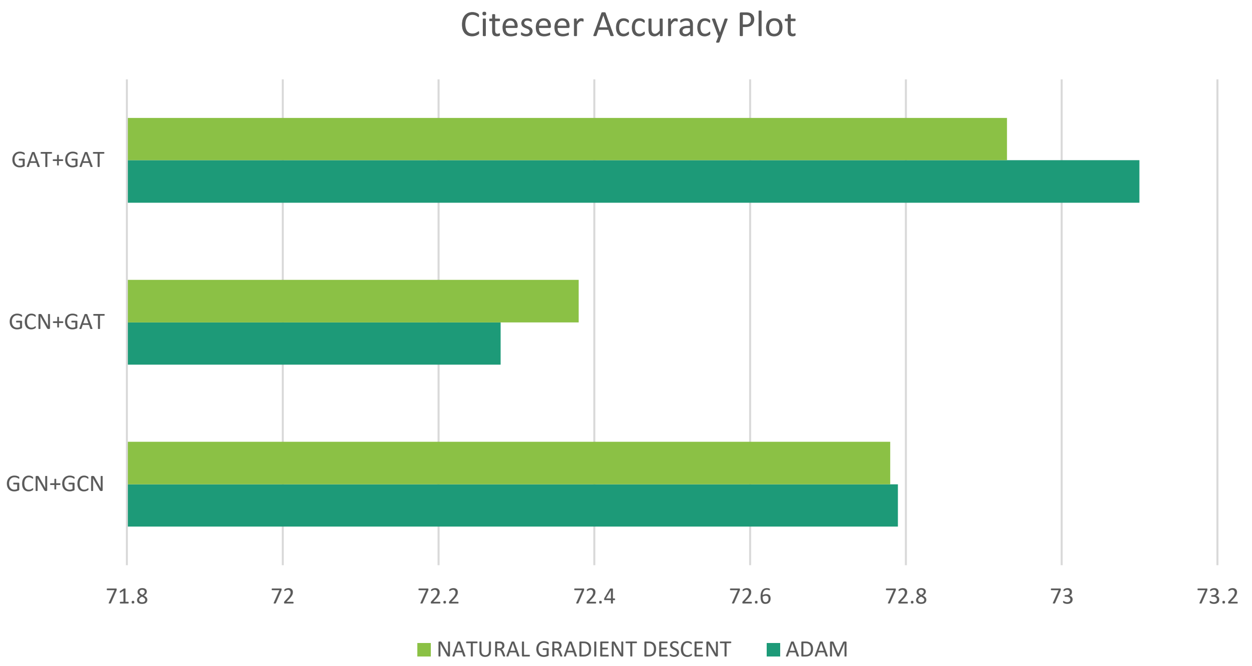
GAT

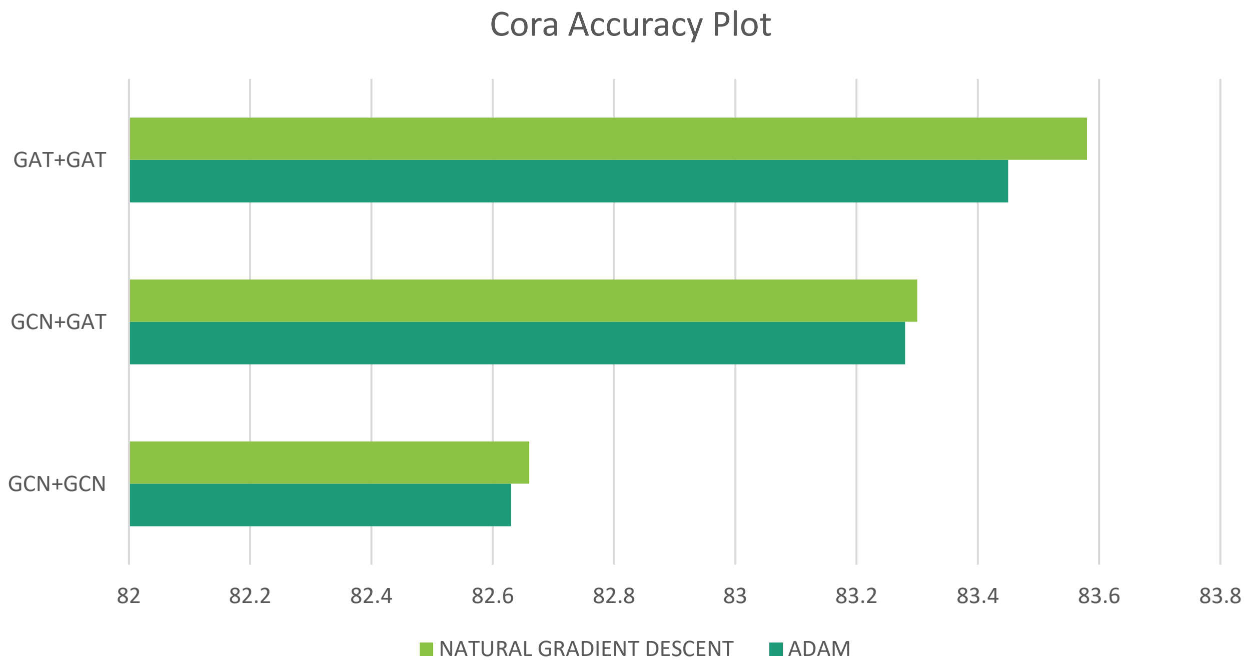
$$\mathbf{x}'_i = \alpha_{i,i}\mathbf{\Theta}_s\mathbf{x}_i + \sum_{j \in \mathcal{N}(i)} \alpha_{i,j}\mathbf{\Theta}_t\mathbf{x}_j,$$

$$\alpha_{i,j} = \frac{\exp\left(\mathrm{LeakyReLU}\left(\mathbf{a}_s^\top \mathbf{\Theta}_s\mathbf{x}_i + \mathbf{a}_t^\top \mathbf{\Theta}_t\mathbf{x}_j\right)\right)}{\sum_{k \in \mathcal{N}(i) \cup \{i\}} \exp\left(\mathrm{LeakyReLU}\left(\mathbf{a}_s^\top \mathbf{\Theta}_s\mathbf{x}_i + \mathbf{a}_t^\top \mathbf{\Theta}_t\mathbf{x}_k\right)\right)}$$



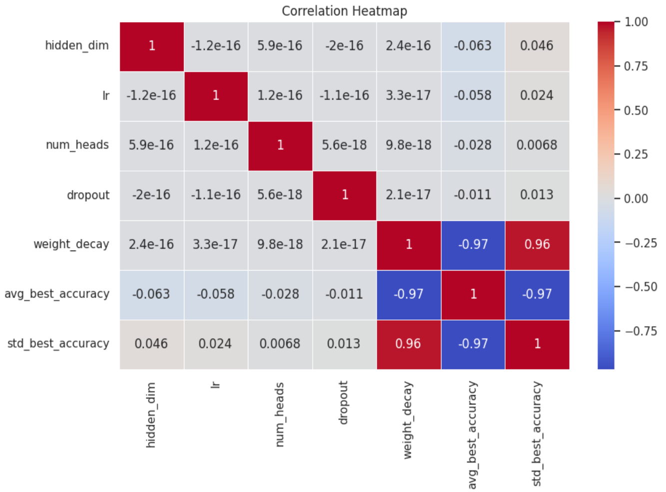




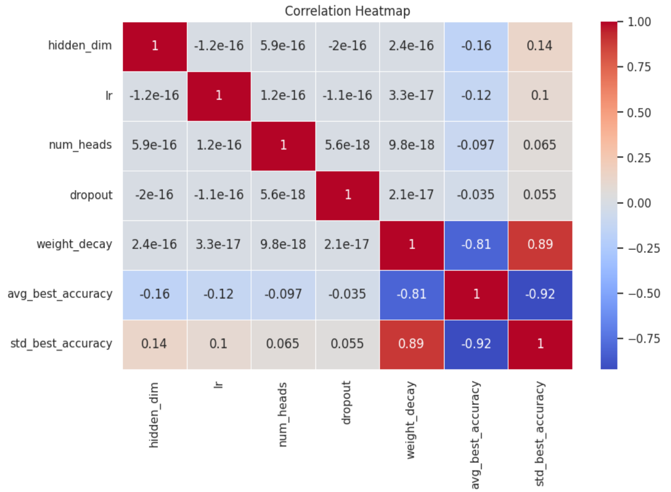


hidden_dim	**128**
dropout	**0.5**
num_heads	**16**
ADAM_LR	**0.005**
NGD_LR	**0.01**
weight_decay	**5.00E-02**
damping	**0.001**

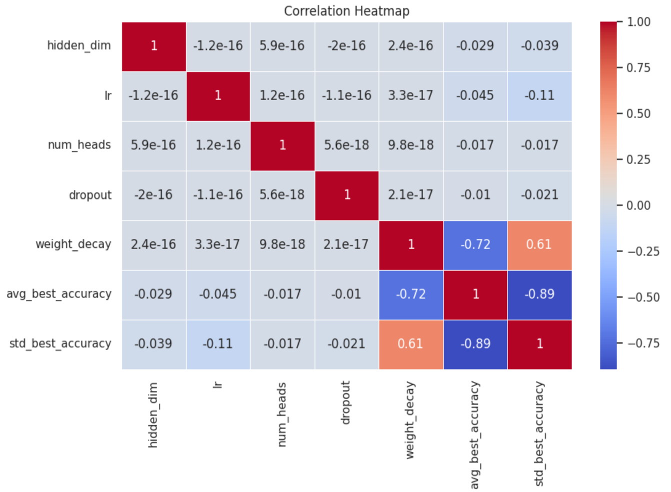
Hyperparameter Analysis (GAT-GCN, 540 M)



Cora



Citeseer



Pubmed

- **Weight Decay has the maximum influence on accuracy of the model irrespective of the dataset.**

	Max	Min	Percentage Difference
Cora	0.83255	0.49745	40.24983484
Citeseer	0.7265	0.58055	20.08947006
Pubmed	0.79065	0.4332	45.20963764

Hyperparameters playing a significant role in accuracy.

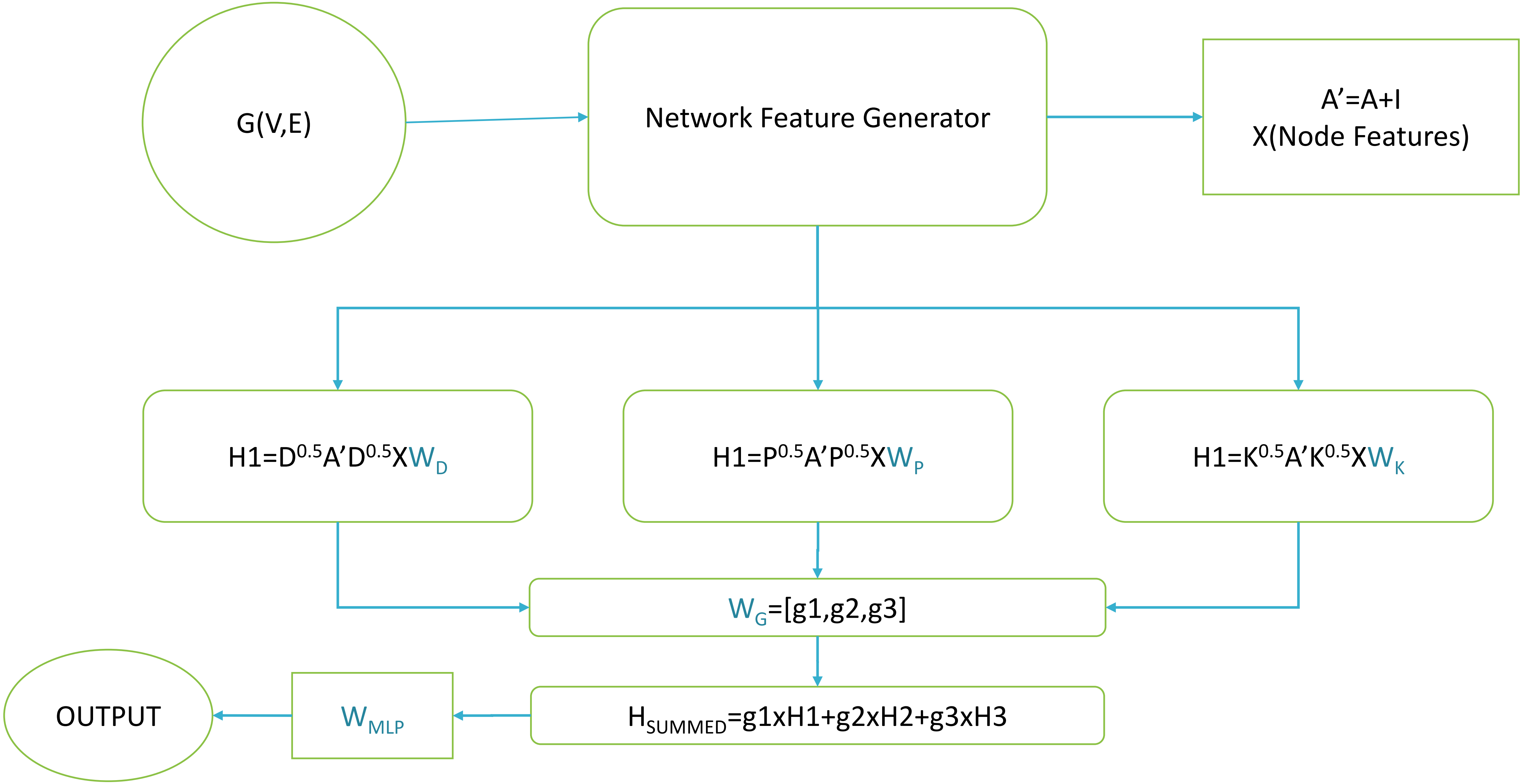
```
# Hyperparameters grid
hidden_dims = [16, 32, 64, 128, 256]
lrs = [0.01, 0.015, 0.02]
num_heads_list = [4, 8, 16]
dropouts = [0.3, 0.4, 0.5]
weight_decays = [5e-1, 5e-2, 5e-3, 5e-4]
```

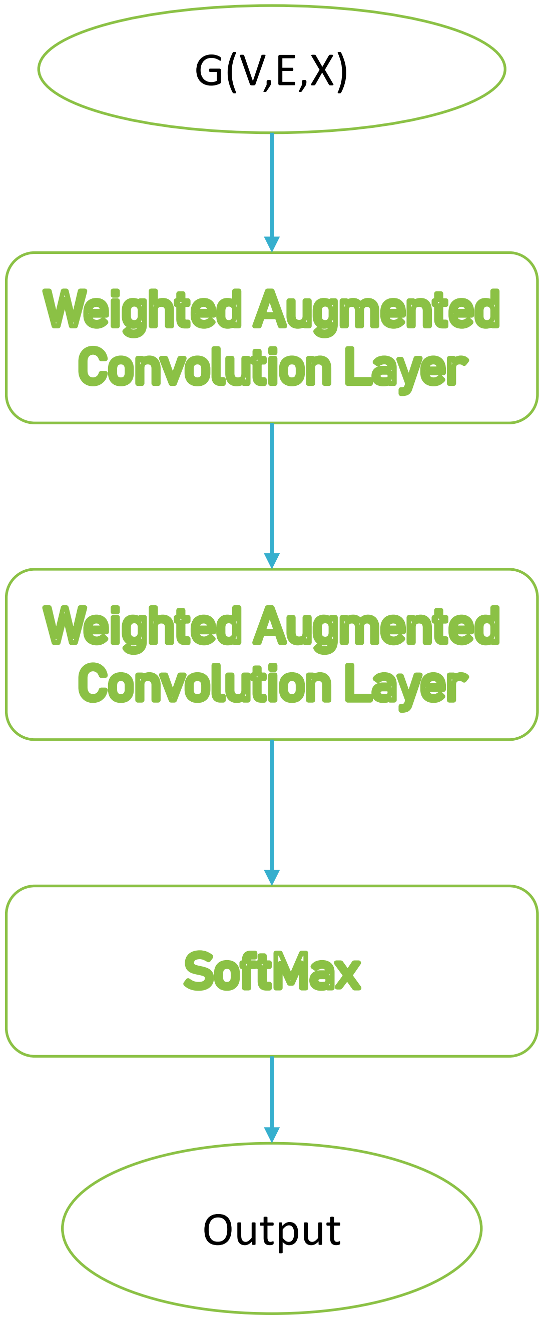
Column1	Cora	Citeseer	Pubmed
hidden_dim	64	256	256
lr	0.015	0.015	0.015
num_heads	4	4	8
dropout	0.5	0.3	0.3
weight_decay	0.05	0.05	0.005

Best case hyperparameters.

MODEL DEVELOPMENT

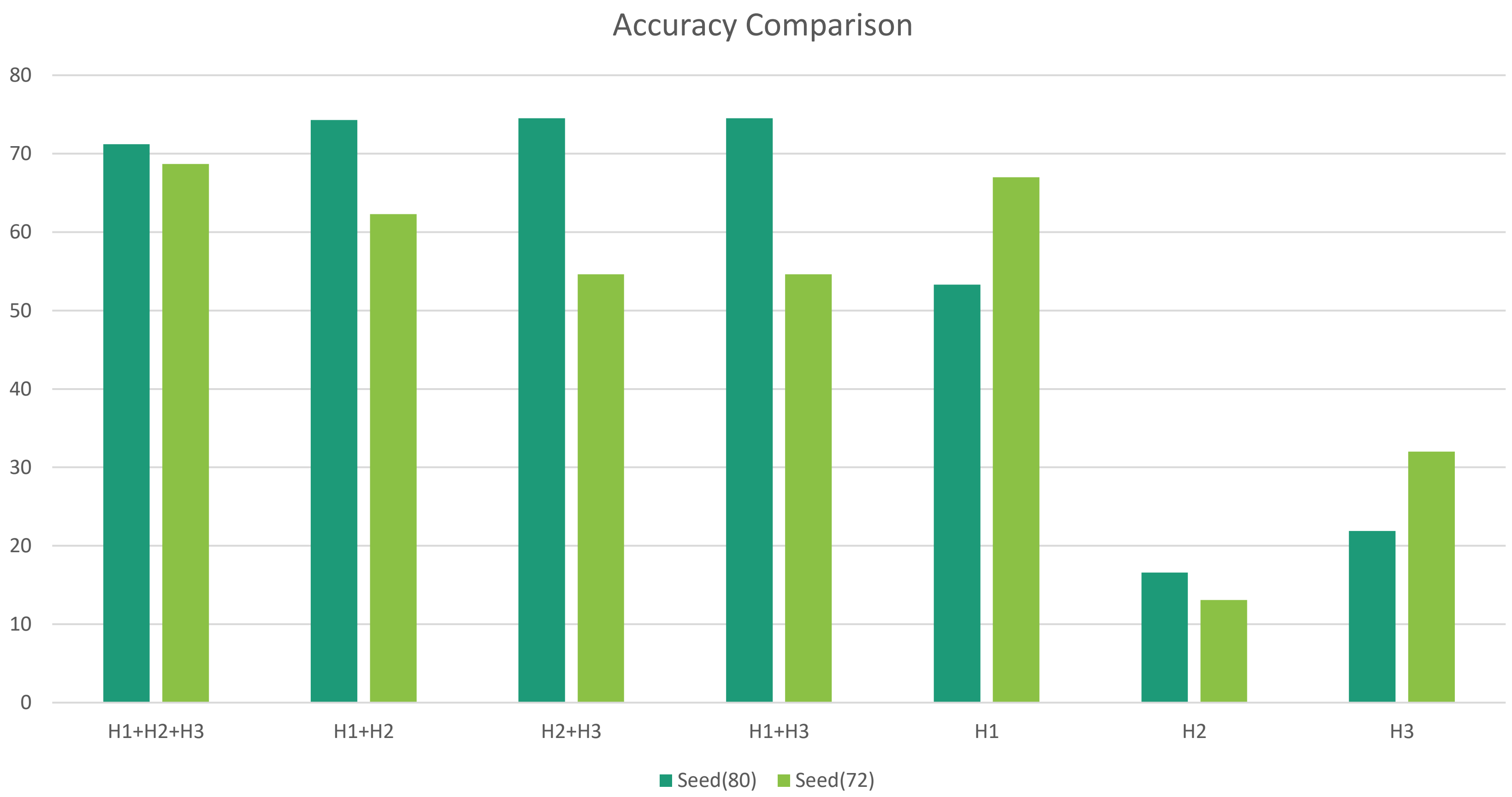
Weighted Augmented Convolution Layer






```
#cora
#set_seed(80)

    #epochs = 1000
    #learning_rate = 0.0005
    #weight_decay = 8e-3  # Weight decay for regularization
    #hidden_features = 2048  # Number of hidden features in the intermediate layer
    #degrees + 5e-6, same with pagerank and katz
    #71.20
```



Additional Models

Node and Features: Friend or Foe

$X_{NxF} = W1_{NxN}\, X_{NxF}\, F_{FxF} + \alpha_{NxN}\, (A\text{-}I)_{NxN}\, W2_{NxN}\, X_{NxF}\, F_{FxF}$

$\alpha_{NxN} = \text{Softmax}(QK^T)$

$F_{FxF} = \text{Softmax}(Q^TK)$

Trainable matrices: W1,W2,Q,K

Total Trainable Parameters: $2(N^2+NxF)$

Here we try to train both node and features using the same set of Q,K matrices.

If success is achieved, greater implications: only N trainable matrices will be required for N! weight matrices.

But easier said than done... The same element of a matrix cannot increase and decrease simultaneously.

Result: Accuracy=31.90%

Conclusion and Way Forward

- Through this project, we were not only able to analyze some of the well known models, but also gained some insights on the model architecture that is stable and hyperparameters that matter.

- We experimented on some model architecture and found a model which gives an accuracy of **71.20%**. The model is novel and implemented from scratch using the message passing layer module, but further modifications are required to come near the standard benchmarks.

- We also tried a different innovative approach to decrease the number of training parameters significantly. The accuracy is not good, but this intuition can be a base for further research and innovation.

- The cross-entropy loss for Deep Learning Models are highly Non-Convex. Results in high number of local minima/maxima. *Different seed values which causes different starting points for optimization algorithms, affect the final accuracy.*

- The loss function needs to be revamped to eliminate a large number of local minima/maxima. *One way can be to create a saddle point at those points.*

References

- Kipf, T. N., & Welling, M. (2017). *Semi-Supervised Classification with Graph Convolutional Networks.* In Proceedings of the International Conference on Learning Representations (ICLR).
- Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2018). *Graph Attention Networks.* In Proceedings of the International Conference on Learning Representations (ICLR).
- Ying, Z., Cai, T., Luo, S., Zheng, S., Ke, G., He, D., & Liu, T.-Y. (2021). *Do Transformers Really Perform Bad for Graph Representation?* Advances in Neural Information Processing Systems (NeurIPS).
- Liu, Y., Khanduri, P., & Leskovec, J. (2021). *Optimization of Graph Neural Networks with Natural Gradient Descent.* In Proceedings of the International Conference on Machine Learning (ICML).
- McCallum, A. K., Nigam, K., Rennie, J., & Seymore, K. (2000). *Automating the Construction of Internet Portals with Machine Learning.* Information Retrieval, 3(2), 127–163.
- Sen, P., Namata, G., Bilgic, M., Getoor, L., Gallagher, B., & Eliassi-Rad, T. (2008). *Collective Classification in Network Data.* AI Magazine, 29(3), 93–106.
- Namata, G. M., London, B., Getoor, L., & Huang, B. (2012). *Query-driven Active Surveying for Collective Classification.* In Proceedings of the 10th International Workshop on Mining and Learning with Graphs (MLG).