Scale-Free, Attributed and Class-Assortative Graph Generation to Facilitate Introspection of Graph Neural Networks

Neil Shah
Snap Inc.
nshah@snap.com

ABSTRACT

Semi-supervised node classification on graphs is a complex interplay between graph structure, node features and class-assortative (homophilic) properties, and the flexibility of a model to capture these nuances. Modern datasets used to push the frontier for such tasks exhibit diverse properties across these aspects, making it challenging to study how these properties individually and jointly influence performance of modern methods like graph neural networks (GNNs). In this work, we propose an intuitive and flexible scale-free graph generation model, CaBaM, which enables simulation of class-assortative and attributed graphs via the well-known Barabasi-Albert model. We show empirically and theoretically how our model can easily describe a variety of graph types, while imbuing the generated graphs with the necessary ingredients for attribute, topology, and label-aware semi-supervised node-classification. We hope our work illustrates the need for graph generation and provides a stepping stone compensating for the lack of manipulability offered in common public graph dataset benchmarks. We also hope this inspires future work towards (a) more principled evaluation and study of GNNs, specifically their sensitivity to varying assortativity and attribute distributions, and (b) development of GNN architectures which facilitate graph context-awareness in line with these properties.

KEYWORDS

graph generation, preferential attachment, assortativity, network embedding

ACM Reference Format:
Neil Shah. 2020. Scale-Free, Attributed and Class-Assortative Graph Generation to Facilitate Introspection of Graph Neural Networks. In *MLG '20: ACM Symposium on Neural Gaze Detection, August 24, 2020, San Diego, CA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Semi-supervised learning on graphs (SSL) is a well-known task, which has gained renewed interest in recent years with the advances of neural node embedding methods, particularly graph neural networks (GNNs) [14, 15, 22, 33, 39, 40, 45]. In modern instances of such tasks, one is typically given a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ (with adjacency $\mathbf{A}$), node features $\mathcal{H}$ and labels y (potentially many of which are undefined). The task is to correctly infer the status of nodes which have undefined labels. Modern graph-based machine learning methods for this problem generally involve learning an embedding function $f : \mathcal{V} \rightarrow \mathbb{R}^d$ which maps each node into a high-dimensional space, where it can be subsequently classified. Most advances in recent years on this task have arisen from various novelties in parameterizing and learning f. While understanding of the importance of architectural, loss-based and situational choices for f has improved substantially in recent years [27, 43], there is comparatively little work in understanding the importance of $\mathcal{G}$ ($\mathbf{A}$), $\mathbf{H}$ and y to the performance of various methods.

This is in large part due to convention and limitations in existing benchmark datasets for evaluating performance on this task. The graph-based machine learning community commonly utilizes several datasets to demonstrate outperformance of a model. These benchmark datasets include citation networks (Cora, Citeseer [22]), protein-protein interactions (PPI [14]), social networks (Flickr, BlogCatalog [18]), air traffic (Air-USA [42]) and more. Recently, [17] curated and released several additional benchmark datasets to improve representation of other domains and standardize method comparisons. Nonetheless, these benchmark datasets have non-homogeneous properties that are not well-characterized or typically considered, making the performance analysis between different methods and graph types challenging to analyze. While the graphs may have similar structure in a skewed, power-law topology sense, they may have (a) very different attribute distributions across nodes (conditional on class), and (b) varying assortative (homophilic) tendencies between nodes of the same class. Both of these could influence the inherent difficulty of the learning task, and limit or facilitate different models.

To facilitate this analysis, we turn to graph generation models, a staple in the network science community [4, 9, 23, 25, 34, 36, 37, 41]. Graph generation models aim to simulate graphs which match (in a statistical sense) various observed processes. For example, [4, 25] aim to model scale-freeness and power-law degree distribution evident in many social networks, [36] aims to preserve local and global topological motifs, and [9] produces random graphs with interesting mathematical properties. Unfortunately, most of these models only generate topology and not attributes, with the exception of [34, 37], which are used in the context of mimicking a given graph, rather than flexibly manipulating the generation process to handle different graph settings. None of the above models explicitly facilitates analysis of the previously mentioned aspects.

In lieu, we propose CaBaM. Our work builds on the Barabasi-Albert (BA) [4] generation model for generating scale-free networks via preferential attachment. Since the model only produces $\mathcal{G}$ ($\mathbf{A}$), we extend it in two key ways which facilitate investigation of SSL

MLG '20, August 24, 2020, San Diego, CA

ACM ISBN 978-X-XXXX-XXXX-X/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

methods: by allowing nodes to flexibly (a) belong to classes, and be associated with the associated (arbitrary) attribute distribution, (b) vary their class-assortativity (for example, based on degree), thereby enabling flexible designation of $\mathbf{H}$ and y choices. Moreover, we show empirically and theoretically both that our extensions preserve the natural degree distribution of the original BA model, and that they have derivable class-assortativity dynamics in terms of the expected number of intra-class edges and inter-class edges in the generated graph.

We hope our model helps facilitate analysis of strengths and weaknesses of relative graph-based machine learning methods and the impact of graph structure, attribute distribution and assortativity on performance, particularly across GNN models which have become prominent in recent years. Moreover, we hope that graph generation via our model appeals to practitioners and researchers who work on development of GNNs to evaluate their models on well-specified and importantly, *tunable* benchmark datasets. We make our code and model publicly available at https://github.com/nshah171/cabam-graph-generation.

2 RELATED WORK

We discuss related work in two settings: graph-based semi-supervised learning, and graph generation models.

Graph-based semi-supervised learning (SSL). Graph-based SSL has a rich history, stemming from early methods such as label spreading [48], label propagation [49], belief propagation [46], and various random-walk and guilt-by-association approaches [24], many of which utilize only graph structure and no attributed information. It has many applications, including stereo matching in vision [38], top-n recommendation [13], fraud and misinformation detection [1, 12, 32], general node classification tasks [45] and more. In recent years, node representation learning techniques [11, 33, 39, 45] and more recently convolutional graph neural networks (GNNs) [14, 22, 40, 43] have dominated the landscape, achieving remarkable performance by proposing neural architectures for the SSL task. Unlike our work-in-progress, most recent works in this space focuses on model architecture improvements, rather than building understanding of model sensitivity to data. However, some touch on relevant spaces: [35, 43] give careful analysis of GNN expressivity given the aggregation operator. [2, 3] aim to incorporate flexible attention to varying k-hop graph contexts during representation learning. [42] explicitly incorporates node degree information into embeddings. [47] bootstraps node classification with self-supervised edge prediction, inherently leveraging structural predictability in the graph. However, none of these works investigate GNN performance contextually across various class attribute and assortativity settings.

Graph Generation Models. Numerous graph generation models have been proposed to capture, generate, and simulate network structure. [9] introduce the Erdos-Renyi random graph model, which generates graphs with independently drawn edges. [23] proposes a block, two-level extension which can be tuned to capture real-world degree distributions and clustering coefficients. Several other works [19, 31] extend Erdos-Renyi-inspired models to non-binary cases and multi-view graph settings. [25] proposes a model to generate edge structure via self-similarity induced by Kronecker products, and infer seed matrices given an input graph. [5, 41] enable production of various graphs with prescribed degree sequences which meet certain structural properties, like existence of a hub or connected components. [36] proposes an approach to generate graphs using concepts from context-free grammars. All the above methods are applicable only to graph data without attributes. [4] proposes a preferential-attachment model, by which nodes join a graph and attach to other nodes with higher degree with a higher probability. [21, 26, 34] discuss models which enable inference and mimicking connectivity given attributes from an input graph, but not flexible simulation of a new graph. The graph generation process discussed in our work differs from these by focusing on flexible simulation of graphs with class-imbued nodes and attributes without an input graph to mimic.

Several works also tackle producing assortativity in generated graphs, but mainly in the context of joint degree distributions. [30] discusses this concept: namely, that nodes tend to connect to others with similar degrees. [44] produces degree assortativity via an edge rewiring process from nodes in an existing graph. [28] uses accept-reject sampling to only keep edges from the model from [41] which satisfy a binned joint degree distribution. [7] modifies the Barabasi Albert (BA) model [4] for assortative mixing, but for degree sequence assortativity (nodes connect to others with similar degree) and with different motive. Closest to our work is [20], which studies an extension to the BA model for class-assortativity (nodes tend to connect to other nodes of the same class) aware preferential attachment, but differs from ours in both (a) model setup: their model does not discuss varying specifications of degree-dependent assortativity within a network, graphs with more than 2 classes, nor node attribute-aware settings, and (b) motivation: their work exposes the impact of class-assortativity to degree rankings between nodes of two classes, without any consideration of generation for evaluation of graph neural networks. [3] leverages [20]'s generation process to examine performance of their own model under various amounts of assortativity. In the same vein, our work here aims to provide a standard and general setup for flexible graph generation for varying numbers of classes, attributes and assortativity properties to facilitate introspection and analysis of GNNs and other graph-based SSL models. To provide further context and motivate these needs, we additionally illustrate huge differences in properties of benchmark datasets used in such tasks, which may be of independent interest to practitioners who design and evaluate such models.

2.1 The case for graph generation

Graphs used for SSL benchmarks share some topological similarities, but have stark differences in their inherent properties which make comparing and reasoning across SSL-based methods which implicitly rely on some assumptions on feature distributions and assortativity challenging. For example, GNNs typically make node-level label predictions by convolving features from neighboring nodes and thereby smoothing decision boundaries [14]. Clearly, these predictions may be impacted by the characteristics of the node's feature distribution (impacting variance in the convolution result), whether the node's degree is low or high (impacting the size of the receptive field), and how diverse or homogeneous the node's

	Cora	CoraML	Citeseer	Pubmed	DBLP	Air-USA	Flickr	BlogCatalog
# Nodes	19,793	2,995	4,230	19,717	17,716	1,190	7,575	5,196
# Edges	146,635	19,311	14,904	108,365	123,450	28,388	487,051	348,682
# Features	8,710	2,879	602	500	1639	238	12,047	8,189
Feature-degree assoc.	.628	.594	.524	.668	.571	1.0	.619	.666
Class assortativity	.670	.845	.972	.863	.871	.722	.262	.419
Cross-class MMD	.233	.102	.151	.156	.070	.491	.023	.139

Table 1: Characteristic differences in properties of commonly used benchmark datasets.

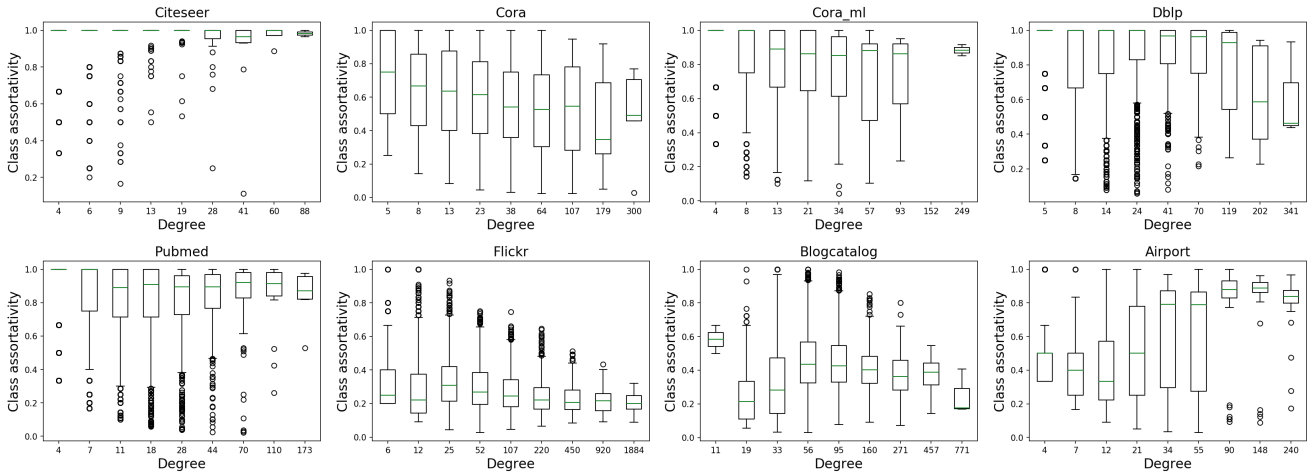


Figure 1: Class-assortativity varies substantially across commonly used graph datasets, and further varies even within datasets for nodes with different degrees. Boxplots show class-assortativity (y-axis) computed for nodes which are logarithmically binned by degree (x-axis).

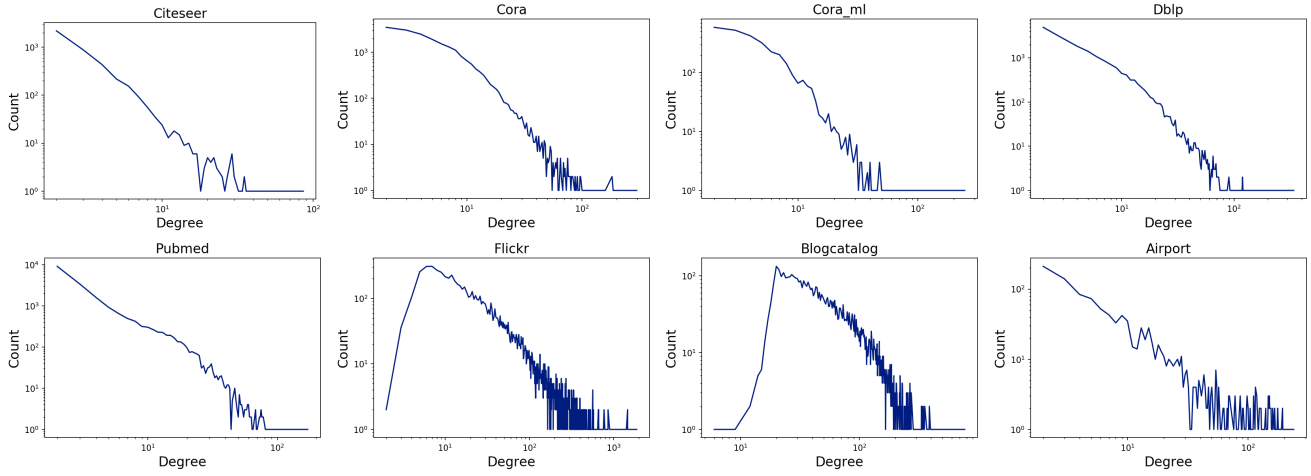


Figure 2: Degree distributions for 8 common benchmark datasets. Despite feature and class-assortativity-related differences, most common benchmark datasets have skewed, power-law-esque degree distributions, making this property desirable for a graph generator to simulate.

neighborhood (impacting the smoothness of resulting boundaries). Studying these properties and how they influence various methods can hopefully lead to further research into, and new insights about

method design, situational relevance and an increased emphasis on the *graph* in the context of model architecture.

To illustrate diverse properties across existing datasets, we compute a few properties over a small benchmark suite: we use citation networks (Cora, CoraML, Citeseer, Pubmed, DBLP), airport traffic (Air-USA), and social networks (Flickr, BlogCatalog). Table 1 details (in addition to summary statistics), three properties which capture some notable differences across the datasets relating to the aforementioned properties.

Feature-degree association. We take a simple Random Forest classifier, and try to predict whether a node's degree is higher or lower than the graph's average degree using the input features. We use 5-fold cross validation with stratification, reporting the AUC (area under ROC curve). High numbers imply that features (and associated classes) are correlated to the graph topology, effectively creating distinct convolution dynamics for nodes of one class versus another.

Class assortativity. We count the total number of intra-class (within one class) and inter-class (between different classes) edges in the graph and report the intra-class edge fraction

$$r_{e_w} = \frac{e_w}{e_w + e_b}$$

where e_w, e_b denote intra/inter-class edge counts respectively. This is effectively a measure of homophily. High values indicate that most edges in the graph are between nodes in the same class, with implications in the smoothness of decision boundaries. Low values indicate that convolution can easily blur decision boundaries between classes.

Between-class MMD. We report the maximum pairwise maximum mean discrepancy (MMD) statistic [10] across classes. We calculate the cross-class MMD as

$$\max_{\forall x, y \in C} \sup \left(\mathbb{E}[x] - \mathbb{E}[y]\right)$$

where $\mathbb{E}[x]$ (wlog) refers to the empirical expected value of (unit-normalized) samples from $\mathbf{H}$ in class x, and x, y are classes in the set C. This is bounded on $[0, 1]$. High values generally indicate that features from classes are more easily separable (i.e. that class x and y have very different feature distributions), whereas low values indicate less distinction.

As is evident from Table 1, the properties vary considerably across datasets. Notably, features are perfectly predictive of above-average degree prediction in Air-USA (but much less so in other datasets). Citeseer has a markedly high class-assortativity, with 97.2% of the edges in the dataset being intra-class ones, while Flickr only has 26.2% such edges, suggesting huge differences in assortativity between the two datasets.

Figure 1 further shows that assortativity can vary significantly even within a graph for nodes with different degrees. MMD differences suggest that features across classes from Flickr and DBLP are less discriminating, whereas they are much more so for Air-USA and Cora. Despite all these variations, Figure 2 shows some marked topological similarities across datasets. Namely, all of them have skewed degree distributions, resembling power-law-esque or lognormal distributions. As numerous works [6, 8, 29] discuss, such degree distributions are commonly observed in empirical, and especially social, datasets, where preferential attachment-like phenomena are common and rich-get-richer effects prevail.

We note that our work here does not aim to address all of the above points and reconcile differences across datasets, but rather aims to expose the significant variations across these datasets which is typically ignored, encourage the reader to consider how these variations may influence downstream SSL model performance, and consider how they may be *systematically* studied. We propose graph generation as a tool to enable this systematic study.

3 GRAPH GENERATION WITH CABAM

Our expository analysis in the previous section shows that while most realistic graph datasets used for SSL tasks are topologically similar (at least in the degree distribution context), they vary considerably in their attribute/feature distributions and assortativity properties. Thus, we consider producing a graph generation model which is not only (a) able to produce graphs with power-law degree distribution, but also (b) is flexible in class and feature distributions for nodes, and (c) allows variation in class-assortativity. We focus on the Barabasi Albert (BA) model, which is simple, analytically shown to produce power-law degree distributions, and is appealing due to its explicit time-aware and preferential generative process. For context, the BA model [4] proceeds as follows:

(1) **Graph initialization.** Initialize a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ with m nodes and no edges.
(2) **Growth.** At each timestep t, add one node v with m edges to existing nodes. The probability of connecting to node w is given by

$$P_{v \to w} = \frac{k_w}{\sum_{y \in \mathcal{V}} k_y} \tag{1}$$

where k_w indicates w's degree (wlog).

Via this preferential attachment form, the BA model has been shown to have a degree distribution $P(k) \propto k^{-3}$, and considerable past analysis has been done on the model [4, 16], making it appealing for use. However, the original BA model does not account for any node classes or attributes, nor their assortativity.

Given these shortcomings, we propose our extended model, CaBaM. Our model has the following generative process:

(1) **Definitions.** Define a multinomial class distribution $\mathcal{M}$ over the $|C|$ classes. For each class $c \in C$, define a class-specific distribution $\mathcal{D}_i$, such that a node v belonging to class c will have its features $h_v \sim \mathcal{D}_c$. Lastly, define $p_c \in [0, 1]$ as a within-class assortativity factor.
(2) **Graph initialization.** Initialize a graph with m nodes and no edges. For each node, draw class and attributes as per (1).
(3) **Growth.** At each timestep t, add one node v. Assign it a class c and draw its attributes as per (1). Attach it to m existing nodes, with the probability of connecting to a node w as

$$P_{v \to w} = \frac{k_w \cdot (q \cdot p_c + (1-q) \cdot (1-p_c))}{\sum_{y \in \mathcal{V}} \left(k_y \cdot (q \cdot p_c + (1-q) \cdot (1-p_c))\right)} \tag{2}$$

where $q = \mathbb{1}(c_w = c_v)$, an indicator reflecting whether the two nodes belong to the same class.

While simple, this model admits a number of desirable properties. Firstly, it generates scale-free graphs with power-law degree distributions. Secondly, it allows flexible specification of class-conditional

node attributes, with the choice of $\mathcal{D}_c$ left open. Finally, it allows flexible specification of class-assortativity via p_c. In fact, in some cases, p_c need not be a constant, but can be a function defined on each node (for example, based on its degree to emulate effects like in Figure 1) while still retaining all these properties.

Our model admits the following theoretical analysis:

Theorem 3.1 (Degree distribution). *As $t \to \infty$, a graph $\mathcal{G}$ generated by CaBaM has a power-law degree distribution with $P(k) \propto k^{-3}$ and minimum degree m, given by*

$$P(k) = \frac{2 \cdot m \cdot (m+1)}{k \cdot (k+1) \cdot (k+2)} \tag{3}$$

if either (a) p_c is a constant in $[0, 1]$, or (b) $\mathbb{E}[q] = 1/2$.

Proof. Let $g = \mathbb{E}[q]$, the probability of a new node v and an existing node w being of the same class. Then, the change in the degree of a node with degree k_w can be written as

$$\frac{dk_w}{dt} = m \cdot \frac{k_w \cdot (g \cdot p_c + (1-g) \cdot (1-p_c))}{\sum_{y \in \mathcal{V}} \left(k_y \cdot (g \cdot p_c + (1-g) \cdot (1-p_c))\right)}$$

by taking the expectation over the indicator, and multiplying the quantity by m given m edges added per timestep. Notice that g only depends on the native class distribution and thus does not depend on $y \in \mathcal{V}$. If p_c is a constant, then $g \cdot p_c + (1-g) \cdot (1-p_c)$ can be moved outside the sum, so that dk_w/dt takes on the form in Eqn. 1. If p_c is not a constant but a function (for example, depending on the node $y \in \mathcal{V}$), then the quantity cannot be moved outside the sum. However, if $g = 1/2$, then the quantity reduces to 1 even inside the sum, again yielding the form in Eqn. 1. Next, notice that $\sum_{y \in \mathcal{V}} k_y = 2mt$, replacing the degree sum with total edge count. Thus, we have

$$\frac{dk_w}{dt} = \frac{k_w}{2t} = \frac{dt}{2t}, \quad \text{so} \quad k_w \propto t^{-1/2}$$

integrating both sides in the last step. Thus, a node's degree grows inversely-proportionally to the square-root of time. This leads us precisely to the degree dynamics of the BA model (see [4]), which derives the exact degree distribution (and exponent) as above from this point. Also note that a node must have degree $\geq m$ given the growth step. □

Our results shows that if non-constant p_c is desired, then $g = 1/2$ is mandated. This functionally constrains the choice of $\mathcal{M}$. If $|C| = 2$, then $\mathcal{M} = Multinomial([1/2, 1/2]$ suffices. If $|C| = 3$, then $\mathcal{M} = Multinomial([2/3, 1/6, 1/6]$ suffices. The choice of parameters given $|C|$ can generally be derived by equating the algebraic expressions for choosing two intra-class nodes and two inter-class nodes, and constraining all class probabilities to sum to 1. Formally, the parametric solutions for class probabilities $P_1 \ldots P_k$ are given by

$$\sum_{i=1}^{k} P_i = 1 \quad \text{and} \quad \sum_{i=1}^{k} P_i^2 = 1/2$$

Conversely, if a constant p_c is used, then any well-defined choice of parameters $(P_1 \ldots P_k)$ which sums to 1 for $\mathcal{M}$ suffices.

Theorem 3.2 (Class assortativity). *As $t \to \infty$, the intra-class edge fraction r_{e_w} of a graph $\mathcal{G}$ generated by CaBaM is*

$$r_{e_w} = \frac{p_{e_w}}{p_{e_w} + p_{e_b}} \tag{4}$$

where

$$p_{e_w} = \sum_{k=m}^{\infty} \frac{g \cdot p_c \cdot (m+1)}{(k+1) \cdot (k+2)} \tag{5}$$

$$p_{e_b} = \sum_{k=m}^{\infty} \frac{(1-g) \cdot (1-p_c) \cdot (m+1)}{(k+1) \cdot (k+2)} \tag{6}$$

if either (a) p_c is a constant in $[0, 1]$, or (b) $\mathbb{E}[q] = 1/2$.

Proof. Our derivation relies on the closed-form exact degree-distribution from Thm. 3.1, thus inheriting its assumptions. The probability of a link from a newly introduced node v to connect to a degree k node is given by

$$\frac{m+1}{(k+1) \cdot (k+2)} \tag{7}$$

since the probability of v linking to a node with degree k is $\frac{k}{2mt}$, and we multiply this by the number of nodes with degree k, or $t \cdot P(k)$ (see Eqn. 3) since there are t nodes in the graph at time t. The probability of v having the same class as the recipient node is $g = \mathbb{E}[q]$, and the assortativity multiplier probability for that link is given by p_c, accounting for the two factors in the numerator of Eqn. 5. Likewise, the probability of v having a different class as the recipient node is $1 - g$, and the dissortativity multiplier probability is $1 - p_c$, account for the two factors in the numerator of Eqn. 6. Summing from $m \ldots \infty$, p_{ew} and p_{eb} denote the (unnormalized) probabilities of v making an intra (inter)-class edge, over nodes with all degrees. Indeed,

$$\sum_{k=m}^{\infty} \frac{m+1}{(k+1) \cdot (k+2)} = 1$$

Note that if p_c is a constant, we can further reduce the (full) form of Eqn. 4 to

$$\frac{g \cdot p_c}{(g \cdot p_c) + (1-g) \cdot (1-p_c)}$$

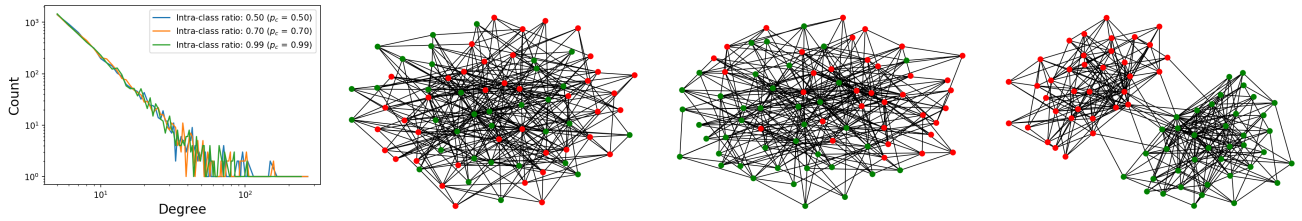
but this cannot be said when p_c depends on the node. Eqn. 4 normalizes the intra-class probability, yielding the result. □

In summary, our CaBaM model extends the BA model to simulate not only $\mathcal{G}$ (**A**), but also **H** and y while offering flexible designation of assortativity, while having several useful theoretical guarantees which can facilitate analysis.

4 EVALUATION

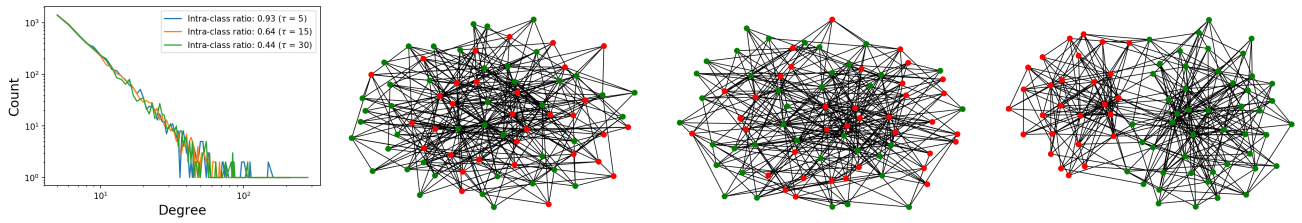
We next evaluate CaBaM's use in flexibly simulate graphs with varying properties for investigation. Our results show 3 key points:

- CaBaM produces scale-free graphs which admit a power-law degree distribution.
- CaBaM can be used to simulate graphs with varying class-assortativity specifications.
- CaBaM's generated graphs empirically match the "in-the-limit" properties with quite small graphs, in practice.



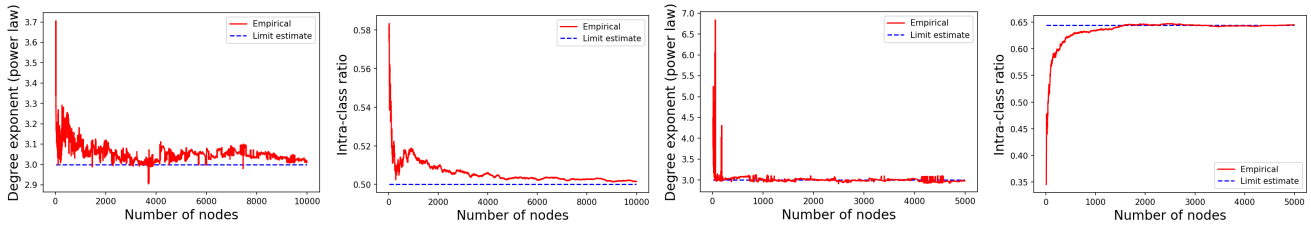
(a) Degree distribution **(b)** $p_c = 0.5$ **(c)** $p_c = 0.70$ **(d)** $p_c = 0.99$

Figure 3: CaBaM simulated graphs with constant p_c values. All simulations (estimated with $|\mathcal{V}| = 10000$) obey the same underlying power-law degree distribution (a). (b-d) show graph snapshots at $|\mathcal{V}| = 75$, demonstrating the varying assortative tendencies with varying constant p_c.



(a) Degree distribution **(b)** $p_c(k) = \tanh k/30$ **(c)** $p_c(k) = \tanh k/15$ **(d)** $p_c(k) = \tanh k/5$

Figure 4: CaBaM simulated graphs with degree-dependent p_c values, defined as $p_c(k) = \tanh k/\tau$. All simulations (estimated with $|\mathcal{V}| = 10000$) obey the same underlying power-law degree distribution (a). (b-d) show graph snapshots at $|\mathcal{V}| = 75$, demonstrating the varying assortative tendencies with varying choices of temperature τ which parameterize p_c.



(a) α **(**$p_c = 0.5$**)** **(b)** r_{e_w} **(**$p_c = 0.5$**)** **(c)** α **(**$p_c(k) = \tanh k/15$**)** **(d)** r_{e_w} **(**$p_c(k) = \tanh k/15$**)**

Figure 5: CaBaM simulated graphs converge to limit estimates (as per Theorems 3.1-3.2) in both constant ($p_c = 0.5$, left) and degree-dependent ($p_c(k) = \tanh k/15$, right) settings. Estimates in both cases depend on graph size and are derived from MLE. Red lines indicate parameter estimates, and blue, dashed-lines the theoretical limit quantities.

4.1 Experimental Setup

We write our simulation code using Python 3.7, and show graph visualization with the `networkx` package. All experiments are conducted on a single `n1-standard-16` Google Cloud Platform virtual machine.

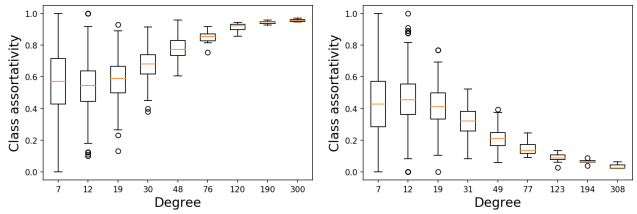
Unless otherwise specified, we set parameters for CaBaM to

- $|\mathcal{V}| = 5000$ maximum (grow the graph to 5,000 nodes),
- $m = 5$ (initialize graph with 5 nodes, and add 5 edges per new node),
- $c = 2$ (let nodes belong to two classes)
- $\mathcal{M} = Multinomial([1/2, 1/2])$ (let both classes be equally prevalent)

We show results in two different contexts of p_c: constant, and degree-dependant. Constant p_c implies that p_c is simply a scalar in $[0, 1]$ which can be tuned from low-high prioritization of assortativity. For degree-dependence, we use $p_c(k) = \tanh k/\tau$, where k is a node's degree, and τ is a temperature parameter controlling the steepness of the tanh function. This allows nodes with different degrees to have different assortativity behaviors. We note that any suitably defined function which outputs scalars suitable for interpretation as probabilities could be used, instead of our choice.

4.2 Power-law degree distribution

Figures 3a and 4a show the degree distributions for 3 constant p_c and 3 degree-dependant p_c specifications, respectively. Notice that



(a) $p_c(k) = \tanh k/15$ **(b)** $p_c(k) = 1 - \tanh k/15$

Figure 6: CaBaM can generate graphs with custom, degree-dependent class-assortativity, mimicking real graphs.

the distributions are indistinguishable despite the varying assortativity properties, which agrees with Theorem 3.1. The power-law degree distribution exponent α is evidently 3.0, showing typical linear behavior in logarithmic-scales. Figure 5a/c shows that this approximate characteristic degree exponent is estimated even for small graphs with hundreds of nodes (along the x-axis), both for constant and degree-dependent p_c, respectively – red lines show maximum likelihood estimates for α when fitting the degree distribution on the (growing) graph. and blue (dashed) lines show the limit quantity according to Theorem 3.1.

4.3 Class-assortativity

Figures 3b-d and 4b-d show visualizations of growing networks with 3 constant p_c and 3 degree-dependant p_c specifications, respectively. The visualization is conducted with the graph at $|\mathcal{V}| = 75$ nodes to prevent overplotting effects. Nodes are colored red or green according to their class designation. Note that with varying specifications of p_c in both cases, we can achieve roughly equally assortative/dissortative graphs in (b) with roughly equal intra/inter-class edges, moderately assortative graphs in (c) and highly assortative graphs in (d) while preserving the degree distribution. These nodes could be further imbued with any class-specific attribute distribution $\mathcal{D}_i$, allowing the simulation of a wide class of graph types. Figure 5b/d shows that estimated class assortativity quantities (in red) approximately match the limit quantity from Theorem 3.2 (in blue, dashed) for small graphs with hundreds of nodes (along the x-axis), both for constant and degree-dependent p_c, respectively. Moreover, 6 shows two different parameterizations of degree-dependent p_c can yield varying degree-dependent assortativities for simulated graphs, matching observations from Figure 1 on real datasets. Careful designation of these p_cs can yield study of arbitrarily assortative or dissortative graphs.

5 USAGE IN PRACTICE

We make available code for the model discussed in this work at https://github.com/nshah171/cabam-graph-generation. While the theoretical properties regarding degree distribution and assortativity in the limit discussed in Section 3 hold under fairly loose conditions, it is worth noting that the generative process for CaBaM is still usable even outside the scope of these conditions to produce attribute-imbued graphs with varying class-assortativity under a preferential attachment regime, albeit without "nice results" for these properties. Arguably, these are not strictly required for graph generation to usefully facilitate analysis of GNNs; in short, even without the theoretical underpinnings, the generative process allows reproducible and flexible generation of graphs with various desiderata, enabling careful empirical analysis which is attentive to these desiderata and the compatibility of existing models in handling them. While there are likely many uses for our model, several important ones include evaluating GNNs (a) on graphs with varying assortativity/dissortativity, (b) on graphs with arbitrarily noisy or pristine features, (c) evaluating the relative value of GNNs versus MLPs under different data scenarios and more.

6 CONCLUSION AND FUTURE WORK

The success of graph-based semi-supervised learning relies on a mixture of factors, including graph structure, node features and class-assortativity between nodes. Modern datasets used to benchmark such methods are few, and quite diverse in several properties relating to node features and relative assortativity, making it difficult to comparatively analyze why some methods perform better on some datasets, and in what cases implicit assumptions can be revised to improve model performance. Graph generation is a promising approach to simulating graph data with various properties for this careful analysis. Thus, in this work (in progress), we introduce CaBaM, a model for generating scale-free, class-aware and assortative graphs which builds upon the celebrated Barabasi-Albert model which exhibits preferential attachment properties. Our model enables generation of flexible $\mathbf{G}$ (via adjacency $\mathbf{A}$), node classes y and class-specific node-features $\mathbf{H}$. We hope our findings and remarks regarding benchmark property variations, relative opacity of various GNN models' performances across these contexts, and proposed graph generation modeling framework inspire future work in evaluating existing methods under different data conditions, and designing new ones with context-aware architecture improvements.

REFERENCES

[1] Sara Abdali, Neil Shah, and Evangelos E Papalexakis. 2020. HiJoD: Semi-Supervised Multi-aspect Detection of Misinformation using Hierarchical Joint Decomposition. *arXiv preprint arXiv:2005.04310* (2020).
[2] Sami Abu-El-Haija, Bryan Perozzi, Rami Al-Rfou, and Alexander A Alemi. 2018. Watch your step: Learning node embeddings via graph attention. In *Advances in Neural Information Processing Systems*. 9180–9190.
[3] Sami Abu-El-Haija, Bryan Perozzi, Amol Kapoor, Nazanin Alipourfard, Kristina Lerman, Hrayr Harutyunyan, Greg Ver Steeg, and Aram Galstyan. 2019. Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing. *arXiv preprint arXiv:1905.00067* (2019).
[4] Albert-László Barabási et al. 2016. *Network science*. Cambridge university press.
[5] Tom Britton, Maria Deijfen, and Anders Martin-Löf. 2006. Generating simple random graphs with prescribed degree distribution. *Journal of statistical physics* 124, 6 (2006), 1377–1397.
[6] Andrea Capocci, Vito DP Servedio, Francesca Colaiori, Luciana S Buriol, Debora Donato, Stefano Leonardi, and Guido Caldarelli. 2006. Preferential attachment in the growth of social networks: The internet encyclopedia Wikipedia. *Physical review E* 74, 3 (2006), 036116.
[7] Michele Catanzaro, Guido Caldarelli, and Luciano Pietronero. 2004. Social network growth with assortative mixing. *Physica A: Statistical Mechanics and its Applications* 338, 1-2 (2004), 119–124.
[8] Aaron Clauset, Cosma Rohilla Shalizi, and Mark EJ Newman. 2009. Power-law distributions in empirical data. *SIAM review* 51, 4 (2009), 661–703.
[9] Paul Erdős and Alfréd Rényi. 1960. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci* 5, 1 (1960), 17–60.
[10] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. 2012. A kernel two-sample test. *Journal of Machine Learning Research* 13, Mar (2012), 723–773.

[11] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 855–864.

[12] Gisel Bastidas Guacho, Sara Abdali, Neil Shah, and Evangelos E Papalexakis. 2018. Semi-supervised content-based detection of misinformation via tensor embeddings. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE, 322–325.

[13] Jiwoon Ha, Soon-Hyoung Kwon, Sang-Wook Kim, Christos Faloutsos, and Sunju Park. 2012. Top-N recommendation through belief propagation. In *Proceedings of the 21st ACM international conference on Information and knowledge management*. 2343–2346.

[14] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*. 1024–1034.

[15] Mark Heimann, Tara Safavi, and Danai Koutra. 2019. Distribution of Node Embeddings as Multiresolution Features for Graphs. ICDM.

[16] Petter Holme and Beom Jun Kim. 2002. Growing scale-free networks with tunable clustering. *Physical review E* 65, 2 (2002), 026107.

[17] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. 2020. Open graph benchmark: Datasets for machine learning on graphs. *arXiv preprint arXiv:2005.00687* (2020).

[18] Xiao Huang, Jundong Li, and Xia Hu. 2017. Label informed attributed network embedding. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. 731–739.

[19] Meng Jiang, Alex Beutel, Peng Cui, Bryan Hooi, Shiqiang Yang, and Christos Faloutsos. 2015. A general suspiciousness metric for dense blocks in multimodal data. In *2015 IEEE International Conference on Data Mining*. IEEE, 781–786.

[20] Fariba Karimi, Mathieu Génois, Claudia Wagner, Philipp Singer, and Markus Strohmaier. 2018. Homophily influences ranking of minorities in social networks. *Scientific reports* 8, 1 (2018), 1–12.

[21] Ji Youn Kim, Michael Howard, Emily Cox Pahnke, and Warren Boeker. 2016. Understanding network formation in strategy research: Exponential random graph models. *Strategic management journal* 37, 1 (2016), 22–44.

[22] Thomas N Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.

[23] Tamara G Kolda, Ali Pinar, Todd Plantenga, and Comandur Seshadhri. 2014. A scalable generative graph model with community structure. *SIAM Journal on Scientific Computing* 36, 5 (2014), C424–C452.

[24] Danai Koutra, Tai-You Ke, U Kang, Duen Horng Polo Chau, Hsing-Kuo Kenneth Pao, and Christos Faloutsos. 2011. Unifying guilt-by-association approaches: Theorems and fast algorithms. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 245–260.

[25] Jure Leskovec, Deepayan Chakrabarti, Jon Kleinberg, Christos Faloutsos, and Zoubin Ghahramani. 2010. Kronecker graphs: An approach to modeling networks. *Journal of Machine Learning Research* 11, Feb (2010), 985–1042.

[26] Dean Lusher, Johan Koskinen, and Garry Robins. 2013. *Exponential random graph models for social networks: Theory, methods, and applications*. Cambridge University Press.

[27] Haggai Maron, Heli Ben-Hamu, Hadar Serviansky, and Yaron Lipman. 2019. Provably powerful graph networks. In *Advances in Neural Information Processing Systems*. 2153–2164.

[28] Stephen Mussmann, John Moore, Joseph John Pfeiffer, and Jennifer Neville. 2015. Incorporating assortativity and degree dependence into scalable network models. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

[29] Mark EJ Newman. 2001. Clustering and preferential attachment in growing networks. *Physical review E* 64, 2 (2001), 025102.

[30] Mark EJ Newman. 2002. Assortative mixing in networks. *Physical review letters* 89, 20 (2002), 208701.

[31] Hamed Nilforoshan and Neil Shah. 2019. SliceNDice: Mining Suspicious Multi-attribute Entity Groups with Multi-view Graphs. *arXiv preprint arXiv:1908.07087* (2019).

[32] Shashank Pandit, Duen Horng Chau, Samuel Wang, and Christos Faloutsos. 2007. Netprobe: a fast and scalable system for fraud detection in online auction networks. In *Proceedings of the 16th international conference on World Wide Web*. 201–210.

[33] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 701–710.

[34] Joseph J Pfeiffer III, Sebastian Moreno, Timothy La Fond, Jennifer Neville, and Brian Gallagher. 2014. Attributed graph models: Modeling network structure with correlated attributes. In *Proceedings of the 23rd international conference on World wide web*. 831–842.

[35] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2008. Computational capabilities of graph neural networks. *IEEE Transactions on Neural Networks* 20, 1 (2008), 81–102.

[36] Satyaki Sikdar, Justus Hibshman, and Tim Weninger. 2019. Modeling Graphs with Vertex Replacement Grammars. *arXiv preprint arXiv:1908.03837* (2019).

[37] Tom AB Snijders. 2002. Markov chain Monte Carlo estimation of exponential random graph models. *Journal of Social Structure* 3, 2 (2002), 1–40.

[38] Jian Sun, Heung-Yeung Shum, and Nan-Ning Zheng. 2002. Stereo matching using belief propagation. In *European Conference on Computer Vision*. Springer, 510–524.

[39] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*. 1067–1077.

[40] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).

[41] M Winlaw, H DeSterck, and G Sanders. 2015. *An in-depth analysis of the chung-lu model*. Technical Report. Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States).

[42] Jun Wu, Jingrui He, and Jiejun Xu. 2019. DEMO-Net: Degree-specific graph neural networks for node and graph classification. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 406–415.

[43] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826* (2018).

[44] R Xulvi-Brunet and IM Sokolov. 2004. Construction and properties of assortative random networks. *arXiv preprint cond-mat/0405095* (2004).

[45] Zhilin Yang, William W Cohen, and Ruslan Salakhutdinov. 2016. Revisiting semi-supervised learning with graph embeddings. *arXiv preprint arXiv:1603.08861* (2016).

[46] Jonathan S Yedidia, William T Freeman, and Yair Weiss. 2001. Generalized belief propagation. In *Advances in neural information processing systems*. 689–695.

[47] Tong Zhao, Yozen Liu, Leonardo Neves, Oliver Woodford, Meng Jiang, and Neil Shah. 2020. Data Augmentation for Graph Neural Networks. *arXiv preprint:2006.06830* (2020).

[48] Dengyong Zhou, Olivier Bousquet, Thomas N Lal, Jason Weston, and Bernhard Schölkopf. 2004. Learning with local and global consistency. In *Advances in neural information processing systems*. 321–328.

[49] Xiaojin Zhu and Zoubin Ghahramani. 2002. Learning from labeled and unlabeled data with label propagation. (2002).