A Metadata-Driven Approach to Understand Graph Neural Networks

Ting Wei Li[1], Qiaozhu Mei[1], Jiaqi Ma[2]
[1] University of Michigan
[2] University of Illinois Urbana-Champaign

NEURAL INFORMATION PROCESSING SYSTEMS

Motivation

1. Data Properties affect GNN performance.

-> Ex: Homophily assumption.

2. Graph learning benchmarks become more accessible.

-> Open Graph Benchmark, Graph Learning Indexer, …

Question: Can we infer critical data properties from GNN performance on benchmark datasets?

Main Contribution

- We introduce a novel **metadata-driven approach to identify critical graph data properties** affecting GNN performance.

- We demonstrate its effectiveness through a case study on a specific salient data property identified by our approach (Gini. of degree distribution, or, Gini-Degree).

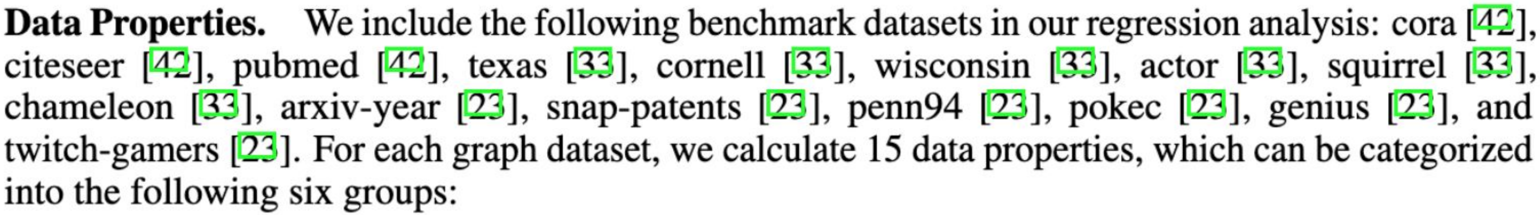
- We develop an in-depth understanding of **how the degree distribution of graph data influences GNN performance** through both a novel theoretical analysis and a carefully controlled experiment.

Regression Analysis: Identifying Salient Factors

Given two observation matrices (Metadata matrices from benchmark experiments),

1) Model Performance Matrix: each entry means the performance (accuracy) of a model when applying on a specific dataset
2) Graph Feature Matrix: each entry means a particular property of the graph dataset (such as clustering coefficient, homophily ratio, …)

Datasets Considered



Data Properties. We include the following benchmark datasets in our regression analysis: cora [42], citeseer [42], pubmed [42], texas [33], cornell [33], wisconsin [33], actor [33], squirrel [33], chameleon [33], arxiv-year [23], snap-patents [23], penn94 [23], pokec [23], genius [23], and twitch-gamers [23]. For each graph dataset, we calculate 15 data properties, which can be categorized into the following six groups:

Graph Data Properties Considered

- Basic: Edge Density, Average Degree, Degree Assortativity;
- Distance: Pseudo Diameter;
- Connectivity: Relative Size of Largest Connected Component (RSLCC);
- Clustering: Average Clustering Coefficient (ACC), Transitivity, Degeneracy;
- Degree Distribution: Gini Coefficient of Degree Distribution (Gini-Degree);
- Attribute: Edge Homogeneity, In-Feature Similarity, Out-Feature Similarity, Feature Angular

SNR, Homophily Measure, Attribute Assortativity

Regression Analysis: Formulation

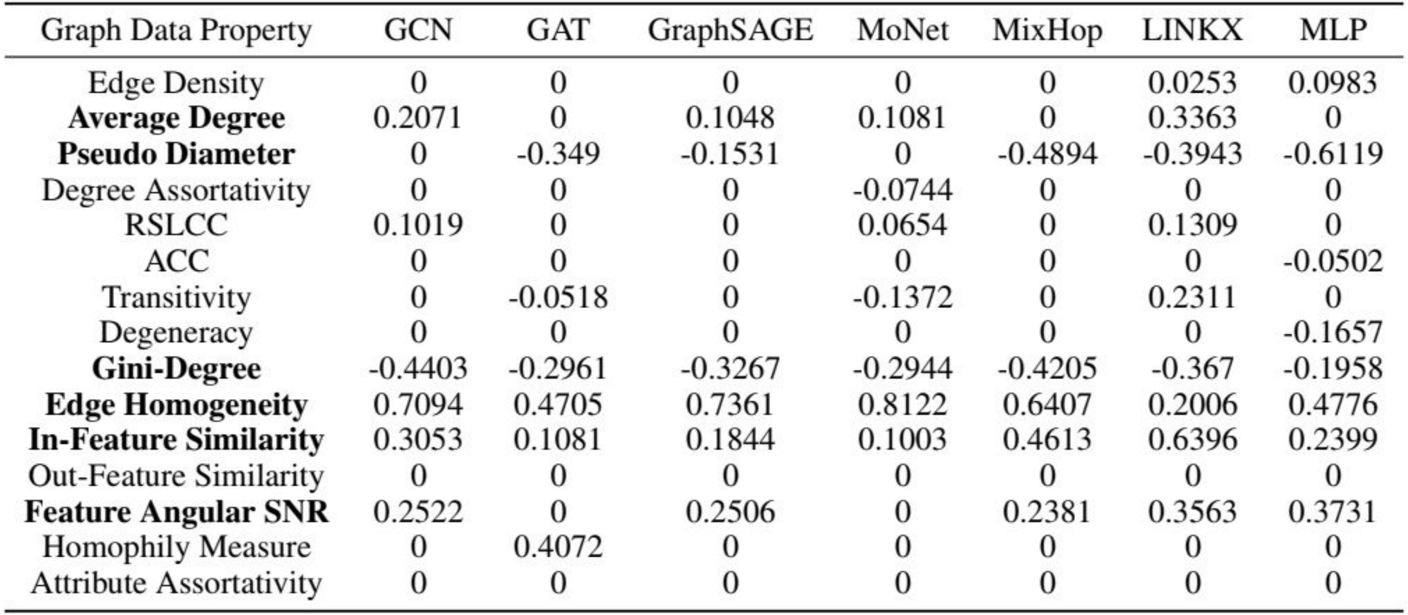
We define a **Multivariate Sparse Group Lasso Problem** to solve the dependency between model performance and dataset properties. [Here **Y is the model performance matrix**, **X is the graph feature matrix** and B is the coefficient matrix]

$$\underset{\mathbf{B}}{\operatorname{argmin}} \frac{1}{2n} ||\mathbf{Y} - \mathbf{XB}||_2^2 + \lambda_1 ||\mathbf{B}||_1 + \lambda_g ||\mathbf{B}||_{2,1}$$

Regression Analysis: Formulation

- The L1 penalty encourages the coefficient matrix B to be sparse, only selecting salient data properties.
- The L2,1 penalty further leverages the structure of the dependent variables and tries to select relevant data properties at a group level. If a property is selected, then most of the coefficients will be non-zero, which is better for us to discover “widely influential factors”.

Regression Analysis: Result



Graph Data Property	GCN	GAT	GraphSAGE	MoNet	MixHop	LINKX	MLP
Edge Density	0	0	0	0	0	0.0253	0.0983
Average Degree	0.2071	0	0.1048	0.1081	0	0.3363	0
Pseudo Diameter	0	-0.349	-0.1531	0	-0.4894	-0.3943	-0.6119
Degree Assortativity	0	0	0	-0.0744	0	0	0
RSLCC	0.1019	0	0	0.0654	0	0.1309	0
ACC	0	0	0	0	0	0	-0.0502
Transitivity	0	-0.0518	0	-0.1372	0	0.2311	0
Degeneracy	0	0	0	0	0	0	-0.1657
Gini-Degree	-0.4403	-0.2961	-0.3267	-0.2944	-0.4205	-0.367	-0.1958
Edge Homogeneity	0.7094	0.4705	0.7361	0.8122	0.6407	0.2006	0.4776
In-Feature Similarity	0.3053	0.1081	0.1844	0.1003	0.4613	0.6396	0.2399
Out-Feature Similarity	0	0	0	0	0	0	0
Feature Angular SNR	0.2522	0	0.2506	0	0.2381	0.3563	0.3731
Homophily Measure	0	0.4072	0	0	0	0	0
Attribute Assortativity	0	0	0	0	0	0	0

Regression Analysis: Result

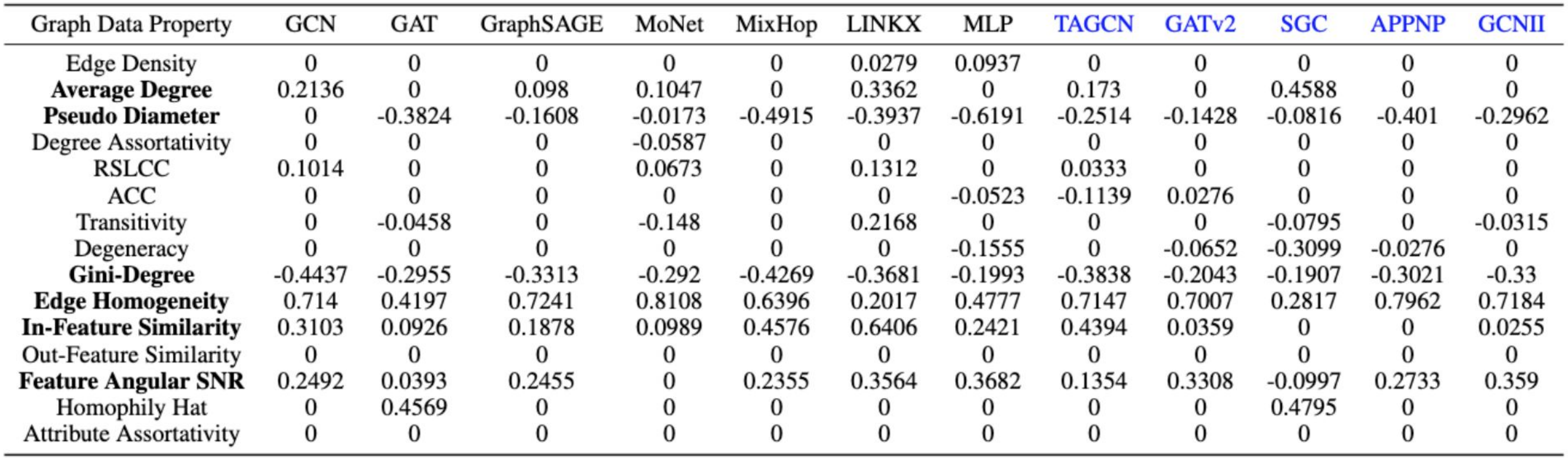
We focused on: **Gini-Degree** (Gini coefficient of the degree distribution)

Gini Coefficient of Degree Distribution measures how the degree distribution deviates from a uniform distribution.

Gini-Degree has a **negative correlation** with GNN performance!

We will focus on the property for a in-depth **theoretical analysis** and a **controlled experiment**.

(Conti.) With additional GNN models

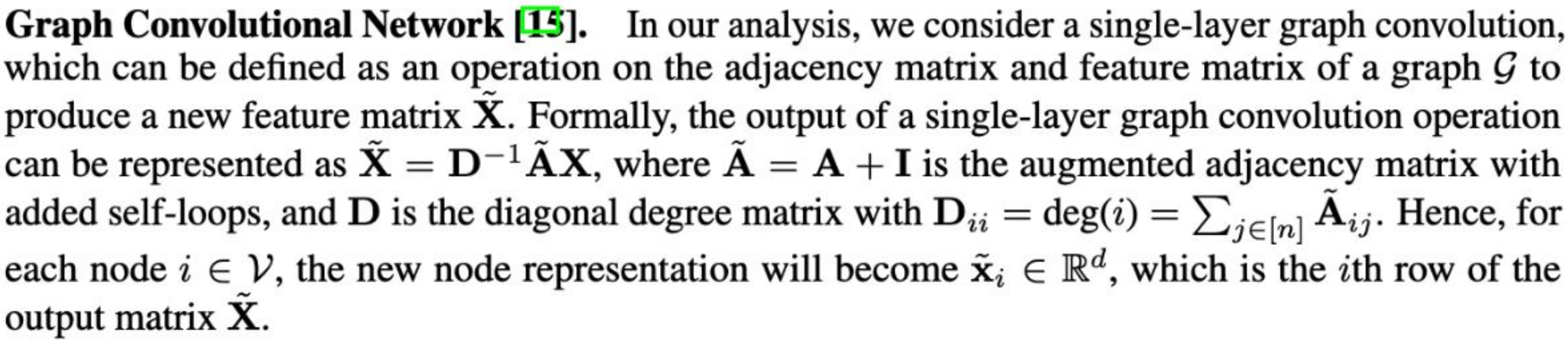


Graph Data Property	GCN	GAT	GraphSAGE	MoNet	MixHop	LINKX	MLP	TAGCN	GATv2	SGC	APPNP	GCNII
Edge Density	0	0	0	0	0	0.0279	0.0937	0	0	0	0	0
Average Degree	0.2136	0	0.098	0.1047	0	0.3362	0	0.173	0	0.4588	0	0
Pseudo Diameter	0	-0.3824	-0.1608	-0.0173	-0.4915	-0.3937	-0.6191	-0.2514	-0.1428	-0.0816	-0.401	-0.2962
Degree Assortativity	0	0	0	-0.0587	0	0	0	0	0	0	0	0
RSLCC	0.1014	0	0	0.0673	0	0.1312	0	0.0333	0	0	0	0
ACC	0	0	0	0	0	0	-0.0523	-0.1139	0.0276	0	0	0
Transitivity	0	-0.0458	0	-0.148	0	0.2168	0	0	0	-0.0795	0	-0.0315
Degeneracy	0	0	0	0	0	0	-0.1555	0	-0.0652	-0.3099	-0.0276	0
Gini-Degree	-0.4437	-0.2955	-0.3313	-0.292	-0.4269	-0.3681	-0.1993	-0.3838	-0.2043	-0.1907	-0.3021	-0.33
Edge Homogeneity	0.714	0.4197	0.7241	0.8108	0.6396	0.2017	0.4777	0.7147	0.7007	0.2817	0.7962	0.7184
In-Feature Similarity	0.3103	0.0926	0.1878	0.0989	0.4576	0.6406	0.2421	0.4394	0.0359	0	0	0.0255
Out-Feature Similarity	0	0	0	0	0	0	0	0	0	0	0	0
Feature Angular SNR	0.2492	0.0393	0.2455	0	0.2355	0.3564	0.3682	0.1354	0.3308	-0.0997	0.2733	0.359
Homophily Hat	0	0.4569	0	0	0	0	0	0	0	0.4795	0	0
Attribute Assortativity	0	0	0	0	0	0	0	0	0	0	0	0

Sketch of the Theoretical Analysis

- Our analysis investigates the linear separability of node representations produced by applying graph convolution to the node features.
- In the case that the graph data comes from a **Degree-Corrected Contextual Stochastic Block Model (DC-CSBM)** with 2 classes, we show that nodes from different classes are more separable when their degree exceeds a threshold.
- This separability result relates the **graph data's degree distribution** to the **GNN model performance**.

Graph Convolution Operation



Graph Convolutional Network [15]. In our analysis, we consider a single-layer graph convolution, which can be defined as an operation on the adjacency matrix and feature matrix of a graph $\mathcal{G}$ to produce a new feature matrix $\tilde{\mathbf{X}}$. Formally, the output of a single-layer graph convolution operation can be represented as $\tilde{\mathbf{X}} = \mathbf{D}^{-1}\tilde{\mathbf{A}}\mathbf{X}$, where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ is the augmented adjacency matrix with added self-loops, and $\mathbf{D}$ is the diagonal degree matrix with $\mathbf{D}_{ii} = \deg(i) = \sum_{j \in [n]} \tilde{\mathbf{A}}_{ij}$. Hence, for each node $i \in \mathcal{V}$, the new node representation will become $\tilde{\mathbf{x}}_i \in \mathbb{R}^d$, which is the ith row of the output matrix $\tilde{\mathbf{X}}$.

DC-CSBM with 2 classes

- **Class assignment**: each node has an i.i.d. Bernoulli random variable (p=0.5) determining its class label.
- **Edge probability**: intra-class edge probability is p and inter-class edge probability is q. (Normally p is larger than q)
- **Degree-correction parameter**: $\theta i \in (0, n]$, which is the propensity of node i to connect with other nodes. We have a constraint that the sum of all θi equals to n.

[Normal SBM: all $\theta i = 1$][We let the average degree be fixed.]

DC-CSBM with 2 classes

Assumptions on adjacency matrix and feature matrix:

- Edge generation: conditioning on the class assignment, if i, j are in the same class, then **Aij ~ Poisson(θi*θj*p)**; if i, j are in different classes, then **Aij~ Poisson(θi*θj*q)**.
- Node feature generation: nodes in class 0 will have their feature from independent d-dimensional Gaussian random vector with mean μ0; nodes in class 1 will have their feature from independent d-dimensional Gaussian random vector with mean μ1.

So for a particular choice of n, μ0, μ1, p, q and θ, we can define a class of random graphs generated by these parameters and we can sample a random graph from this model.

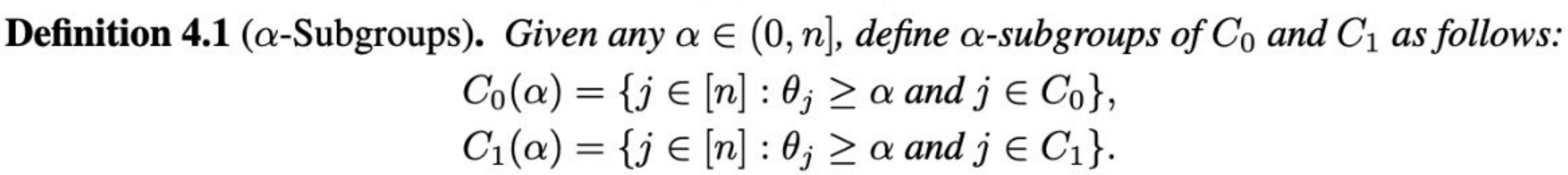
Linear Separability

Linear separability refers to the ability to linearly differentiate nodes in the two classes based on their feature vectors. (i.e. finding a hyperplane to separate two groups of nodes)

Linear separability is closely related to GNN performance. Intuitively, more nodes being linearly separable will lead to better GNN performance.

Degree-Thresholded Subgroups

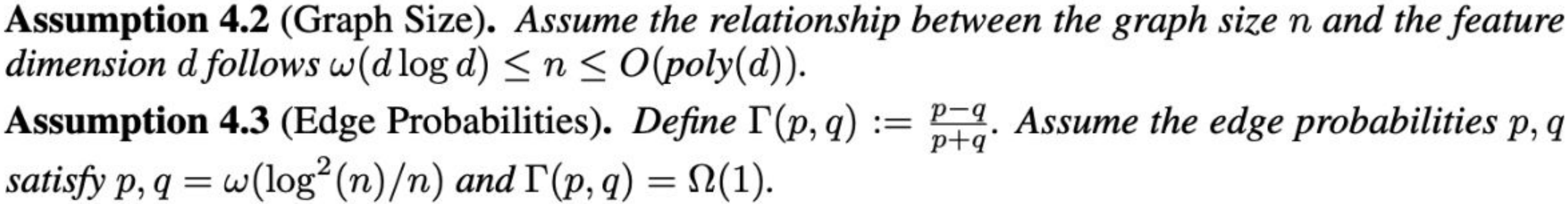
To better control the behavior of graph convolution operation, we will focus on particular subgroups of C0 and C1 where the member nodes having their degree-corrected factor (θ) larger or equal to a pre-defined threshold α > 0.



Definition 4.1 (α-Subgroups). *Given any $\alpha \in (0, n]$, define α-subgroups of C_0 and C_1 as follows:*

$$C_0(\alpha) = \{j \in [n] : \theta_j \geq \alpha \textit{ and } j \in C_0\},$$
$$C_1(\alpha) = \{j \in [n] : \theta_j \geq \alpha \textit{ and } j \in C_1\}.$$

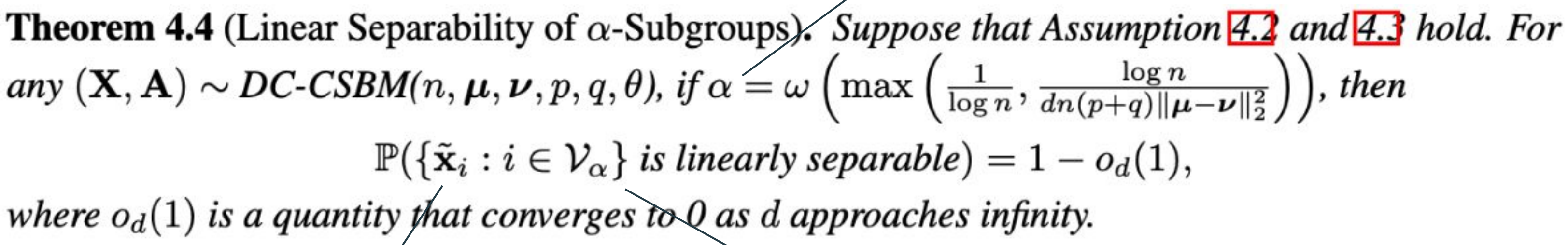
Main Theorem



Assumption 4.2 (Graph Size). *Assume the relationship between the graph size n and the feature dimension d follows $\omega(d \log d) \leq n \leq O(poly(d))$.*

Assumption 4.3 (Edge Probabilities). *Define $\Gamma(p, q) := \frac{p-q}{p+q}$. Assume the edge probabilities p, q satisfy $p, q = \omega(\log^2(n)/n)$ and $\Gamma(p, q) = \Omega(1)$.*

Main Theorem



Theorem 4.4 (Linear Separability of α-Subgroups)**.** *Suppose that Assumption 4.2 and 4.3 hold. For any* $(\mathbf{X}, \mathbf{A}) \sim$ *DC-CSBM*$(n, \boldsymbol{\mu}, \boldsymbol{\nu}, p, q, \theta)$*, if* $\alpha = \omega\left(\max\left(\frac{1}{\log n}, \frac{\log n}{dn(p+q)\|\boldsymbol{\mu}-\boldsymbol{\nu}\|_2^2}\right)\right)$*, then*

$$\mathbb{P}(\{\tilde{\mathbf{x}}_i : i \in \mathcal{V}_\alpha\} \textit{ is linearly separable}) = 1 - o_d(1),$$

where $o_d(1)$ *is a quantity that converges to 0 as d approaches infinity.*

the degree threshold!

convoluted node feature

the group of nodes whose degree exceeds the threshold alpha

Theoretical Analysis: Implications

The theorem implies the negative correlation between Gini-Degree and GNN performance

-> higher Gini-Degree implies more high-degree nodes in the network

-> result in more nodes receiving lower degrees

-> fewer nodes having degrees exceeding a certain threshold

-> smaller size of alpha subgroup

-> fewer nodes that can be linear separated

-> worse GNN model performance

Controlled Experiment: Setup

- Use **GraphWorld**[1] to generate synthetic datasets and adjust the parameter: **power-law coefficient of the degree distribution** to control the “Gini coefficient of the degree distribution”
- validate the efficacy of our regression analysis and open the door of further study on other graph properties (that potentially affect GNN’s performance)

[1] Graphworld: Fake graphs bring real insights for gnns, J Palowitch et al. 2022

Controlled Experiment: Result

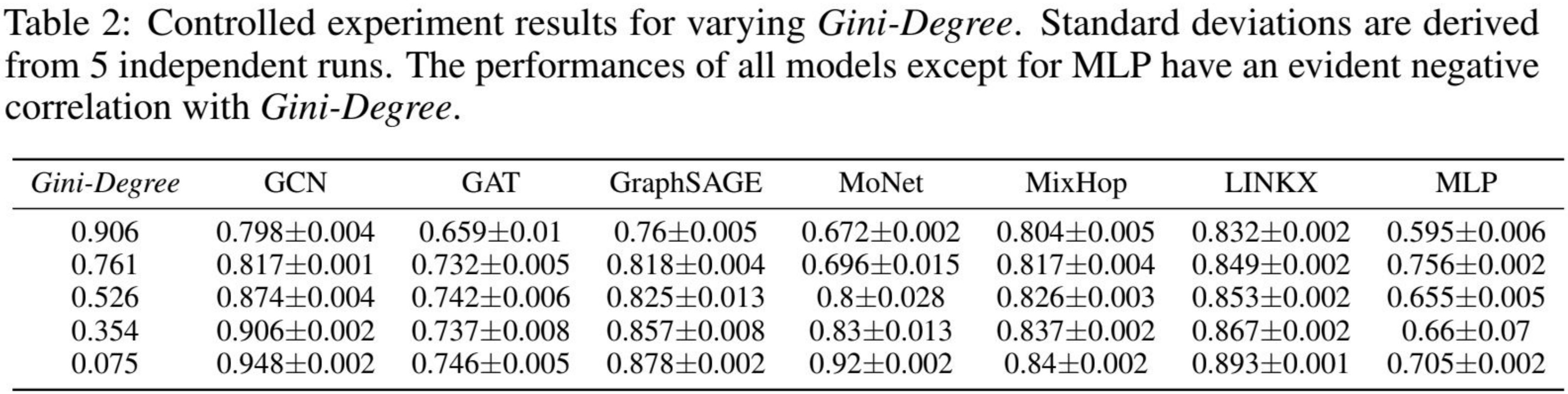


Table 2: Controlled experiment results for varying *Gini-Degree*. Standard deviations are derived from 5 independent runs. The performances of all models except for MLP have an evident negative correlation with *Gini-Degree*.

Gini-Degree	GCN	GAT	GraphSAGE	MoNet	MixHop	LINKX	MLP
0.906	0.798±0.004	0.659±0.01	0.76±0.005	0.672±0.002	0.804±0.005	0.832±0.002	0.595±0.006
0.761	0.817±0.001	0.732±0.005	0.818±0.004	0.696±0.015	0.817±0.004	0.849±0.002	0.756±0.002
0.526	0.874±0.004	0.742±0.006	0.825±0.013	0.8±0.028	0.826±0.003	0.853±0.002	0.655±0.005
0.354	0.906±0.002	0.737±0.008	0.857±0.008	0.83±0.013	0.837±0.002	0.867±0.002	0.66±0.07
0.075	0.948±0.002	0.746±0.005	0.878±0.002	0.92±0.002	0.84±0.002	0.893±0.001	0.705±0.002

Future Direction

- Further Investigation on other salient factors using other appropriate graph generation model and mathematical tools.
- Better GNN design to overcome the dependency on these dataset factors.
- AutoML perspective: when new datasets coming in, we can use the graph data properties to identify the most “similar” datasets where benchmark GNN performance is available. Then we can apply the best GNN design we know on these new datasets.

Thank you!!