Understanding the Representation Power of Graph Neural Networks in Learning Graph Topology

Nima Dehmamy, Albert-László Barabási, Rose Yu

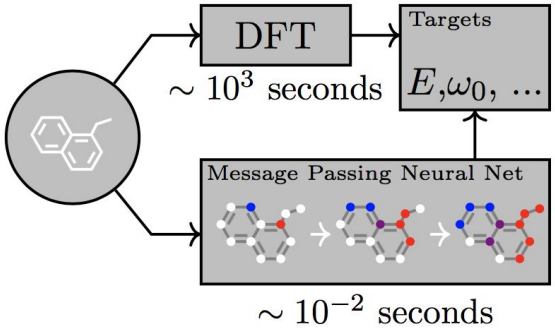
NeurIPS 2019

https://arxiv.org/pdf/1907.05008.pdf
https://youtu.be/kk_x0wOvZYQ

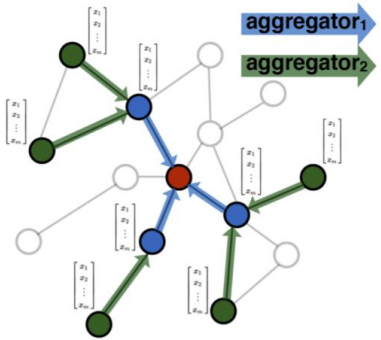
Success of Graph Neural Networks

- Web-Scale Recommendation
- Physical Dynamics
- Quantum Chemistry

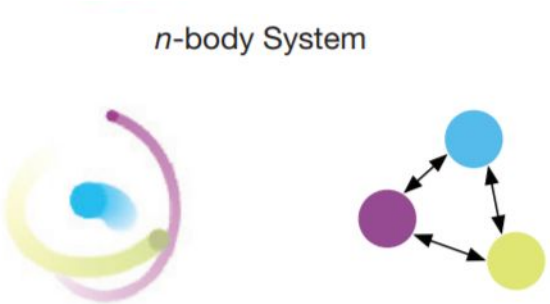
How well do Graph Neural Networks learn graph topology?



Gilmer et al. 2017



Hamilton et al. NIPS 2017



Battaglia et al. 2018

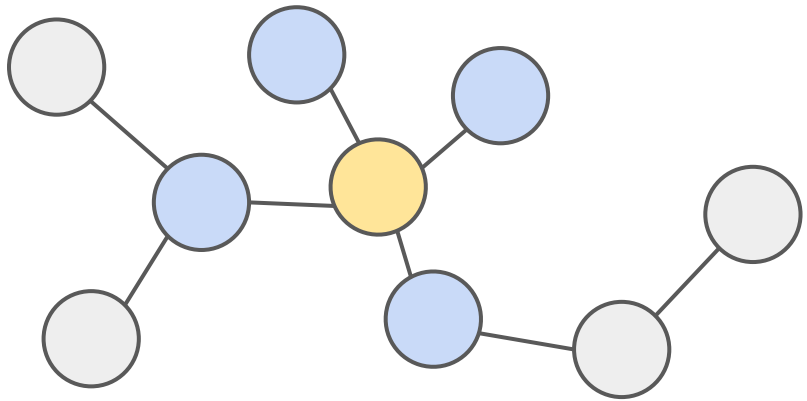
Learning Graph Moments

The adjacency matrix of a graph A encodes graph topology, where each element A_{ij} represents an edge from node i to node j.

Graph Moments: a pth order graph moment M_p is the ensemble average of an order p polynomial of A

$$M_p(A) = \prod_{q=1}^{p} (A \cdot W_q + B_q)$$

Graph moments are important because they encode higher order structural information such as the number of paths of length p



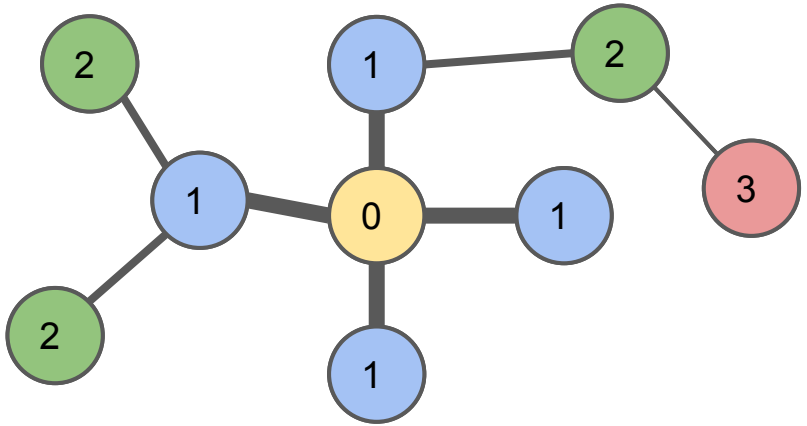
Assumption: Permutation Invariance

Requiring **Permutation Invariance** (i.e. invariance under relabeling nodes) the simplest graph moments become powers of A, counting number of path of length p.

$$M_p = \sum_j (A^p)_{ij}$$

GCN: $$F(A, h) = \sigma\left(f(A) \cdot h \cdot W + b\right)$$

- **Graph convolutional networks** (GCNs) are permutation invariant
- Weights and biases need to be **shared** among all nodes, similar to ConvNet.
- Restrict the representation power of GCNs

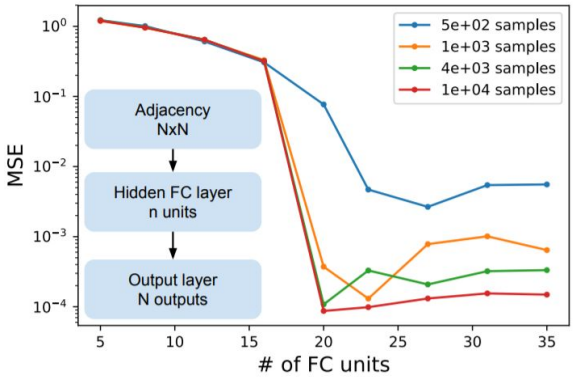


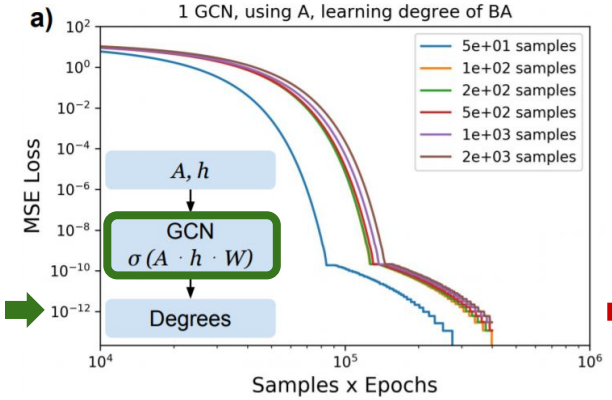
Learning Graph Moments, 20 nodes, FC vs GCN

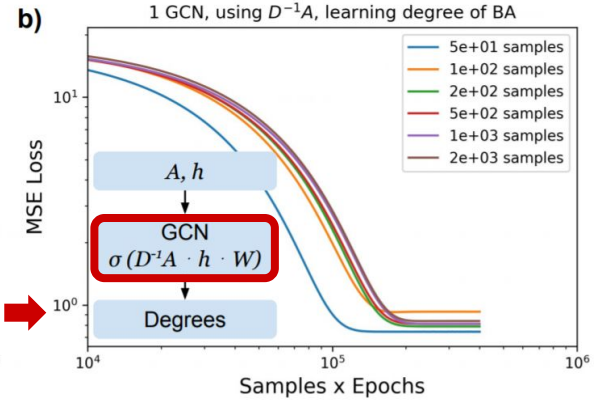
- Fully Connected (FC) networks can approximate any function, but require large amounts of data.

- GCN can perform very well with small samples and $O(1)$ parameters.
- However, choosing the correct **Propagation Rule (PR)** is crucial for GCN's performance.

⇒ How restrictive are GCN and how do we make them more expressive?







Learning Graph Moments, FC vs GCN

Theorem 1. *A fully connected neural network with one hidden layer requires* $n > O(C_f^2) \sim O(p^2 N^{2q})$ *number of neurons in the best case with* $1 \leq q \leq 2$ *to learn a graph moment of order* p *for graphs with* N *nodes. Additionally, it also needs* $S > O(nd) \sim O\left(p^2 N^{2q+2}\right)$ *number of samples to make the learning tractable.*

Theorem 2. *With the number of layers* n *greater or equal to the order* p *of a graph moment* $M_p(A)$*, graph convolutional networks with residual connections can learn a graph moment* M_p *with* $O(p)$ *number of neurons, independent of the size of the graph.*

Should say the take-away from the theorem rather than putting the statements directly

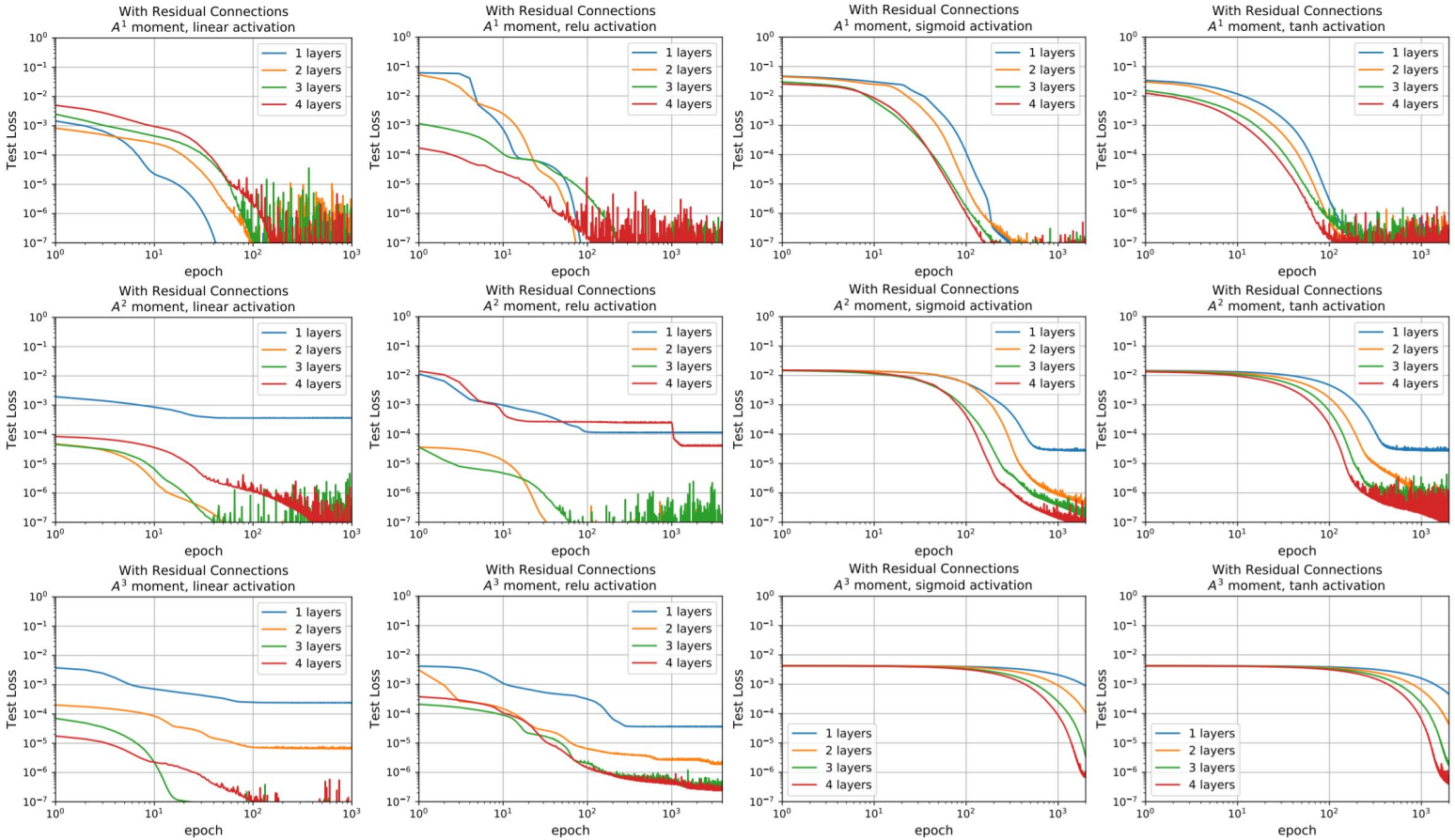
Learning Graph Moments, FC vs GCN

$$\text{GCN:} \quad F(A,h) = \sigma\left(f(A)\cdot h\cdot W + b\right)$$

To learn a graph moment of order p :

1) A fully connected layer, or a general graph neural network learning $f(A)$ requires $\sim O(p^2N^{2q})$ parameters ($1\leq q \leq 2$, N=# graph nodes) and a large amount of training data ($\sim O(p^2N^{2q+2})$) (from Barron 1994)
2) A GCN with $n \geq p$ layers can learn it with $O(p)$ parameters, independent of graph size, if the appropriate $f(A)$ is given and fixed.

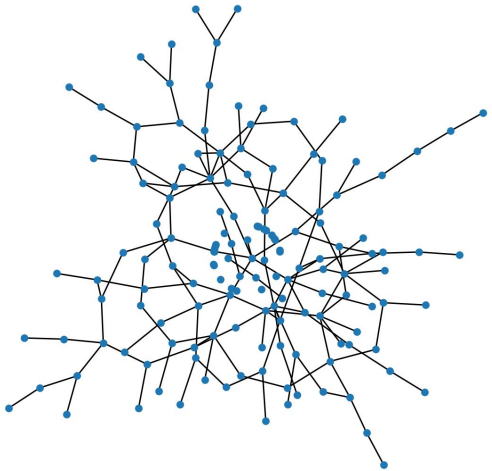
Learning Graph Moments with GCN



Importance of multiple Propagation Rules (PR)

- Multiple PR can significantly improve expressivity

Erdős–Rényi (ER, random graph)



Barabasi-Albert (BA, rich gets richer)

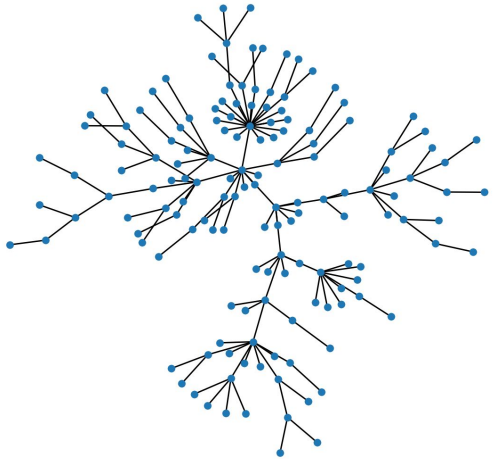


Table 1: Test accuracy with different modules combinations for BA-ER. $f_1 = A$, $f_2 = D^{-1}A$, and $f_3 = D^{-1/2}AD^{-1/2}$.

Modules	Accuracy
f_1	53.5 %
f_3	76.9 %
f_1, f_3	89.4 %
f_1, f_2, f_3	98.8 %

Suggested Modular Design to Improve Expressivity

1) Use Multiple PR and bias
2) Use a per-node FC layers after PR to mix outputs
3) Use skip layers (like Jumping Knowledge Networks)

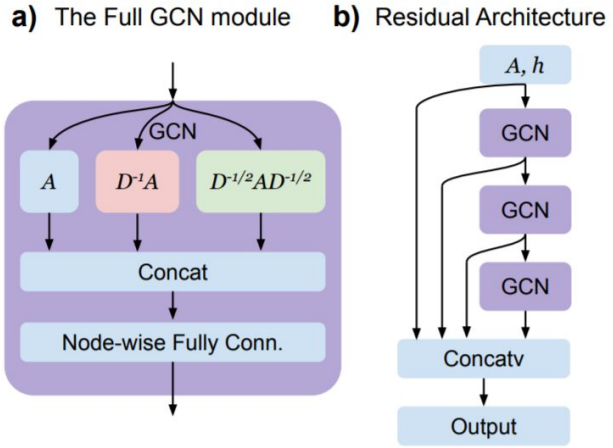
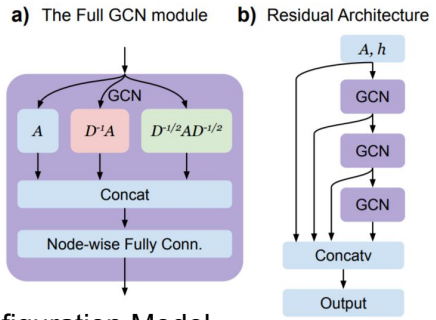


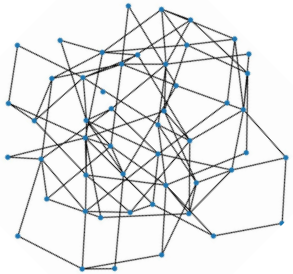
Figure 3: GCN layer (a), using three different propagation rules and a node-wise FC layer. Using residual connections (b) allows a n-layer modular GCN to learn any polynomial function of order n of its constituent operators.

Tests on Graph Classification Problems

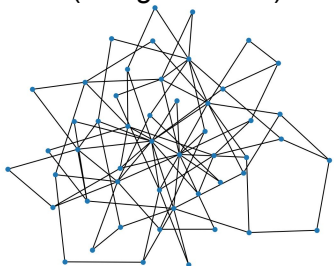
- Examine the effect of width vs depth
- 1) BA vs ER; 2) BA vs shuffled links BA (aka Configuration model)



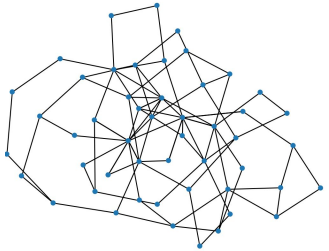
ER (Random Graph)



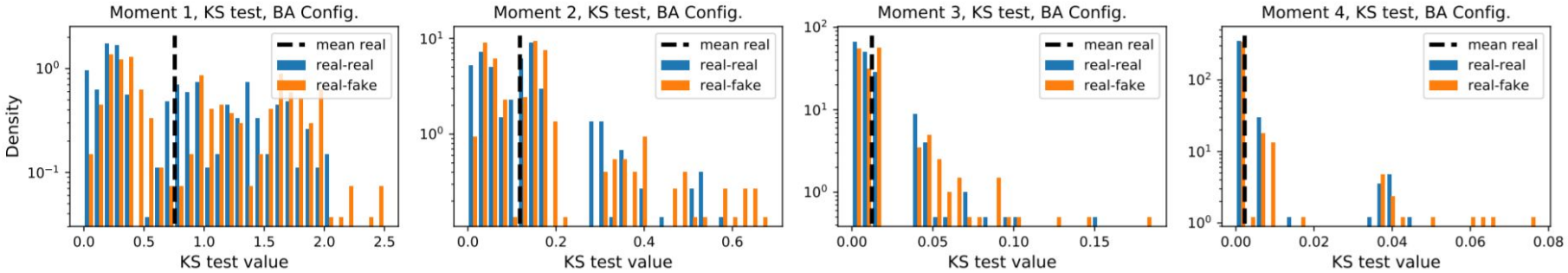
BA (rich gets richer)



BA Configuration Model



- 2) BA vs Config BA is hard; Degree distributions and higher moment hard to distinguish



Tests on Graph Classification Problems

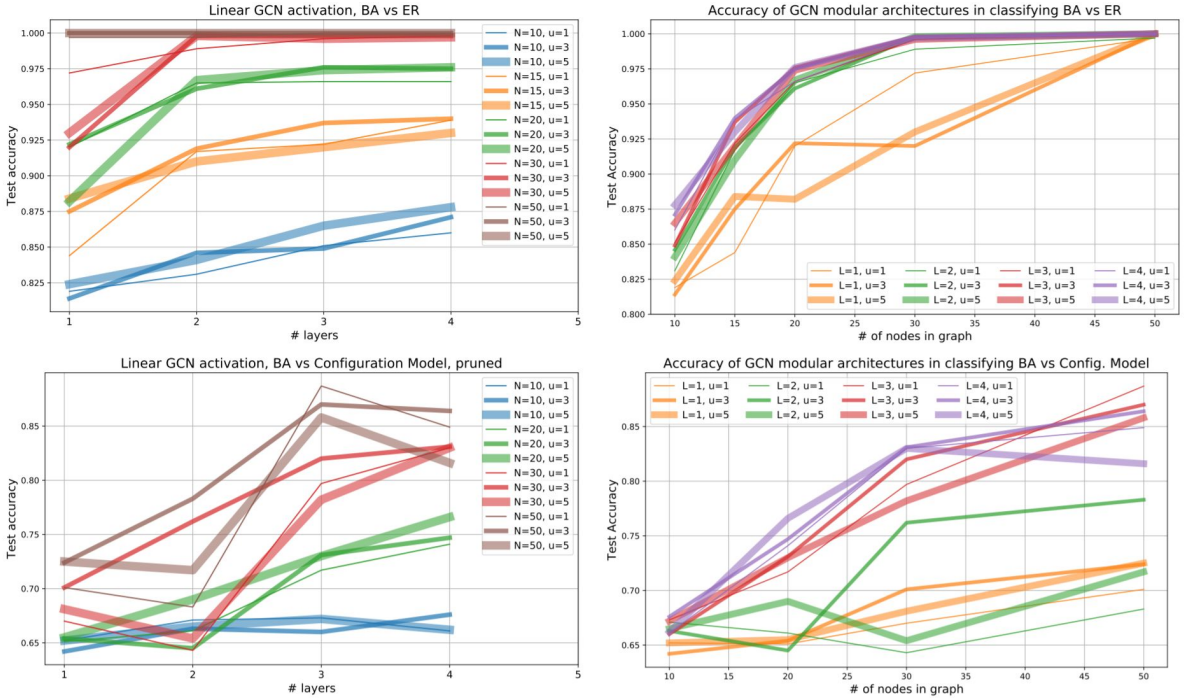
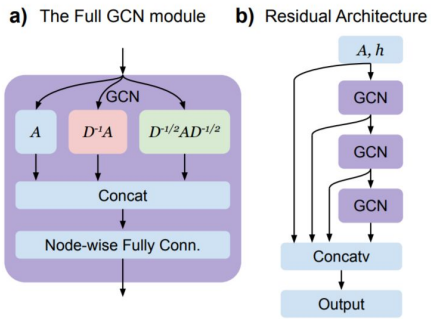


Figure 6: Classify graphs of **Barabasi-Albert** model vs. **Erdos-Renyi** model (top) and **Barabasi-Albert** model vs. **configuration** model (bottom). Left: test accuracy with respect to network depth for different number of nodes (N) and number of units (U). Right: test accuracy with respect to graph size for different number of layers (L) and number of units (U).



Number of layers is more influential than layer width for learning graph topology

Conclusions

- GCN, though being very restrictive, are ideal for learning graph moments
- GCN can be made much more expressive by combining propagation rules
- GCN can learn graph moments and solve hard graph structure classification problems using very few parameters
- Depth is much more important than width in GCN