Understand Representation Power of Graph Neural Network on Graphs

Group G1
Zhenhan Huang, Anubhav Halder, Shiuli Subhra Ghosh, Alanta Kwok

Introduction

Graphs plays an important role in our daily lives

- Social networks (e.g. Facebook)
- Utility networks (e.g. power grid)
- Neural networks (e.g. brain neurons)
- etc.

Image credit: https://www.microsoft.com/en-us/research/group/networking-research/

Introduction

Characterizing the network is essential for understanding complex system

For example:

In power grid, attacking node with highest degree can destroy the power grid much faster compared to random attacking

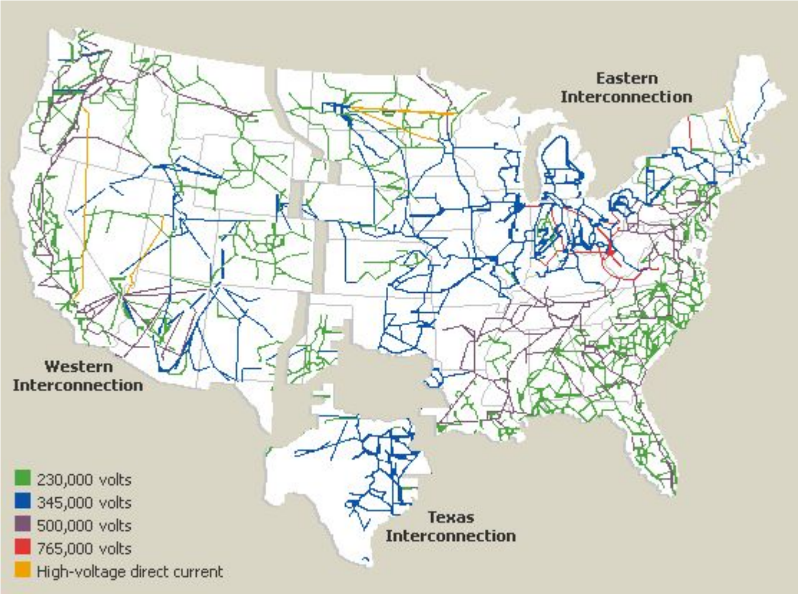
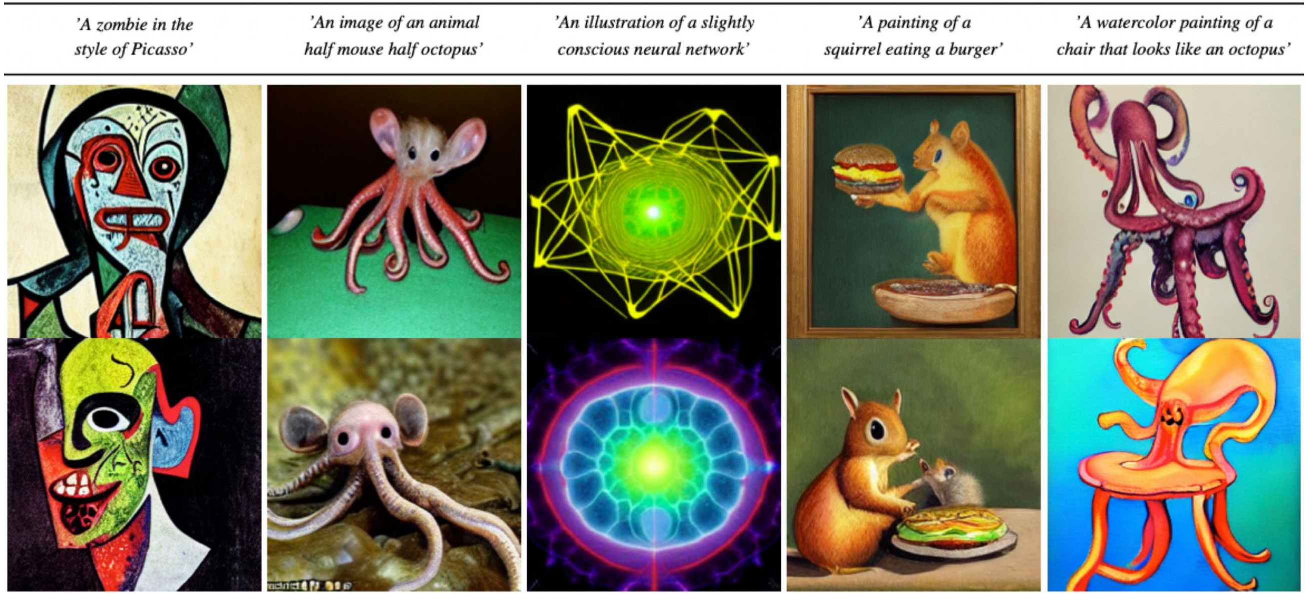


Image credit: https://alternativeenergy.procon.org/questions/what-is-the-electricity-grid/

Introduction

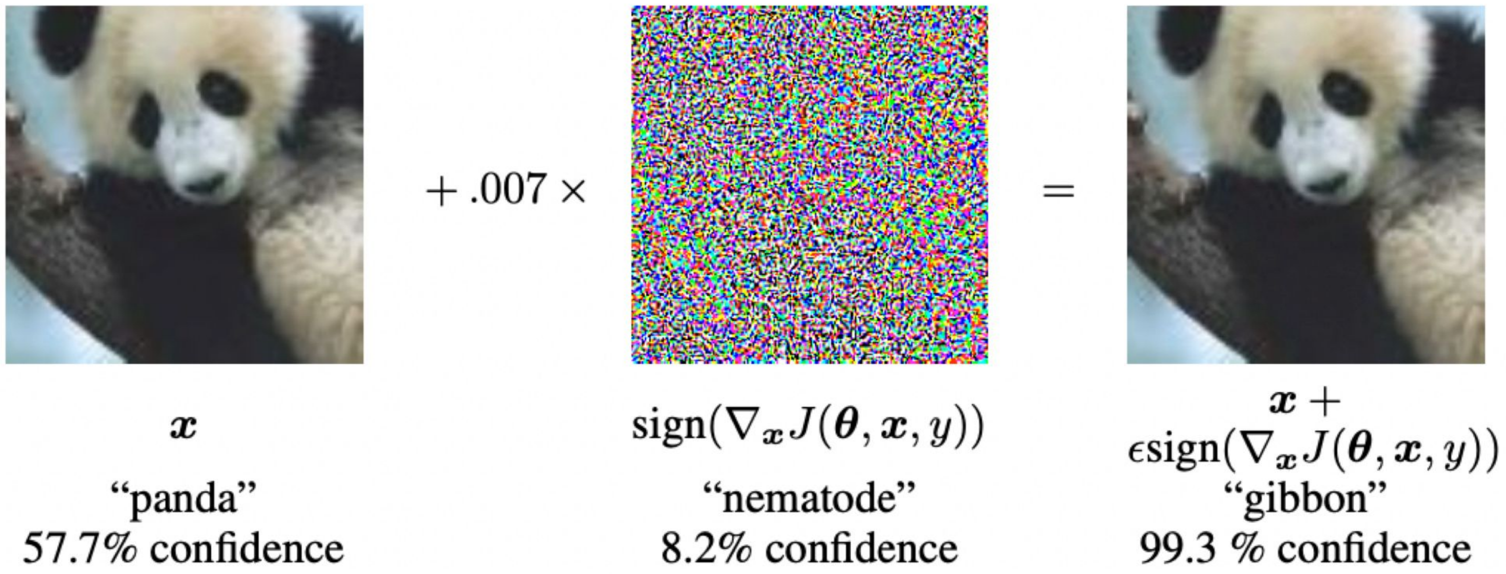
Deep learning makes lots of achievements in computer vision, natural language processing, etc.

Rogers, Everett M. "A prospective and retrospective look at the diffusion model." Journal of health communication 9.S1 (2004): 13-19.



Introduction

Deep learning can be unpredictable



Rogers, Everett M. "A prospective and retrospective look at the diffusion model." Journal of health communication 9.S1 (2004): 13-19.

Introduction

Deep learning is not omnipotent: it is outperformed by conventional data-driven methods when the dataset size is small.

Example: tabular dataset

Artificial characters: 6000 instances

Common datasets for deep learning models:

CIFAR-10: 50,000 training images, 10, 000 test images
CIFAR-100: 50,000 training images, 10, 000 test images
ImageNet: 1,281,167 training images, 50,000 validation images and 100,000 test images

Our dataset: 5,000 synthetic graphs (train set - validation set split: 4:1)

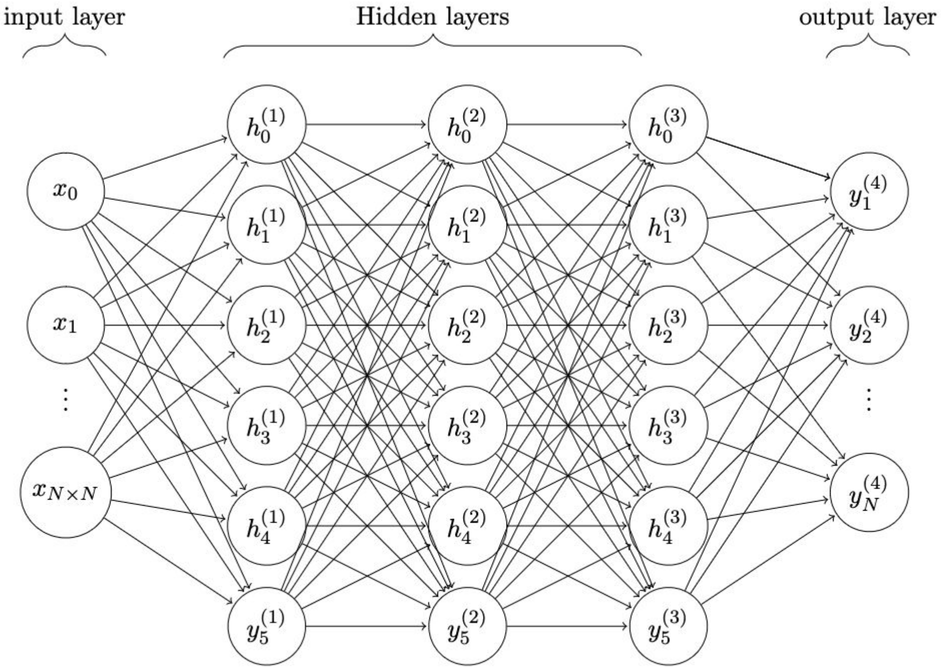
Methods

Baseline:

Multilayer Perceptron (MLP)

- 3-layer
- 1, 2, 4 hidden units per layer

- Sigmoid activation
- Adam optimizer
    - Learning rate: 0.001
    - Beta 1: 0.9
    - Beta 2: 0.999



Methods

Model comparison:

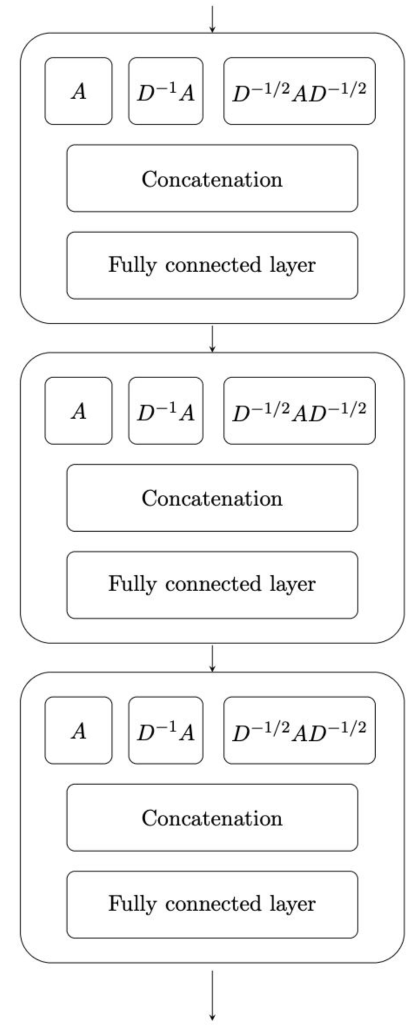
Graph Convolution Network (GCN)

- Layer configurations: 1, 2, 4
- Three different propagation rules:

1) A

2) $D^{-1}A$

3) $D^{-1/2}AD^{-1/2}$

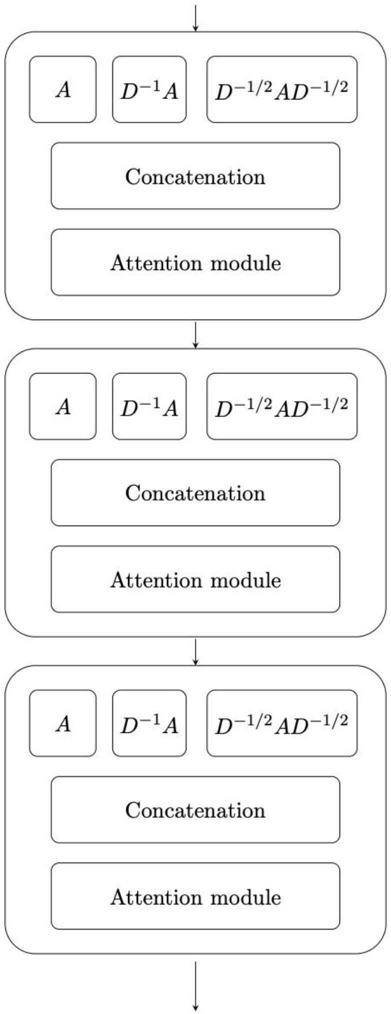


Methods

Model comparison:

GCN with Attention module

- Layer configurations: 1, 2, 4
- Attention mechanism:
  - Self-attention using equation $\frac{Q^T K}{\sqrt{n}} V$



Chosen Graph Properties

- Degree Centrality
- Clustering Coefficient
- Eccentricity
- Page Rank
- Average Neighbor Degree
- Average Shortest Path Length

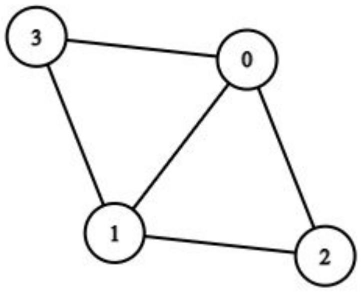
Choosing Criteria

- Easy to calculate and have physical understanding
- Have routines built into NetworkX library, which uses conventional state-of-the-art methods.

Degree Centrality

Degree centrality of a node is **the fraction of nodes it is connected to**.

It is a measure of **the number of neighbors** each node has.

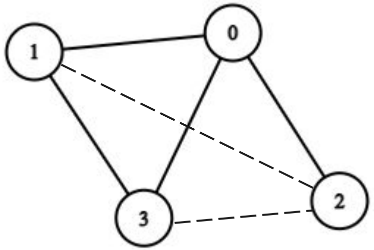


Node 0:	3	Divide by	Node 0:	1
Node 1:	3	max(degree)	Node 1:	1
Node 2:	2	=3	Node 2:	2/3
Node 3:	2		Node 3:	2/3

Clustering Coefficient

The fraction of **existing triangles through that node over all possible triangles** through that node.

It is a measure of how **tightly connected the neighborhood** of the node is.

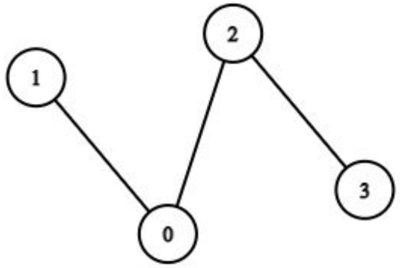


For node 0: $$\frac{\text{Existing } \Delta}{\text{Possible } \Delta} = \frac{1}{3}$$

Eccentricity

The **maximum shortest distance** from a node to all the other nodes.

It is a measure of how **far the node is from the center of the graph.**

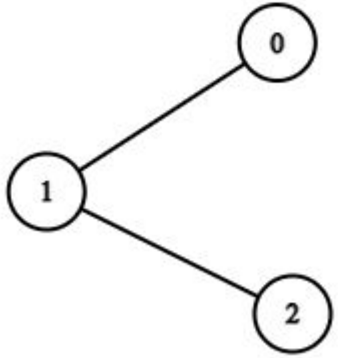


Node 0:	max(1,1,2)	=	2
Node 1:	max(1,2,3)	=	3
Node 2:	max(1,1,2)	=	2
Node 3:	max(1,2,3)	=	3

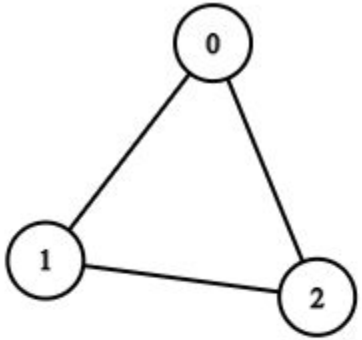
Page Rank

The page ranks of nodes are used to **order their relative importance**.

It is calculated by finding **the dominant eigenvector of the Google matrix**. This matrix is a modified version of the adjacency matrix of the graph.



PR(Node 0) = PR(Node 2) < PR(Node 1)

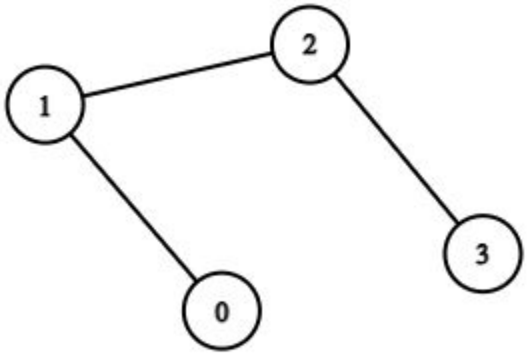


PR(Node 0) = PR(Node 2) = PR(Node 1)

Average Neighbor Degree

The **average degree of the neighborhood** of each node.

It is a **measure of the heterogeneity** (if the graph forms any **clusters of nodes**) of the network.

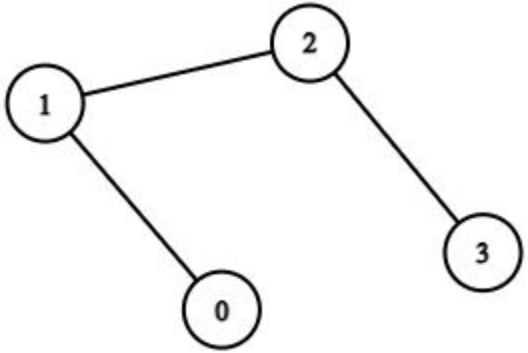


Degree:		Neighbors:		Avg. Nei. Deg.:	
Node 0:	1	Node 0:	Node 1	Node 0:	2
Node 1:	2	Node 1:	Nodes 0, 1	Node 1:	1.5
Node 2:	2	Node 2:	Nodes 1, 3	Node 2:	1.5
Node 3:	1	Node 3:	Node 2	Node 3:	2

Average Shortest Path Length

It is the **average of the shortest paths** from a node to all the other nodes.

It measures **how efficiently information can be transported** across the network.



For Node 1, the shortest path lengths are:

To node 0: 1

To node 2: 1

To node 3: 2

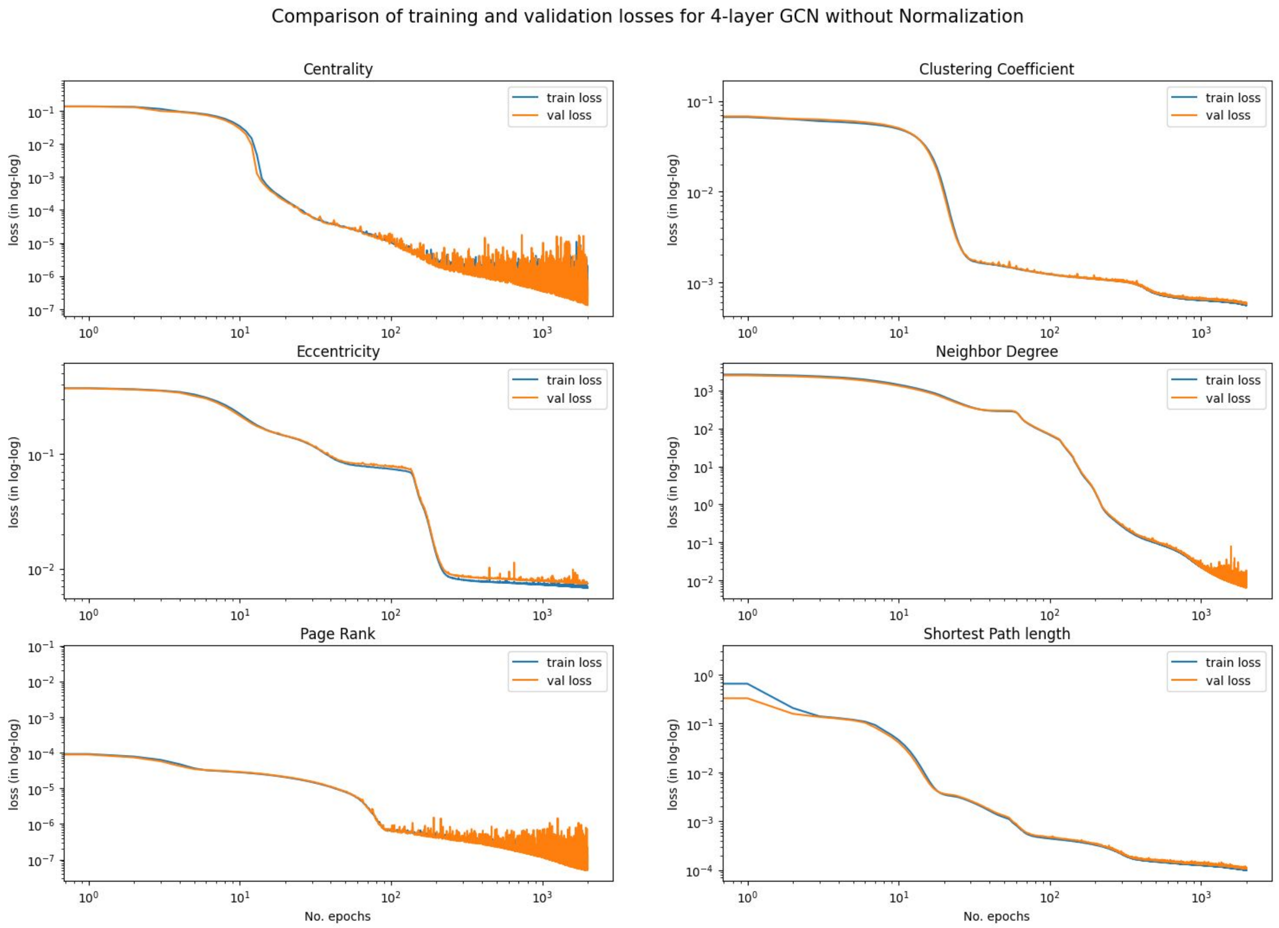
So, Average shortest path length is: 1.33

Time and Space Complexity (NetworkX routines)

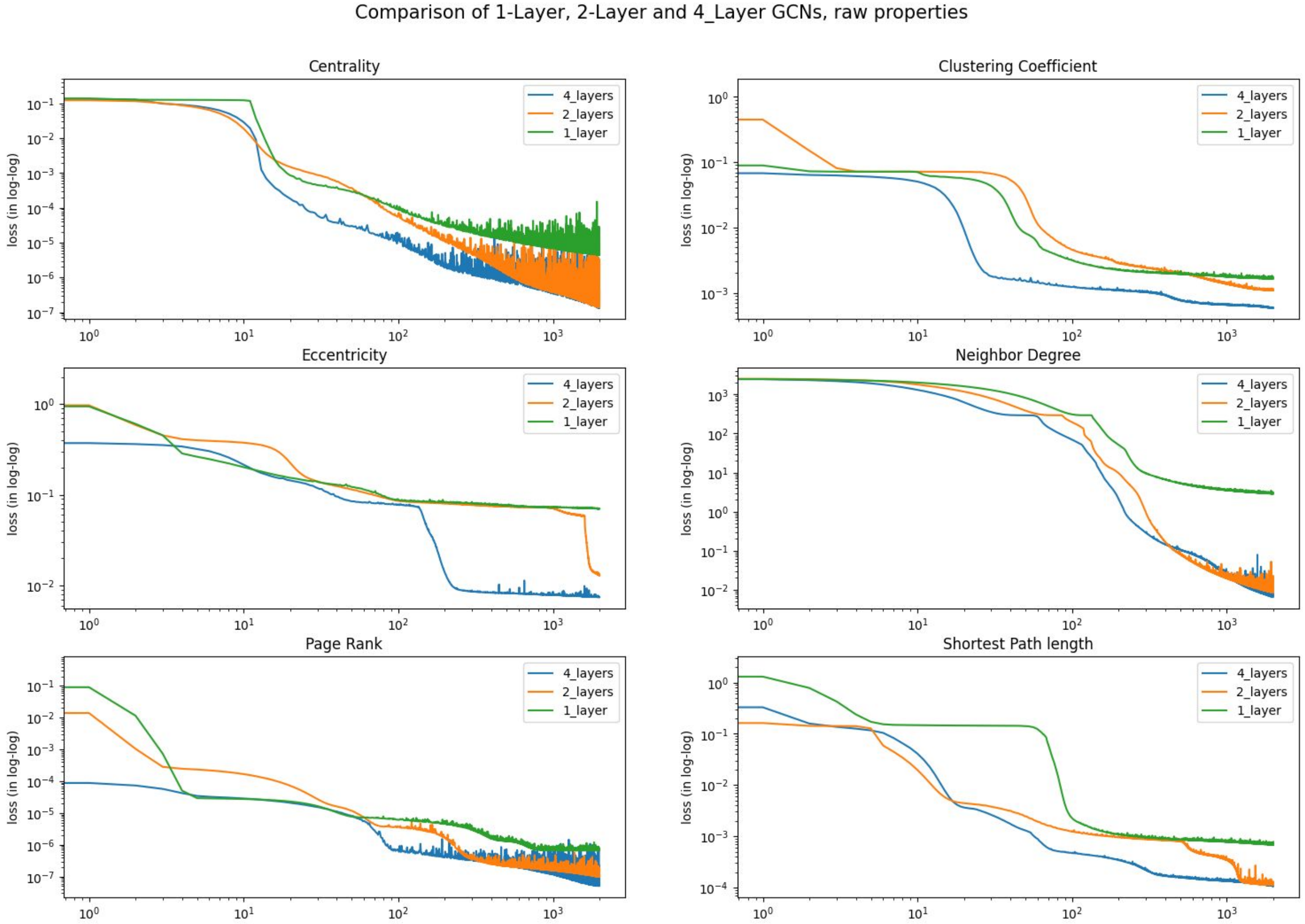
Properties	Time (s) (per 5000 graphs)	Time (Big-O)	Space (Big-O)
Degree Centrality	0.1	$O(n)$	$O(n)$
Avg. Neighbor Degree	1.5	$O(m + n)$	$O(n)$
Page Rank	12	$O(\log n)$	$O(m + n)$
Eccentricity	20	$O(n)$	$O(n)$
Avg. Shortest Path Length	27	$O(n.m + n^2)$	$O(n^2)$
Clustering Coefficient	40	$O(m + n)$	$O(n)$

All NN models had an inference time of **less than 2 Seconds (per 5000 graphs)**.

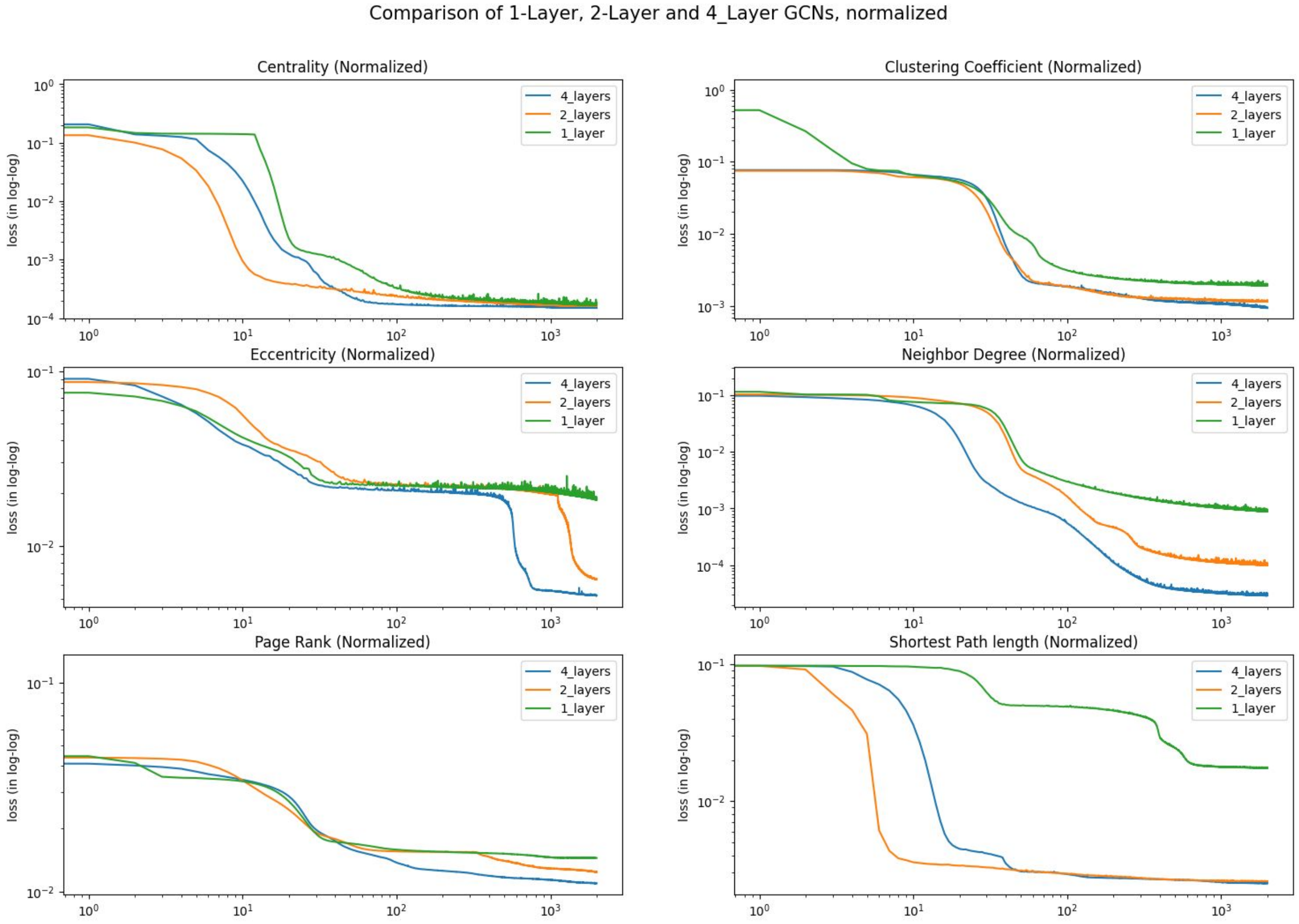
GCN Training and Validation comparison



GCN Results

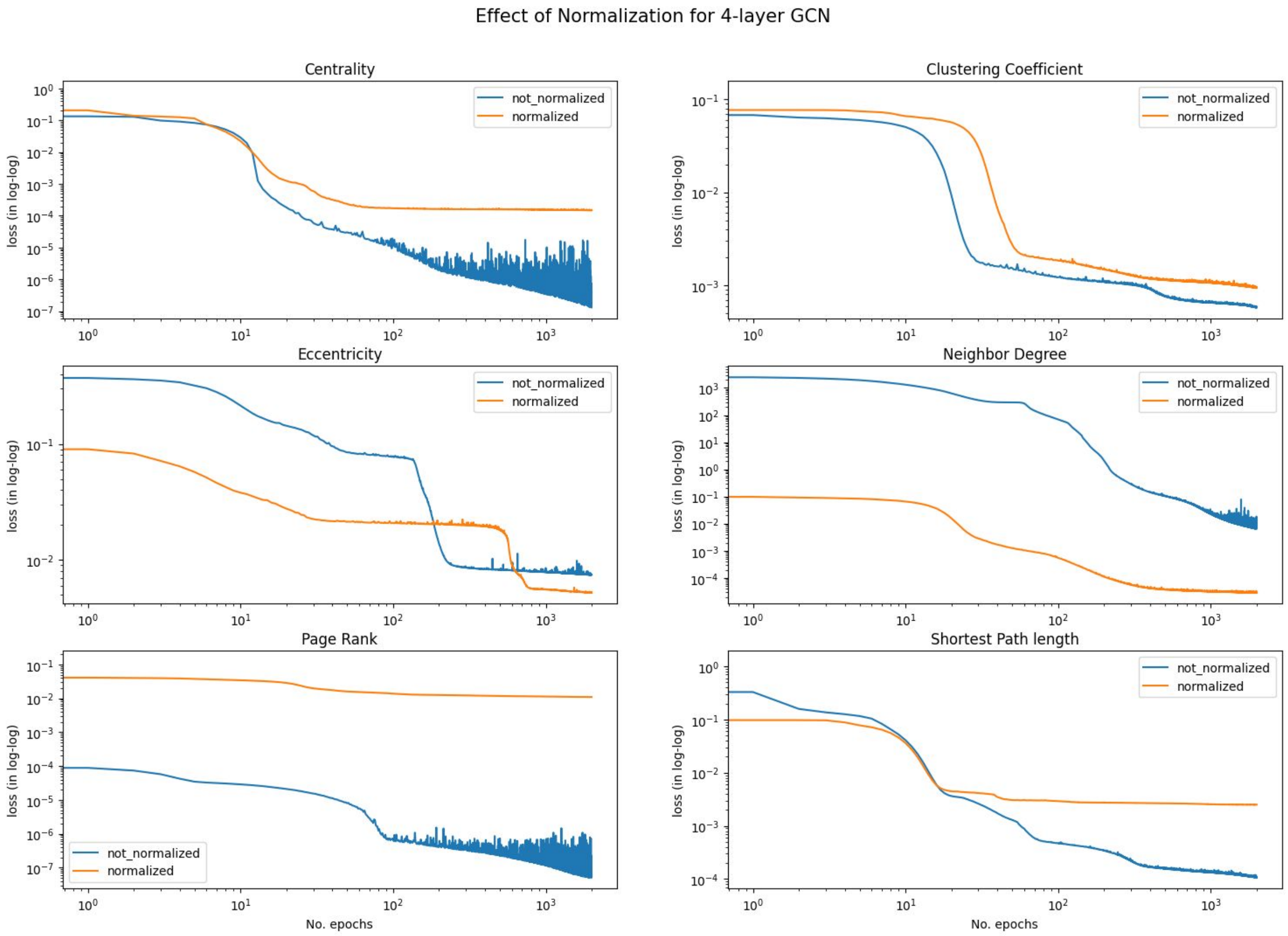

Comparison of 1-Layer, 2-Layer and 4_Layer GCNs, raw properties
Centrality
Clustering Coefficient
Eccentricity
Neighbor Degree
Page Rank
Shortest Path length
loss (in log-log)
4_layers
2_layers
1_layer

GCN Results

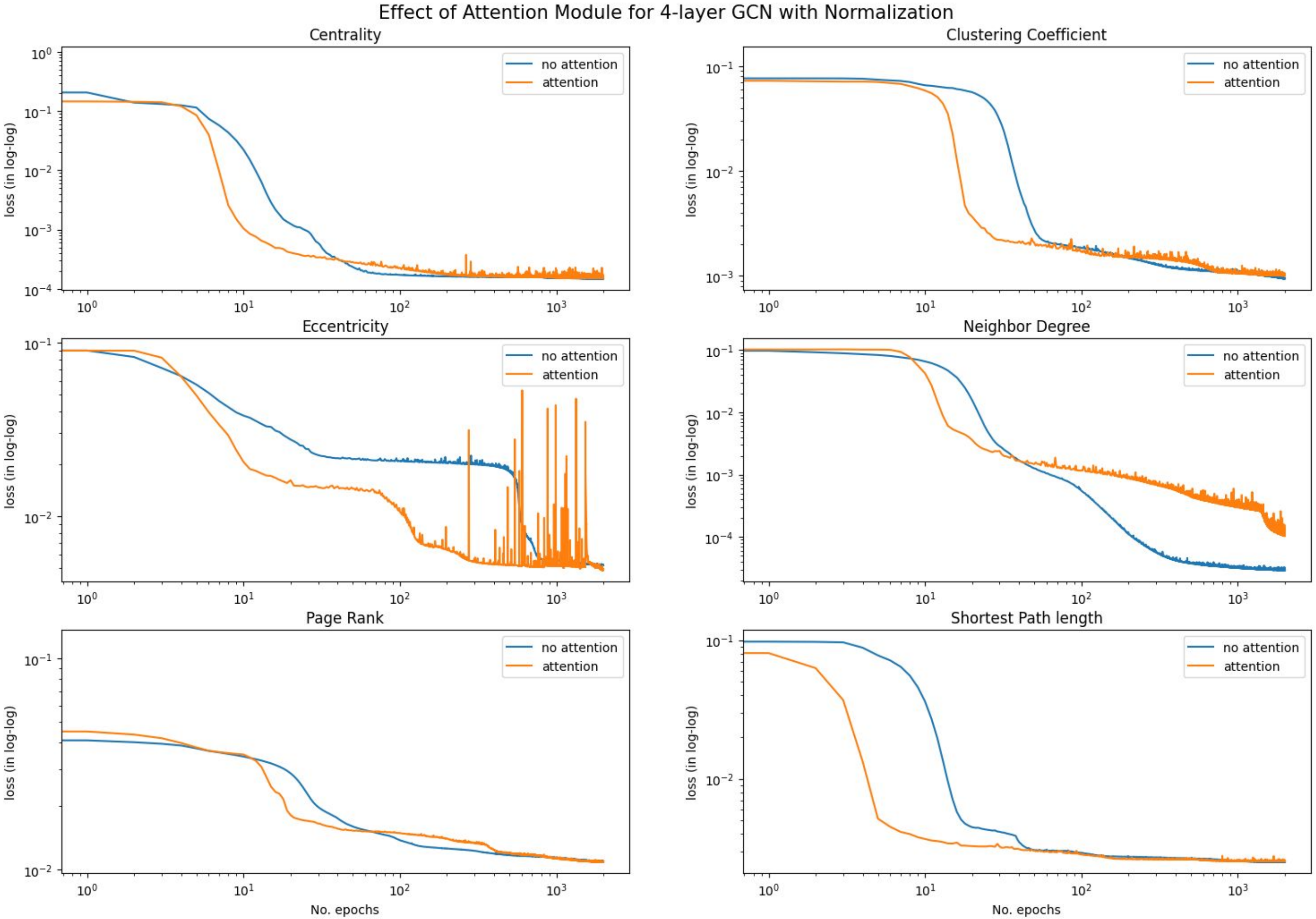


Comparison of 1-Layer, 2-Layer and 4_Layer GCNs, normalized
Centrality (Normalized)
Clustering Coefficient (Normalized)
Eccentricity (Normalized)
Neighbor Degree (Normalized)
Page Rank (Normalized)
Shortest Path length (Normalized)
loss (in log-log)
4_layers
2_layers
1_layer

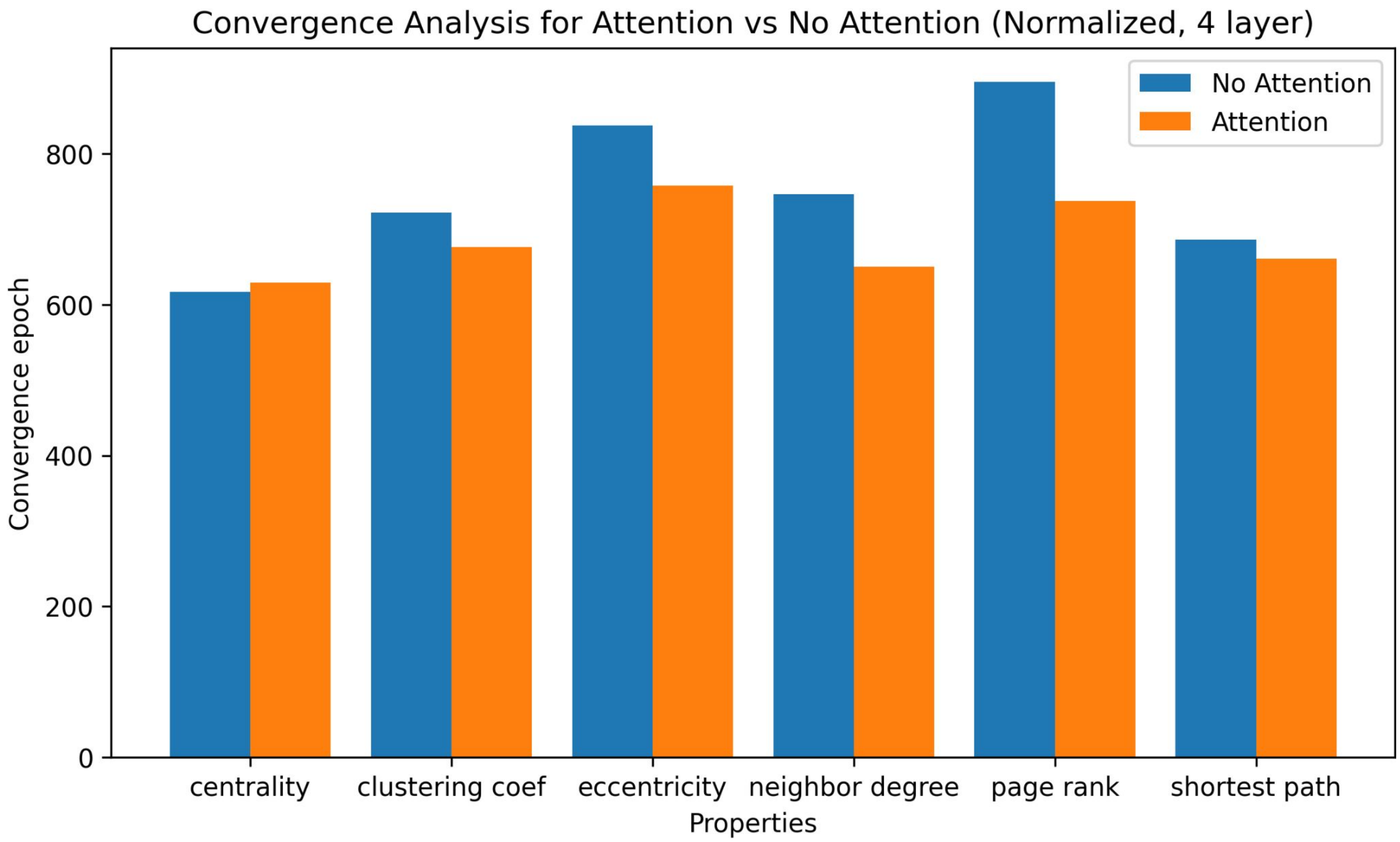
GCN Results


Effect of Normalization for 4-layer GCN
Centrality
Clustering Coefficient
Eccentricity
Neighbor Degree
Page Rank
Shortest Path length
not_normalized
normalized
loss (in log-log)
No. epochs

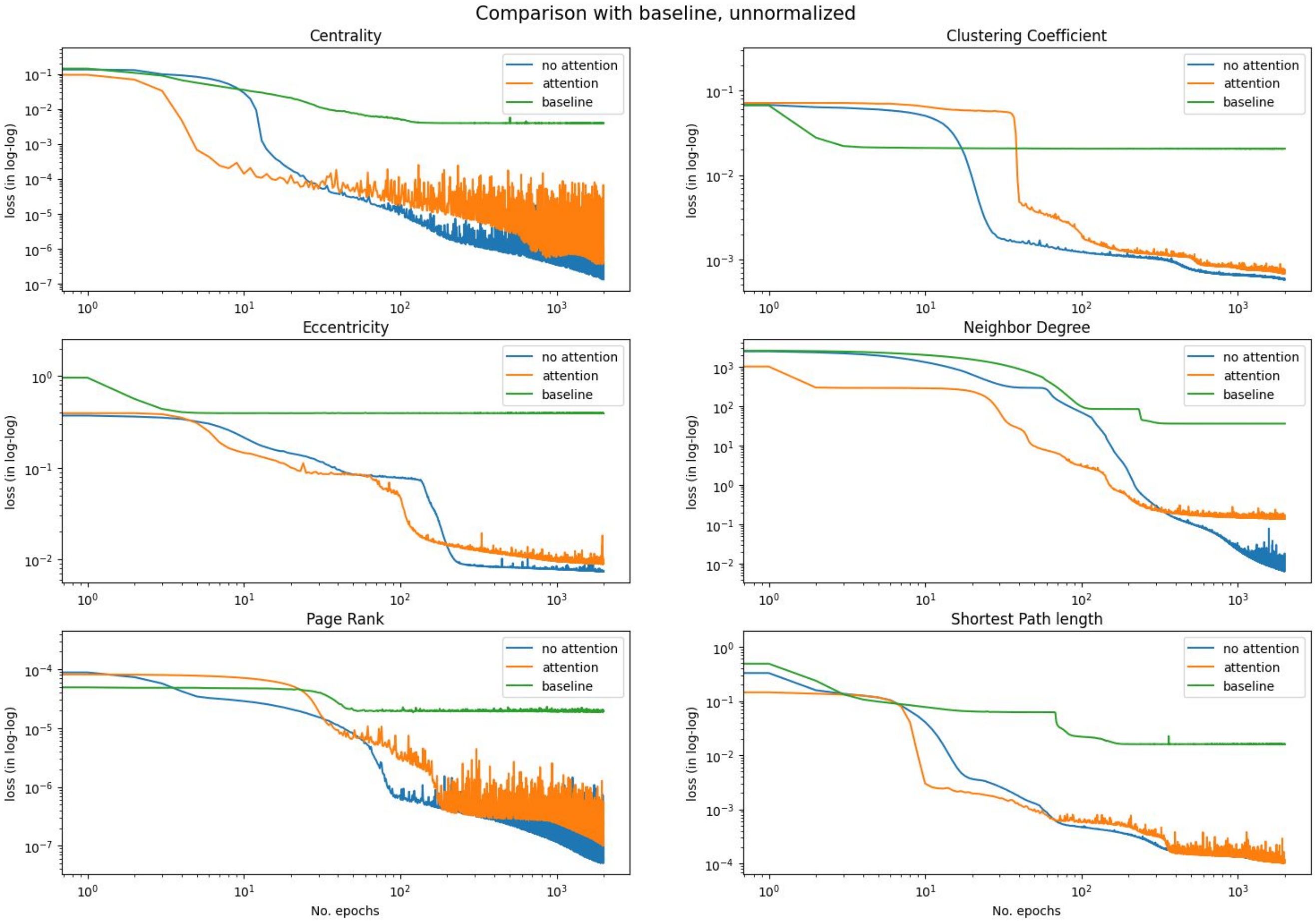
GCN + Attention Module Results



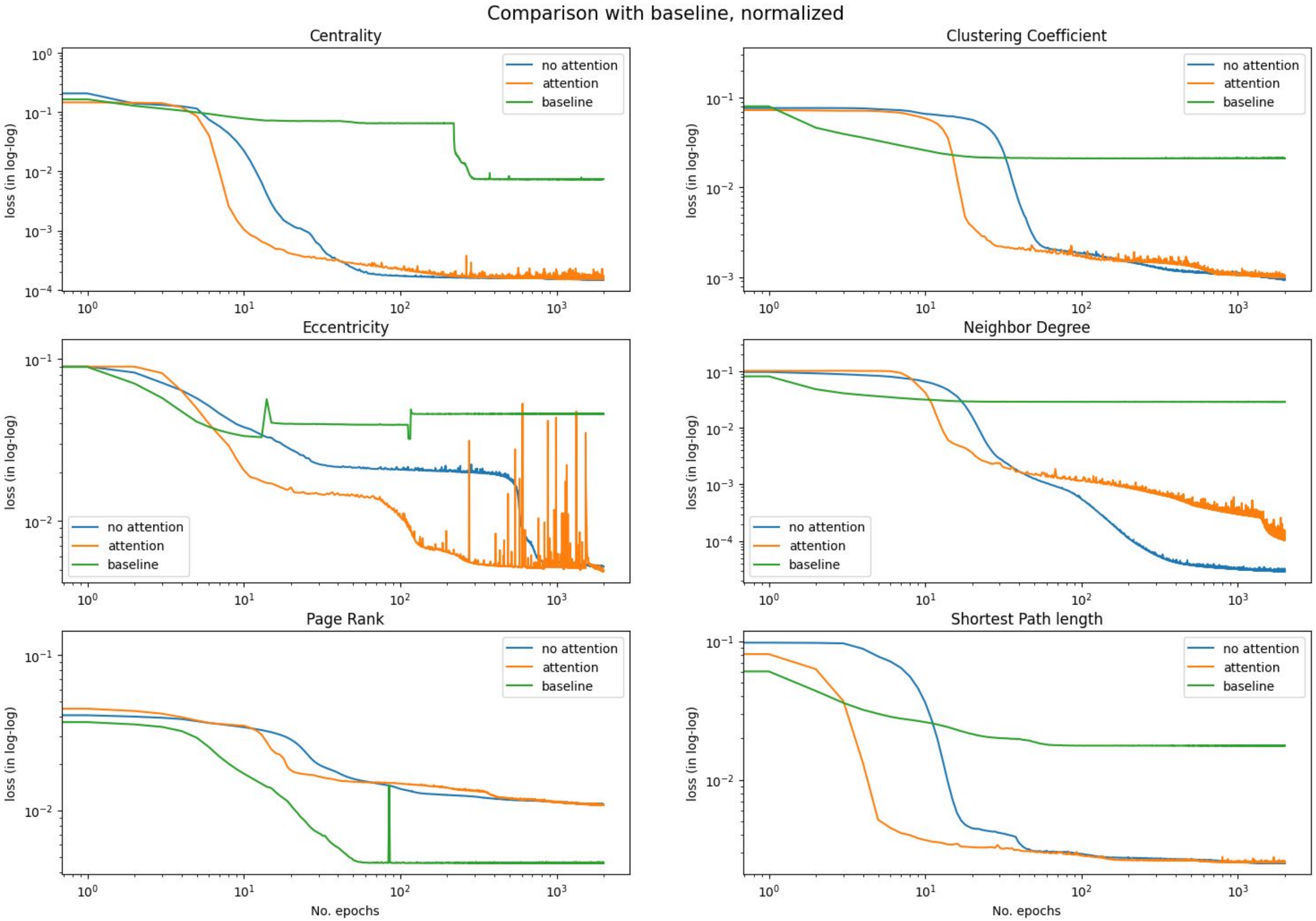
GCN + Attention Module Convergence



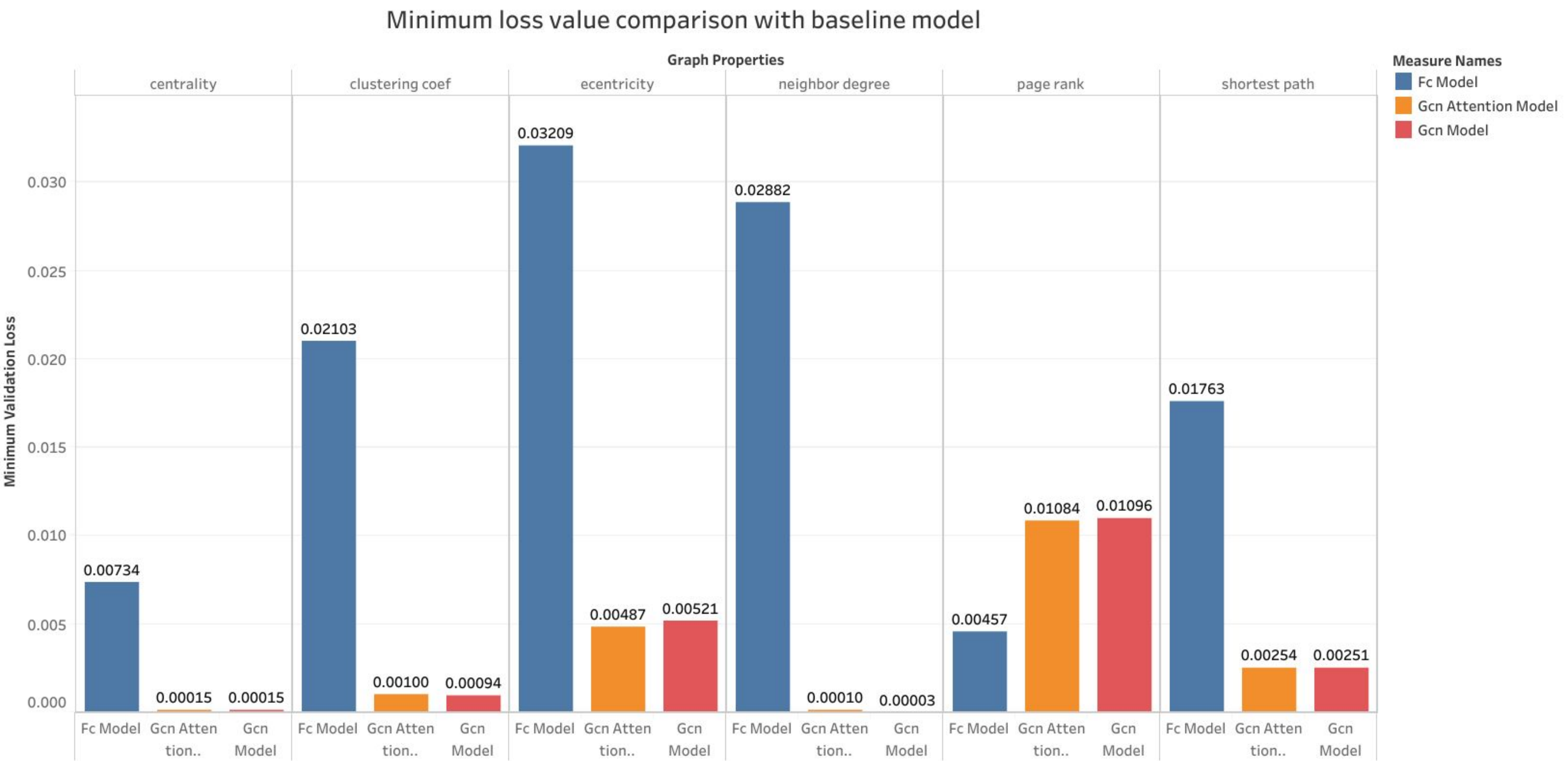
Validation loss comparison between models



Validation loss comparison between models



Summary of best epochs



Conclusion

- GCN provides significantly better accuracy over MLP
- Attention module provides faster convergence
- GCN is generally reliable for majority of unnormalized simple properties
- GCN is computationally feasible

Conclusion

- Extending capabilities of GCN and other neural network models
  - NP-complete/hard properties
  - Non-deterministic algorithmic attributes
  - Etc.
- Solid option for predicting raw graph data