# Unified Representation and Lifted Sampling for Generative Models of Social Networks

**Pablo Robles-Granda**[1], **Sebastian Moreno**[2], and **Jennifer Neville**[1]
[1]Department of Computer Science, Purdue University
[2]Faculty of Engineering and Science, Universidad Adolfo Ibañez
[1]{problesg,neville}@purdue.edu, [2]sebastian.moreno@uai.cl

## Abstract

Statistical *models* of network structure are widely used in network science to reason about the properties of complex systems—where the nodes and edges represent entities and their relationships. Recently, a number of *generative network models* (GNM) have been developed that accurately capture characteristics of real world networks, but since they are typically defined in a procedural manner, it is difficult to identify commonalities in their structure. Moreover, procedural definitions make it difficult to develop statistical sampling algorithms that are both efficient and correct. In this paper, we identify a family of GNMs that share a common latent structure and create a Bayesian network (BN) representation that captures their common form. We show how to reduce two existing GNMs to this representation. Then, using the BN representation we develop a generalized, efficient, and provably correct, sampling method that exploits parametric symmetries and *deterministic context-specific dependence*. Finally, we use the new representation to design a novel GNM and evaluate it empirically.

## 1 Introduction

Many complex systems are modeled using networks—where nodes represent entities and edges represent some type of relation between the entities. An open problem in network science is how to create generative models that accurately capture the characteristics of real world, sparse networks, in order to better understand the properties of the systems, and to facilitate statistical analysis.

Recent research has explored various models to accurately characterize network structure (e.g., [Chung and Lu, 2002; Hoff, 2008; Leskovec *et al.*, 2010; Benson *et al.*, 2014]). A number of these *generative network models* (GNMs) use a common procedure to generate the edges in the network using a matrix of probabilities. The approaches differ in the number and structure of parameters used to specify the edge probabilities. While several GNMs can capture important characteristics of real world networks such as power law degree distributions and community structure, it has been more difficult to develop methods to capture longer-range dependencies that impact global characteristics. To address these limitations, new hierarchical GNMs have been proposed with more complex dependencies between the edge probabilities, e.g., mixed Kronecker product graph model (mKPGM) [Moreno *et al.*, 2013], Block two-level Erdős-Rényi Model (BTER) [Seshadhri *et al.*, 2012], bipartite stochastic block model (biSBM) [Larremore *et al.*, 2014], and hierarchical graph models [Peixoto, 2014].

Moreover, while GNMs specify a generative process, it can still be computationally intensive to naively sample large networks. There have been efficient sampling methods proposed for some GNMs [Leskovec *et al.*, 2010; Yun and Vishwanathan, 2012], but because models are specified procedurally it is difficult to guarantee correctness, and consequently the efficient sampling methods can generate improbable network structures [Moreno *et al.*, 2014]. Thus, it is still an open question how to develop accurate and efficient sampling methods. Another challenge is to identify commonalities across GNMs. Again, since models are specified procedurally, it is difficult to discern the impact of algorithmic differences–both w.r.t. the number and structure of parameters–except by comparing the structure of the networks that are generated from the methods.

In this paper, we show that hierarchical GNMs can be abstracted as Bayesian networks (BNs) with random variables (RVs) representing the existence of edges in the network, and hierarchies of latent RVs representing relations among groups of edges. With this representation we can identify common properties of unrelated models, such as mKPGM and BTER, and use the knowledge to guide the design of new GNMs. The BN representation also facilitates the development of a general sampling method that is both efficient and accurate. The transformation of a GNM to BN representation makes it easy to identify *parametric symmetries* allowing us to sample groups of edge RVs rather than sequentially—using insights from *lifted inference* [Poole, 2003; Jha *et al.*, 2010; Van den Broeck and Darwiche, 2013; Mittal *et al.*, 2015]. Moreover, we can maximize computational efficiency and guarantee correctness by using insights from *context specific independence* [Boutilier *et al.*, 1996; Poole and Zhang, 2003] to *dynamically sparsify* the set of RVs to be sampled based on the context of parent RV values.

Previous work had provided important insights about the relationship among some GNMs (e.g. [Jacobs and Clauset, 2014]) and BNs have been used to model GNMs (e.g.

[Schein *et al.*, 2016; Liang *et al.*, 2016; Ho *et al.*, 2011; Neiswanger *et al.*, 2014]), However, to our knowledge, this work is the first attempt to model a family of *existing* GNMs in a principled way that is general enough to facilitate the creation of new classes of GNMs. We do that by taking advantage of BNs to model GNMs but without enforcing restrictions on the topology of a BN except for a hierarchical conditional dependence which we describe in this paper.

In summary, the key contributions of this work are:

1. Demonstration that hierarchical GNMs can be represented in a universal, succinct way that supports the development of new models.
2. Development of a general efficient, and provably correct, sampling algorithm that applies to all GNMs in the family of hierarchical GNMs. Our algorithm is based on two important properties:
   (a) *Deterministic context specific dependence*–which sparsifies the RV sampling space.
   (b) *Parametric symmetries*–which, due to GNM parameterizations, can be exploited with *lifted* sampling.
3. Design, and evaluation, of a new GNM to illustrate the utility of the BN representation; in particular, of the related sampling algorithm.

## 2 Background: Generative Network Models

First, we review some details about generative network models. Let $G=(\mathbf{V},\mathbf{E})$ be a network with set of vertices $\mathbf{V}$ and edges $\mathbf{E}\subset\mathbf{V}\times\mathbf{V}$. We define a generative network model (GNM) $\mathcal{M}$ with parameters $\Theta$ as follows.

**Definition 1.** *Generative network model (GNM)*
*A GNM is a statistical model $\mathcal{M}$ with parameters $\Theta$ that define, either explicitly or via a construction process, a size $|\mathbf{V}|\times|\mathbf{V}|$ matrix $\mathcal{P}$ of probabilities. Each cell $[i,j]\in\mathcal{P}$ corresponds to the binary random variable $A_{ij}$ and the value $\mathcal{P}_{ij}$ represents the Bernoulli probability that the [un]directed edge $e_{i,j}$ exists in the network (i.e., if $A_{ij}=1$ then $e_{i,j}\in\mathbf{E}$ and $P(A_{ij}=1)=\mathcal{P}_{ij}$). Thus, $\mathcal{P}$ models the structure of the network through the set of binary random variables $A_{ij}$ $\forall i,j\in\{1,\ldots,|\mathbf{V}|\}$.*

The size of $\Theta$ differs for each GNM and can vary in the range $[1,|\mathbf{V}|^2]$. This restriction in the number of parameters is intended to make sampling of random graphs efficient. However, most GNMs have a small number of parameters (usually $|\Theta|\ll|\mathbf{V}|$) to avoid overfitting. Thus, multiple cells in $\mathcal{P}$ (and RVs $A$) could have the same probability values. We use $<\mathbf{U},\mathbf{T}>$ to represent this, where $\mathbf{U}$ is the set of unique probabilities that appear in the matrix $\mathcal{P}$, i.e. $\mathbf{U}=\texttt{unique}(\mathcal{P})=\{\pi_1,\pi_2,\ldots,\pi_u,\ldots\pi_\kappa\}$, and $\mathbf{T}$ is the set of the list of positions $\mathbf{T}_u$ where each of the $\pi_u$ appears in $\mathcal{P}$.

### 2.1 Basic Edge-Oriented GNMs

There is a range of basic GNMs such as Erdős-Rényi Model (ER) [Erdos and Renyi, 1960], Stochastic Block Model (SBM) [Holland *et al.*, 1983], Chung-Lu Model (CL) [Chung and Lu, 2002], etc. that build $\mathcal{P}$ with some mathematical operation over a set of parameter(s). For space reasons we will define only CL here since is relevant for our later discussion.

**Chung-Lu Model (CL)** [Chung and Lu, 2002]. In a CL model, $\mathcal{P}_{ij}=w_iw_j/\sum w_k$ for a sequence of expected node-degrees $\mathbf{w}=(w_1,\ldots,w_{|\mathbf{V}|})$. There are then, for the case of undirected networks, at most $|\mathbf{V}|(|\mathbf{V}|-1)/2$ probabilities $\pi_i$ in $\mathcal{P}$ (and associated $\mathbf{U}$), in which case $\mathbf{T}$ would consist of lists of a single element each.

### 2.2 Hierarchical GNMs

Hierarchical GNMs are a super-classs of GNMs that use a hierarchical process to introduce dependencies among sets of edge RVs. While a basic GNM defines the matrix $\mathcal{P}$ directly, a hierarchical GNM uses intermediate latent variables at a certain level in the hierarchy to model the variable interactions of the next level in the hierarchy, modeling the entries in $\mathcal{P}$ indirectly. Thus, there are two levels of randomness associated with sampling in hierarchical GNMs: one for the intermediate (hierarchical) RVs that impact the generation of $\mathcal{P}$, and one for edge sampling. Notice that basic GNMs can be regarded as hierarchical GNMs with a single layer. Most hierarchical GNMs are defined iteratively or procedurally, but not all iterative or procedural GNMs are hierarchical. We denote the levels in the hierarchy by $\lambda\in\{0,1,\ldots,\phi-1\}$, where $\phi$ is the number of levels in the hierarchy, $\lambda=0$ represents the root of the hierarchy, and $\lambda=\phi-1$ the edge layer. $B^{[\lambda]}$ is the adjacency matrix associated with the blocks at layer $\lambda$.

**Block two-level Erdős-Rényi Model (BTER)** [Seshadhri *et al.*, 2012]. In a preprocessing step, BTER creates groups of nodes based on the sequence of expected degrees. Then, sampling is done in two steps: (1) An ER model is used to link nodes *within* groups, where the probability is proportional to the smallest node-degree in the group; (2) A CL model is used to create link *between* groups, with probability of edges is proportional to the excess degree from (1). Step 1 ensures that edges within-blocks have higher probability than between-blocks. Step 2 ensures a power law degree-distribution and since it uses a CL model, $\mathbf{U}$ & $\mathbf{T}$ have the same worst case scenario as CL.

**Mixed Kronecker Product Graph Model (mKPGM)** [Moreno *et al.*, 2013]. mKPGM is a GNM that overcomes known limitations (see e.g. [Seshadhri *et al.*, 2013]) of stochastic Kronecker graphs [Leskovec *et al.*, 2010]. Given a $b\times b$ parameter matrix $\Theta$, $K$ and $\ell$, it samples a network as follows.

1. Compute $\mathcal{P}^{[0]}$, a $b^\ell\times b^\ell$ matrix, equal to $\ell-1$ Kronecker products of $\Theta$ with itself.
2. Construct $G^{[0]}$from $\mathcal{P}^{[0]}$ by sampling each edge RV independently $\sim Bernoulli(\mathcal{P}^{[0]}_{ij})$.
3. For $l=1\ldots K-\ell$: Set $\mathcal{P}^{[l]}=G^{[l-1]}\otimes\Theta$ and sample $G^{[l]}$.

Note, $|\mathbf{V}^{[0]}|=b^\ell$ and $|\mathbf{V}^{[K-\ell]}|=b^K$. Since $G^{[0]}\ldots G^{[K-\ell-1]}$ represent auxiliary graphs, where each edge influences a *block* of possible edges in the next iteration of the hierarchical sampling process (and they are not present in the final output network); we refer to them as block adjacency matrices: $B^{[0]}\ldots B^{[K-\ell-1]}$. To simplify notation we will refer to the final sampled network $G^{[K-\ell]}=(\mathbf{V}^{[K-\ell]},\mathbf{E}^{[K-\ell]})$ as $G=(\mathbf{V},\mathbf{E})$. In mKPGMs, the size of $\mathbf{U}$ is $b^2$ for each level $l>0$ of the hierarchy and $\mathbf{T}$ consists of $b^2$ lists of edges.

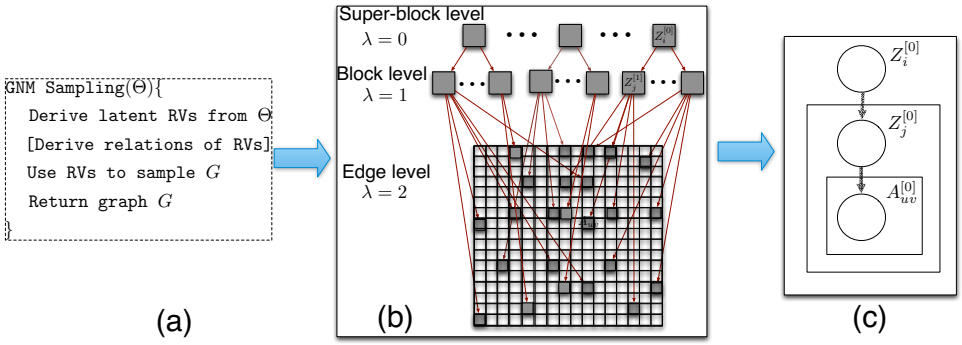


Figure 1: Example transformation from an original GNM defined procedurally (a) to BN form (b) and its corresponding plate notation (c).

## 3 Representing GNMs as Bayesian Networks

In this section, we introduce a common BN representation for GNMs with hierarchical dependencies among the edge-probabilities, and develop an associated, efficient sampling method. We also identify common properties of the GNMs based on their BN form and exploit these properties for efficient and provably correct sampling.

The *BN representation* is comprised of a hierarchical structure of edges and blocks. We denote the levels in the hierarchy by $\lambda \in \{0, 1, \ldots, \phi - 1\}$, where $\phi$ is the number of levels in the hierarchy, $\lambda = 0$ represents the root of the hierarchy, and $\lambda = \phi - 1$ the edge layer. A block $j$ at level $\lambda$ has an associated RV $Z_j^{[\lambda]}$ which represents its state: sampled ($Z_j^{[\lambda]} = 1$) or not ($Z_j^{[\lambda]} = 0$). Since blocks could have different sizes we keep a one dimensional subindex for each block RV ($Z_i$) rather than use the rectangular structure (with two subindices $Z_{ij}$) of some GNMs (e.g., mKPGM in Fig. 2). Likewise, $\mathcal{P}$ could be regarded as a one-dimensional structure for this same purpose (notice that $P(B_j^{[\lambda]} = 1) = \mathcal{P}_j^{[\lambda]}$ is the probability of sampling block $j$ as defined by the GNM). This illustrates that our algorithm is easily applicable to hierarchical models with complex topology of the interacting layers and not just to tree-like BNs. The function $pa\left(Z_j^{[\lambda]}\right) := Z_i^{[\lambda-1]}$ for $\lambda > 0$ returns the parent RVs for $Z_j^{[\lambda]}$. We denote as $\mathbf{Z}^{[\lambda]}$ the set $\{Z_j^{[\lambda]}\}$. It is important to notice that the BN is comprised of $\mathbf{Z}^{[0]}, \ldots, \mathbf{Z}^{[\phi-1]}$ and that the parameters of the BN are the associated probabilities. These probabilities are a function of, or derived from, $\mathcal{P}^{[\lambda]}$. However, this relation is not straightforward for all GNMs.

### 3.1 Transformation of GNMs to BNs

As discussed in the previous section, hierarchical GNMs are a superclass of GNMs that include edge-based GNMs as a special case, as well as more complex models that use a hierarchy of latent variables to represent dependencies among edge RVs. However, since most hierarchical GNMs are defined iteratively, the hierarchical structure is not immediately evident and a transformation is necessary. In particular for procedural models, transforming a GNM to BN form consists of two steps, as outlined in Figure 1: (1) Restructure the output of each step in the iterative sampling of a GNM as a set of RVs. This allows the process to be reorganized as levels in a hierarchy of RVs. (2) Use the hierarchy of RVs to build a BN using parameters for the BN derived from the GNM. Some care should be exercised for this transformation as the relations among RVs are not trivial.

**General Transformation from Hierarchical GNM to BN**
A hierarchical model $\mathcal{M}$, with parameter matrix $\Theta$, can be represented as a BN $\mathcal{N}$ with parameters $\Theta'$ obtained from $\Theta$: $\mathcal{M}_\Theta \vec{r} \mathcal{N}_{\Theta'}$ as follows:

1. Represent as $Z_j^{[0]}$ the RV that models the block edge $B_j^{[0]}$
2. Define the probability of $Z_j^{[0]}$: $P(Z_j^{[0]} = 1) = \mathcal{P}^{[0]}[j]$
3. For $\lambda = 1, \ldots, \phi - 1$
   (a) Represent as $Z_k^{[\lambda]}$ the RV that models the block $B_k^{[\lambda]}$
   (b) Specify $pa\left(Z_k^{[\lambda]}\right) = Z_{k'}^{[\lambda-1]}$ to be the corresponding parent of RV $Z_k^{[\lambda]}$, then, define the CPD of $Z_k^{[\lambda]}$:
   $$P\left(Z_k^{[\lambda]} = 1 | pa\left(Z_k^{[\lambda]}\right) = 1\right) = \mathcal{P}^{[\lambda]}[k]$$
   $$P\left(Z_k^{[\lambda]} = 1 | pa\left(Z_k^{[\lambda]}\right) = 0\right) = 0$$
4. Add the RVs $\mathbf{Z}^{[0]}, \ldots, \mathbf{Z}^{[\phi-1]}$ (from 1 & 3.a) to the BN $\mathcal{N}$ and add their associated CPDs (from 2 & 3.b) to $\Theta'$.

Notice that when $\phi$ (number of levels) is 1 then the hierarchical GNM collapses to a traditional edge-based GNM. Thus, hierarchical GNMs include edge-based GNMs as a special case. Finally, $\Theta'$ is fully defined by $\Theta$ because all $\mathcal{P}$ are derived from it. We now discuss two specific transformations.

**Transforming mKPGM to BN representation**

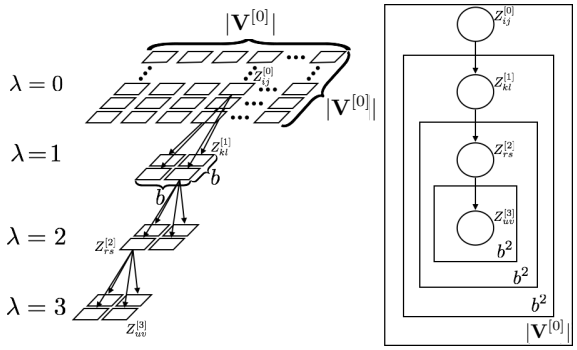


Figure 2: Example mKPGM in BN form. The left subplot, descendants for 1 RV per level show the relations among the variables. The right subplot shows the BN using plate notation.

An mKPGM model $\mathcal{M}$ with parameter matrix $\Theta$, $K$, and $\ell$ (Sec. 2.2) can be represented as a BN $\mathcal{N}$ with a tree structure and parameters $\Theta'$ obtained from $\Theta$: $\mathcal{M}_\Theta \vec{r} \mathcal{N}_{\Theta'}$ as follows.

1. Represent as $Z_{ij}^{[0]}$ the RV that models the block edge $B_{ij}^{[0]}$ $\forall i,j \in [1, b^\ell]$
2. Define the CPD of $Z_{ij}^{[0]}$: $P(Z_{ij}^{[0]}=1) = \mathcal{P}^{[0]}[i,j]$ for all $i,j$, where $\mathcal{P}^{[0]} = \otimes_{k=1}^{\ell}\Theta$
3. For $\lambda = 1,\ldots,\phi$, where $\phi = K-\ell$ do:
   (a) Represent as $Z_{kl}^{[\lambda]}$ the RV that models the block edge $B_{kl}^{[\lambda]}$ $\forall k,l \in [1, b^{\ell+\lambda}]$
   (b) Specify $pa\left(Z_{kl}^{[\lambda]}\right) = Z_{\lceil k/b\rceil,\lceil l/b\rceil}^{[\lambda-1]}$ to be the parents of RV $Z_{kl}^{[\lambda]}$, where $x = \text{mod}\,(k-1,b)+1$ and $y = \text{mod}\,(l-1,b)+1$, then, define the CPD of $Z_{kl}^{[\lambda]}$, $\forall k,l \in [1, b^{\ell+\lambda}]$:
$$P\left(Z_{kl}^{[\lambda]}=1 | pa\left(Z_{kl}^{[\lambda]}\right)=1\right) = \Theta[x,y]$$
$$P\left(Z_{kl}^{[\lambda]}=1 | pa\left(Z_{kl}^{[\lambda]}\right)=0\right) = 0$$
4. Add the RVs $\mathbf{Z}^{[0]}, \mathbf{Z}^{[1]}, ..., \mathbf{Z}^{[\lambda=K-\ell]}$ (from steps 1 and 3.a) to the BN $\mathcal{N}$ and add their associated CPDs (from steps 2 and 3.b) to $\Theta'$.

An example BN representation of an mKPGM is shown in Fig. 2 for $\lambda = 0,1,2,3$. Here $\lambda = 0$ corresponds to $B^{[0]}$ in the mKPGM sampling process. There is a total of $(b^\ell)^2 = |\mathbf{V}^{[0]}|^2$ RVs each of them represented by a $Z_{ij}^{[0]}$ (a double subindex is used to indicate the position of the RV in the block/edge matrix). Each of the RVs has $b^2$ descendants at $\lambda > 0$.

**Transforming BTER to BN representation**

A BTER model $\mathcal{M}$ with parameters $\Theta = (\{n_d\}_{d\in\mathbb{N}}, \{c_d\}_{d\in\mathbb{N}})$, where $\{n_d\}_{d\in\mathbb{N}}$ is a sequence of target degrees and $\{c_d\}_{d\in\mathbb{N}}$ is the target clustering coefficient per degree, can be represented as a BN $\mathcal{N}$ with a tree structure and parameters $\Theta'$ obtained from $\Theta$: $\mathcal{M}_{\Theta} \overrightarrow{r} \mathcal{N}_{\Theta'}$ as follows.

As discussed earlier, BTER creates groups of nodes with possibly the same degree and uses an ER model to generate blocks of edges linking those nodes together. These are called *affinity*-blocks. To produce a power law degree distribution, there are many affinity-blocks of edges linking nodes with small degree and fewer blocks for nodes with larger degree. Recall that each node has an expected degree; then, *bulk nodes* are nodes where the minimum degree of its affinity block is equal to its expected degree. Otherwise, if the minimum degree is smaller, they are called *filler nodes*. Contrary to mKPGMs, BTER has an heterogenous number of descendent edges. The nodes of degree one have no edges between them because they are used as *filler* nodes; i.e. they are used to link blocks of degrees $> 1$.

Let $n_b, r_d, d_b$ be the number of edges in affinity block $\mathbf{B}_{ij}^{[0]}$, the ratio of the number of filler nodes to the total nodes, and the minimum degree in the block, respectively. Let, $Bin^{-1}$ be the inverse CDF of a binomial. Then, the mapping of BTER sampling to BN representation is as follows:

1. Represent as $Z_{ij}^{[0]}$ the RV modeling the affinity blocks $\mathbf{B}_{ij}^{[0]}$
2. Obtain the parameters $n_b, r_d, d_b$ of BTER as described in Algorithm 1 of [Kolda *et al.*, 2014]
3. Define the CPD of $Z_{ij}^{[0]}$: $P(Z_{ij}^{[0]}=1) = \mathcal{P}^{[0]}[i,j]$ $\forall$ $i,j$ where $P^{[0]}$ is probability of sampling edges from degree equal to the degree of $B_{ij}^{[0]}$
4. Let $pa(\cdot)$ be the affinity block containing $\cdot$. Define $P(Z_{kl}^{[\phi]}=1 | pa(Z_{kl}^{[\phi]})=0) = 0 \quad \forall k,l \in [1,\ldots,|\mathbf{V}|]$. Otherwise, for edges $\mathbf{e}_{k,l}$ s.t. $pa(Z_{kl}^{[\phi]}) = 1$ and $b$ its affinity block with clustering coefficient $c_{d_b}$
   (a) Represent its associated RV as $Z_{kl}^{[\phi]}$
   (b) If $k = l$: Define the CPD of $Z_{kl}^{[\phi]}$: $P(Z_{kl}^{[\phi]}=1 | pa(Z_{kl}^{[\lambda]})=1) = Bin^{-1}\left(c_{d_b}^{1/3}\begin{pmatrix} n_b \\ 2 \end{pmatrix}\right)$
   (c) Else: Given $\sigma_{type} = \frac{r_d}{2}$ if *type* = *filler*, and $\sigma_{type} = \frac{1-r_d}{2}$ if *type* = *bulk*, define the CPD of $Z_{kl}^{[\phi]}$: $P(Z_{kl}^{[\phi]}=1 | pa(Z_{kl}^{[\lambda]})=1) = Bin^{-1}\left(\sum_{i\in V_d}\sigma_{type_i}[d_i - (c_{d_b}^{1/3}\cdot d_b)]\right)$ for all $k,l$ and nodes $i \in V_d$ of degree $d$
5. Add the RVs $\mathbf{Z}^{[0]}, \mathbf{Z}^{[1]}$ to the BN $\mathcal{N}$(steps 1, 5.a.i, 5.b) and their associated CPDs (steps 3, 4, 5.a.2, & 5.c) to $\Theta'$.

As we can see BTER and mKPGM are similar in the sense that both models constrain the sampling so very few edges are sampled with higher probabilities. This allows for skewed degree distribution. The difference in the depth of the hierarchies is now easily seen, mKPGM groups edges in blocks and super-blocks to abstract their relations as separate latent SBM models, while BTER uses shallower hierarchies for redistribution of probabilities according to a power-law.

## 3.2 GNM-BN Common Properties

We identify 2 properties of GNMs transformed to BN form.

**1. Group Symmetries**—The first key insight about our BN representation is that symmetries appear in the BN from the parameterization of the GNM. Since there are fewer than $|\mathbf{V}|^2$ unique $\Theta$s, RVs can be grouped and sampled together. Moreover, since the models are designed to generate sparse networks, it is more efficient to sample the number of edges and then select their locations rather than sequentially sample each edge RV, as proved in [Moreno *et al.*, 2014]. As in earlier lifted techniques, that were based on finding identical structures for which a common computation was performed only once ([Koller and Pfeffer, 1997; Pfeffer *et al.*, 1999]), we propose an algorithm that identifies RVs with the same conditionals and group them together in a single representation. Then, sampling will be done only once by using a Binomial (this determines the number of edges to sample) and randomly selecting the edges from the locations with the same unique probability (as illustrated in Fig. 3). Because in GNMs the symmetries are parametric similarities among a set of independent RVs there is no need to assess the dependencies among individual RVs or among groups of RVs.

**2. DCSD Sparsification**—The second property of the BN representation for GMNs is *deterministic context-specific dependence* (DCSD). As seen in the earlier transformations, this property arises because some values of the variables are sampled with value 0. Since sampling at the next layer of the hierarchy is conditional on the sampling of the previous layer, the sampling space becomes more and more sparse as we move through the hierarchy, as shown in Figure 4. Because a

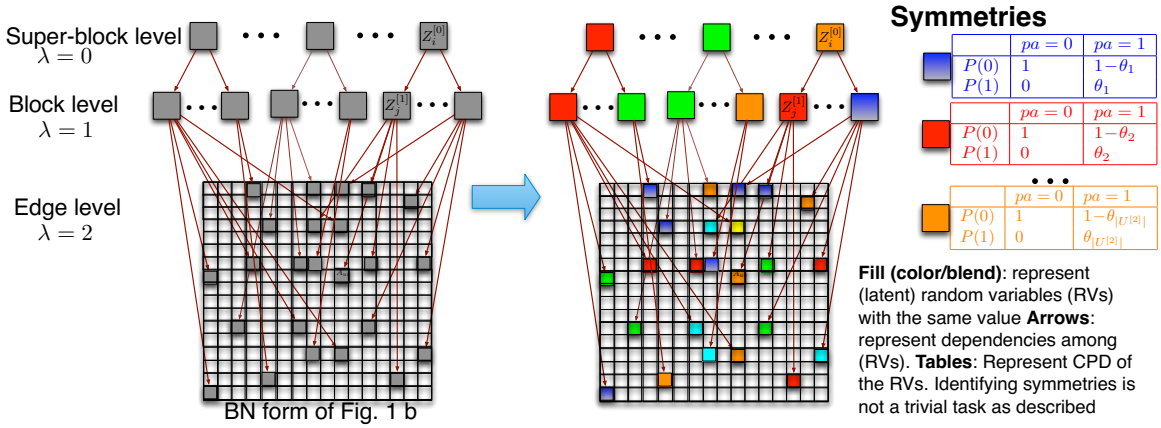


Figure 3: Sampling of GNM-BN: Symmetries of RV values are exploited for efficiency and correctness.

value in any given $\mathbf{U}^{[\lambda]}$ may be 0, we don't need to sample at all for the associated $T$. Thus this sparsification can be used for more efficient sampling. Formally:

**Definition 2.** ***Deterministic context-specific dependence (DCSD):*** *Let $\mathcal{N}$ be a BN that generates a network $G$ through a hierarchical sampling process. Let $P\left(Z_j^{[\lambda]}\right)$ be the probability that a block is sampled. Then $\mathcal{N}$ is "DCSD"* iff *it partitions all $Z_j$, such that:*

$$P\left(Z_j^{[\lambda]} = 1 \left| pa\left(Z_j^{[\lambda]}\right) = 0\right.\right) = 0 \quad \forall\, j, \lambda$$

*where $P\left(Z_j^{[\lambda]}=1|pa\left(Z_j^{[\lambda]}\right)=1\right)>0\ \forall\, j, \lambda$, at each layer $\lambda$.*

**Remark 1.** *$Z_j^{[\lambda]}$ is sampled iff its parent is sampled with value 1. However, if a superblock is not sampled, sampling of subblocks or edges is inhibited. Note that while $P(E_{uv}|pa(E_{uv})=0)=0$, the marginal $P(E_{uv})>0$.*

DCSD is related to *context-specific independence* (independence in a BN due to specific realizations of RVs) [Boutilier *et al.*, 1996], but in this case the specific context precludes further sampling of subblocks and edges. We show below how these two properties can be used to develop an *efficient and provably correct* general sampling algorithm for the GNM-BNs.

### 3.3 GNM-BN Sampling

**RV Reorganization:** As we described before, a GNM can be represented as a set of unique probabilities $\pi_u \in \mathbf{U}$ and their positions $\mathbf{T}_u$. We can apply this organization to the RVs $\mathbf{Z}^{[0]}, ..., \mathbf{Z}^{[\lambda=\phi-1]}$ (and their probabilities) in $\mathcal{N}$ from the transformations $\mathcal{M}_{\vec{r}}\mathcal{N}$ explained before. Thus, $\mathcal{N}$ can be represented by the set of ordered pairs $\{(\mathbf{U}^{[\lambda]}, \mathbf{T}^{[\lambda]})\}_{\lambda=0}^{\phi-1}$, where $\mathbf{U}^{[\lambda]}=\{\pi_1, ..., \pi_u, \ldots\}=\texttt{unique}(P(Z^{[\lambda]}))$ and $\mathbf{T}^{[\lambda]}$ is a vector with the list of positions $\mathbf{T}_u^{[\lambda]}$ where each of the probabilities $\pi_u$ appear at level $\lambda$.

**Algorithm:** We propose an algorithm for sampling from models trasnformed to BN form that is efficient (as proved below). Our algorithm is easily applicable to hierarchical models with complex topology. The input of the algorithm is the BN $\mathcal{N}$ and its parameters. $\mathcal{N}$ has an associated set of unique probabilities $\mathbf{U}^{[\lambda]}$ and list of positions $\mathbf{T}^{[\lambda]}$ as we described above.

**Algorithm** *GNM-BN-Sampling*

1: **Input:** $\mathcal{N}$
2: **Ouput:** $G = (\mathbf{V}, \mathbf{E})$ {defined by its adjacency matrix $\mathbf{A}$}
3: $\mathbf{V} = \{1, \ldots, |\mathbf{V}|\}$
4: Obtain the set $\mathbf{U}^{[0]}$ and the list $\mathbf{T}^{[0]}$ using $\mathcal{N}$
5: **for** $\lambda = 0, \ldots, \phi - 1$ **do**
6: **for** $i = 1, \ldots, |\mathbf{U}^{[\lambda]}|$ **do**
7: Take $\pi_i = \mathbf{U}_i^{[\lambda]}$ {unique prob in $\mathcal{P}^{[\lambda]}$}
8: Take $\tau_i = |\mathbf{T}_i^{[\lambda]}|$ {# of positions in $\mathcal{P}^{[\lambda]}$ where $\pi_i$ appears}
9: Sample $N_{b_i} \sim Bin(\tau_i, \pi_i)$
10: $\overrightarrow{\mathbf{pos}}$ = Random $N_{b_i}$ positions from $\mathbf{T}_i^{[\lambda]}$
11: **if** $\lambda < \phi - 1$ **then**
12: Set $\mathbf{Z}_j^{[\lambda]} = 1$, for all $j \in \overrightarrow{\mathbf{pos}}$
13: Obtain the set $\mathbf{U}^{[\lambda+1]}$ and the list $\mathbf{T}^{[\lambda+1]}$ using $\mathbf{Z}^{[\lambda]}$
14: **else**
15: Set $\overrightarrow{\mathbf{u}} = \lceil \frac{\overrightarrow{\mathbf{pos}}}{|\mathbf{V}|} \rceil$ and $\overrightarrow{\mathbf{v}} = \text{mod}(\overrightarrow{\mathbf{pos}} + |\mathbf{V}| - 1, |\mathbf{V}|) + 1$
16: Set $\mathbf{A}_{u,v} = 1$ where $u = \overrightarrow{\mathbf{u}}_w$, $v = \overrightarrow{\mathbf{v}}_w$ for $w = 1, \ldots, N_{b_i}$

Line 4 obtains the set of unique probabilities $\mathbf{U}^{[0]}$ and the list $\mathbf{T}^{[0]}$ (i.e. the corresponding positions where those probabilities appear). This information is used for sampling at the root layer $\lambda = 0$. The algorithm consists of using each $\pi_i \in \mathbf{U}^{[\lambda]}$ and the corresponding list of positions for randomly sampling blocks. Instead of sampling from a Bernoulli for each position of the block, we sample from a binomial which models the distribution of successes in $|\mathbf{T}_i^{[\lambda]}|$ Bernoulli trials with probability $\pi_i$. Sampling from this distribution gives the number of sampled blocks ($N_{b_i}$) for layer $\lambda$ whose positions are randomly picked and stored in the vector $\overrightarrow{\mathbf{pos}}$. This vector is used to sample the blocks, i.e. to set their RVs $\mathbf{Z}_j^{[\lambda]}$ to one, where $j \in \overrightarrow{\mathbf{pos}}$ (lines 11-13). Sampling of edges (lines 15-16) follows the same idea except the positions $\overrightarrow{\mathbf{pos}}$ are transformed to two indices: the row and column, $u \in \overrightarrow{\mathbf{u}}$, and $v \in \overrightarrow{\mathbf{v}}$ of the adjacency matrix.

**Complexity:** The average-case complexity of the algorithm is $O(\phi|E|)$, where $\phi$ is the depth of the hierarchy (typically a constant) and $|E| = \mathbb{E}[|\mathbf{E}|]$. This can be proved as follows. Lines 9-10 have average-case complexity $O(\mathbb{E}[|\mathbf{Z}^{[\lambda-1]}|])$. This complexity is $O(\tau_i \pi_i)$ ($\tau_i$=# of positions in $\mathcal{P}^{[\lambda]}$ where $\pi_i$ appears). The average complexity of the loop for ($\lambda<\phi-1$)

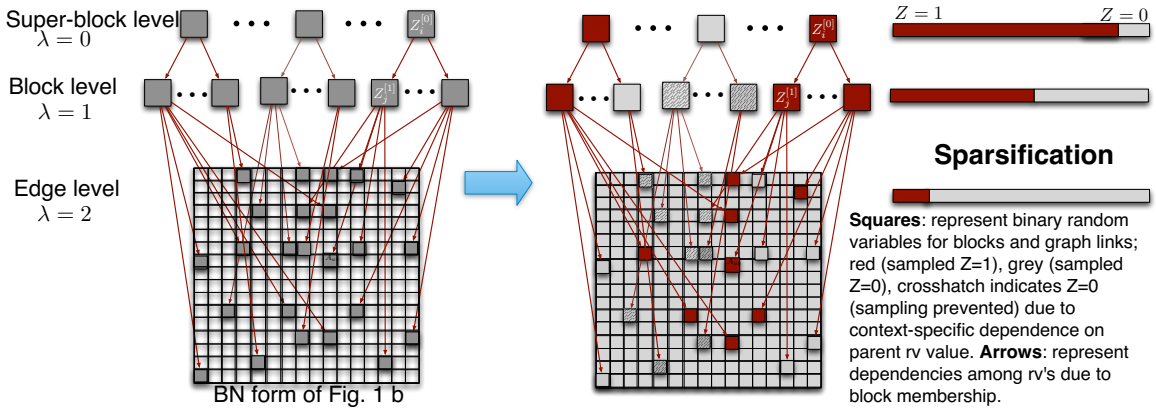


Figure 4: Sampling of GNM-BN: context-specific dependence occurs when RVs $Z$ condition sampling of blocks/edges.

is approximately $O(\mathbb{E}[\sum_{\lambda=0}^{\phi-1}|\mathbf{Z}^{[\lambda]}|])=O(\phi|E|)$ ($\phi$ =# layers) Since the average-case complexity for $\lambda=\phi-1$ is $O(|E|)$.

**Scalability:** Because of the linear (in the number of edges) average case complexity of the algorithm, a BN-GNM is scalable for sufficiently sparse networks. Additionally, a BN-GNM is amenable to be parallelized. After symmetries are identified, sampling could be separated in independent tasks due to the independence of the parameters.

**Lemma 1.** *Given a valid model in BN representation $\mathcal{N}_{\Theta'}$ with probability $P_{\mathcal{N}}$, and the algorithm GNM-BN-Sampling with probability $P_{\mathcal{S}}$, then $\forall G^{[\lambda]}$ and $0<\lambda\leq\phi-1$:*

$$P_{\mathcal{N}}(e_{k,l}\in G^{[\lambda]}|G^{[\lambda-1]})=P_{\mathcal{S}}(Z_{kl}^{[\lambda]}=1|pa(Z_{kl}^{[\lambda]}))$$

*Proof.* Let $\gamma(k,l)$ be the indices of the block containing $e_{k,l}$, and $\pi_{\sigma_{kl}}$ the unique probability associated with $Z_{kl}^{[\lambda]}$. Then, $P_{\mathcal{N}}(e_{k,l}\in G^{[\lambda]}|G^{[\lambda-1]})=\begin{cases}P^{[\lambda]}[k,l] & \text{if } E_{\gamma(k,l)}\in G^{[\lambda-1]}\\ 0 & \text{if } E_{\gamma(k,l)}\notin G^{[\lambda-1]}\end{cases}$. Likewise, $P_{\mathcal{S}}(Z_{kl}^{[\lambda]}=1|pa(Z_{kl}^{[\lambda]}))=\begin{cases}\pi_{\sigma_{kl}}=\mathcal{P}^{[\lambda]}[k,l] & \text{if } pa(Z_{kl}^{[\lambda]})=1\\ 0 & \text{if } pa(Z_{kl}^{[\lambda]})=0\end{cases}$.
Since $E_{\gamma(k,l)}\in G^{[\lambda-1]}\iff pa(Z_{kl}^{[\lambda]})=1$, then $P_{\mathcal{N}}(e_{k,l}\in G^{[\lambda]}|G^{[\lambda-1]})=P_{\mathcal{S}}(Z_{kl}^{[\lambda]}=1|pa(Z_{kl}^{[\lambda]}))$. □

**Lemma 2.** *Given a valid model in BN representation $\mathcal{N}_{\Theta'}$ with probability $P_{\mathcal{N}}$, and the algorithm GNM-BN-Sampling with probability $P_{\mathcal{S}}$, then $\forall G^{[\lambda]}$ and $0<\lambda\leq\phi-1$:*

$$P_{\mathcal{N}}(G^{[\lambda]}|G^{[\lambda-1]})=P_{\mathcal{S}}(\mathbf{Z}^{[\lambda]}|pa(\mathbf{Z}^{[\lambda]}))$$

*Proof.* By lemma 1 and sampling independence of $\mathbf{Z}^{[\lambda]}$ then $P_{\mathcal{S}}(\mathbf{Z}^{[\lambda]}|pa(\mathbf{Z}^{[\lambda]}))=\prod_{\forall k,l}P_{\mathcal{S}}(Z_{kl}^{[\lambda]}=1|pa(Z_{kl}^{[\lambda]}))=\prod_{\forall k,l}P_{\mathcal{N}}(e_{k,l}\in G^{[\lambda]}|G^{[\lambda-1]})=P_{\mathcal{N}}(G^{[\lambda]}|G^{[\lambda-1]})$ □

**Theorem 1** (Correctness)**.** *Given a valid model in BN representation $\mathcal{N}_{\Theta'}$ with probability $P_{\mathcal{N}}$, and the algorithm GNM-BN-Sampling with probability $P_{\mathcal{S}}$, then $\forall G\ P_{\mathcal{N}}(G)=P_{\mathcal{S}}(G)$.*

*Proof.* $P_{\mathcal{N}}(G)=\sum_{G^{[\lambda-1]}\in\Gamma^{[\lambda-1]}}P_{\mathcal{N}}(G^{[\lambda]}|G^{[\lambda-1]})P_{\mathcal{N}}(G^{[\lambda-1]})$ for the set $\Gamma$ of all possible graphs at $\lambda-1$. By Lemma 2 $P_{\mathcal{N}}(G)=\sum_{G^{[\lambda-1]}\in\Gamma^{[\lambda-1]},Z^{[\lambda]}}P_{\mathcal{S}}(\mathbf{Z}^{[\lambda]}|pa(\mathbf{Z}^{[\lambda]}))P_{\mathcal{N}}(G^{[\lambda-1]})$
By the same procedure, it can be proven recursively that: $P_{\mathcal{N}}(G^{[1]})=\sum_{G^{[0]}\in\Gamma'^{[0]},Z^{[1]}}P_{\mathcal{S}}(\mathbf{Z}^{[1]}|pa(\mathbf{Z}^{[1]}))P_{\mathcal{N}}(G^{[0]})$ for the set $\Gamma'$ of all possible graphs at level 0. Since $P_{\mathcal{S}}(pa(\mathbf{Z}^{[0]}))=1$, by Lemma 1: $P_{\mathcal{N}}(G^{[1]})=\sum_{G^{[0]}\in\Gamma'^{[0]},Z^{[1]}}P_{\mathcal{S}}(\mathbf{Z}^{[1]}|pa(\mathbf{Z}^{[1]}))P_{\mathcal{S}}(pa(\mathbf{Z}^{[1]}))$.
Then, $\forall G\ P_{\mathcal{N}}(G)=P_{\mathcal{S}}(G)$ □

**Remark 2.** *Given a valid mKPGM model $\mathcal{M}(\Theta)$ with probability $P_{\mathcal{M}}$, there is a BN $\mathcal{N}$ such that the algorithm GNM-BN-Sampling, with sampling probability $P_{\mathcal{S}}$, samples correctly from $P_{\mathcal{M}}$, i.e. $\forall G\ P_{\mathcal{S}}(G)=P_{\mathcal{M}}(G)$.*

Let $P_{\mathcal{S}}$ be the probability of the GNM-BN sampling. Given the mapping $\mathcal{M}_{\Theta}\overrightarrow{\tau}\mathcal{N}_{\Theta'}$, as shown in the transformation of mKPGM to BN. Since the CPD are the same for the sampling and the BN representation, then, by Thm 1, $\forall G\ P_{\mathcal{M}}(G)=P_{\mathcal{N}}(G)=P_{\mathcal{S}}(G)$

**Remark 3.** *Given a valid BTER model $\mathcal{M}(\Theta)$ with probability $P_{\mathcal{M}}$, and the algorithm GNM-BN-Sampling with probability $P_{\mathcal{S}}$, then from Thm. 1 it also follows that $\forall G\ P_{\mathcal{M}}(G)=P_{\mathcal{S}}(G)$.*

## 4 Designing New GNMs: a Case Study

To take advantage of the properties of the BN representation, we propose a new GNM, namely the *BN-Graphlet* model. We will show that the new GNM can model graph characteristics that are not possible with several existing GNMs. We consider undirected networks but *BN-Graphlet* could be easily extended for directed graphs as well.

In order to take advantage of our framework, and in particular our sampling algorithm, we design a GNM that is directly interpretable as a BN hierarchy with common parameters per layer and with a generative process that exploits sampling conditioned on the previous layer. The key idea in *BN-Graphlet* is to redistribute symmetries and sparsity based on several graph patterns, to convey (1) locations where edge probability mass is concentrated in the desired graphs, (2) primitives regarding interactions among nodes in the graph, and (3) hierarchical dependencies among primitives. To do so we consider (a) the degree distribution, (b) minimal graphlets (*mingraphlets*, i.e. triangles, wedges, and edges) that encode node interactions per degree, and (c) a graph decomposition due to the mingraphlets.

The reason to use mingraphlets is that larger graphlets can be constructed via algebraic transformations of the minimal graphlets. Since each of these characteristics are dependent on the previous one, each will be encoded in a level of the BN hierarchy.

Let $\Delta \sim f_{deg}(\theta_\Delta)$ be the sequence of degrees drawn from the degree distribution $f_{deg}$. *BN-Graphlet* creates groups of $n_{\delta_i}$ nodes of degree $\delta_i$ for $\delta_i \in \Delta$. Then we can partition the adjacency matrix via the joint degree distribution, and use block $b_{ij} = B^{[0]}_{ij}$ to refer to links among nodes with degrees $\delta_i, \delta_j$ respectively. Let $\mathcal{P}^{[0]}[i,j]=0$ for blocks with no links; otherwise, $\mathcal{P}^{[0]}[i,j]=1$.

Next, the model partitions a single block in subgroups of nodes that link only triangles, only wedges, and the remainder, i.e., every block will be subdivided into nine subblocks $B^{[1]}_{..}$. To simplify notation, we use $B^{[1]}_{k=[1..3]l=[1..3]}(i,j)$ to refer to appropriate cell in $B^{[1]}$ associated with its parent $i,j$ in $B^{[0]}$. Then: $\mathcal{P}^{[1]}[k,l] =$

$$\begin{cases} p_{\triangle_{b_{ij}}} & \text{if } k=1, l=1 \text{ i.e., both nodes link only to triangles} \\ p_{\wedge_{b_{ij}}} & \text{if } k=2, l=2 \text{ i.e., both nodes link only to wedges} \\ 1 & \text{o.w.} \end{cases}$$

Here $p_{\triangle_{b_{ij}}}$ and $p_{\wedge_{b_{ij}}}$ are respectively: the probability that the two nodes incident to an edge in block $B^{[0]}_{ij}$ participate only in triangles, and the probability that the two nodes incident to an edge in block $B^{[0]}_{ij}$ participate only in wedges.

Lastly, the final adjacency matrix is constructed from submatrices of size $n_{\delta_i} \times n_{\delta_j}$ for each degree pair $\delta_i, \delta_j$. The nodes of each sub-matrix will be associated with *roles* based on their parent $B^{[1]}_{kl}(i,j)$. For 3x3 submatrices of edges $e_{m,n}$, where $m,n$ have the role *only triangles*, the off-diagonal edge probabilities in $B^{[2]}_{mn}$ will be all 1; for blocks where the nodes have the role *only wedges*, there will be two edges with probability 1s and one with probability 0. For all other edges, the probability will be $\kappa^{1/3}_{b_{ij}}$, where $\kappa_{b_{ij}}$ is the clustering coefficient centered at a node in block $B^{[0]}_{ij}$.

It is straightforward to see that the BN $\mathcal{N}$ of *BN-Graphlet* is directly derived from the blocks $B^{[\lambda]}_{ij}$ and their probabilities, contrary to mKPGM and BTER where transformations were needed. This is because we use our framework directly to build the GNM. The equivalence can be directly obtained with the following procedure:

1. For $\lambda = 0, \ldots, 2$
   (a) For $i,j = [1, rows(\mathcal{P}^{[\lambda]})], [1, cols(\mathcal{P}^{[\lambda]})]$
      i. Define $Z^{[\lambda]}_{ij}$ the RV that models the blocks $B^{[\lambda]}_{ij}$
      ii. Define $P(Z^{[\lambda]}_{ij} = 1 | pa(Z^{[\lambda]}_{ij}) = 0) = 0$
      iii. Define $P(Z^{[\lambda]}_{ij} = 1 | pa(Z^{[\lambda]}_{ij}) = 1) = \mathcal{P}^{[\lambda]}[i,j]$

**Estimation:** We need to estimate $\theta_\Delta$, the sequence $\kappa_{b_{ij}}$, and probabilities $p_{\triangle_{b_{ij}}}$ and $p_{\wedge_{b_{ij}}}$. $\theta_\Delta$ is learned as the MLE of the fitted degree distribution. $\kappa_{b_{ij}}$ is the clustering coefficient per block of the data graph. The parameters $p_{\triangle_{b_{ij}}}$ and $p_{\wedge_{b_{ij}}}$ are estimated as the MLE of the respective multinomial.

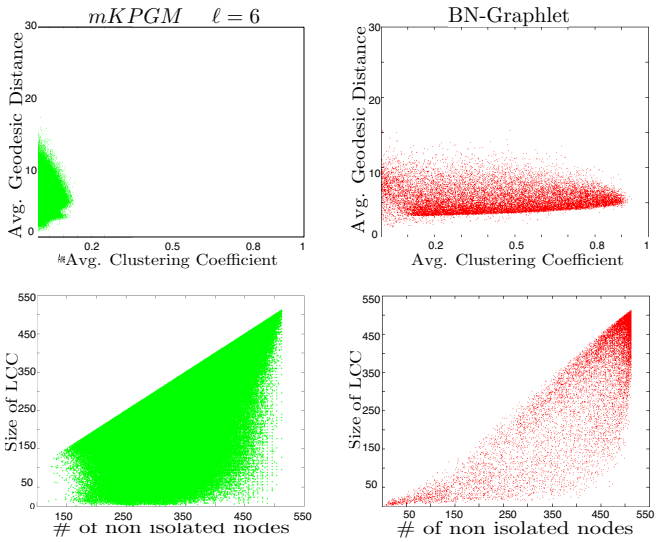


Figure 5: Variation of mKPGM and *BN-Graphlet* graph properties in synthetic networks. (LCC: largest connected component)

## 4.1 Experiments and Analysis

We performed two sets of experiments. First, we encoded *BN-Graphlet* using the new representation and show, using synthetic networks, that it can model different networks compared to mKPGM. Because mKPGM was shown in [Moreno *et al.*, 2013] to be capable of modeling a wider range of network characteristics than Chung Lu and KPGM models, in our experiments we compare to mKPGM. We generated networks for a wide range of parameter values $\Theta$ and plotted the characteristics of the sampled graphs. For mKPGM, we used 22,060 different values of $\Theta$ ($b = 2$). The parameters were generated using every possible combination of $\Theta$, such that $\theta_{11}, \theta_{12} \in \{0.01 : \delta : 1.00\}$, $\theta_{12} = \theta_{21}$, $\theta_{22} \in \{\theta_{11} : \delta : 1\}$, and $2.1 \leq \sum_{ij} \theta_{ij} \leq 2.4$ We utilized $\delta = 0.015$ and $\theta_{12} \in \{0.01 : \delta : 1\}$ to avoid repetition of the parameters due to isomorphisms of $\Theta$. For each $\Theta$ setting, we generated 75 undirected networks with mKPGM ($K=9, \ell=6$). For *BN-Graphlet* we considered $\theta_\Delta$ that lead to networks with a number of edges in the range $[800, 2650]$ for a fair comparison with mKPGM models. For $\kappa_{b_{ij}}$ we generated random values under the restriction that the global clustering coefficient is realistic. The values of $p_{\triangle_{b_{ij}}}$ and $p_{\wedge_{b_{ij}}}$ were assigned using grid search.

We evaluate the characteristics of the generated networks using: (1) average cluster coefficient, (2) average geodesic distance, (3) number of non isolated nodes, and (4) size of the largest connected component. Figure 5 reports the results. *BN-Graphlet* produces the lowest geodesic distance (small world phenomena), the highest cluster coefficient (community structure), and largely reduces the number of isolated nodes (larger connected component). Thus *BN-Graphlet* can model networks that are not easily modeled with mKPGM.

Second, we show that *BN-Graphlet* can model real world networks better than the other models. We fitted the models to three real datasets: the CoRA citations network (comprises 11,881 AI papers with 31,482 citations between them), the National Longitudinal Study of Adolescent Health (AddHealth)

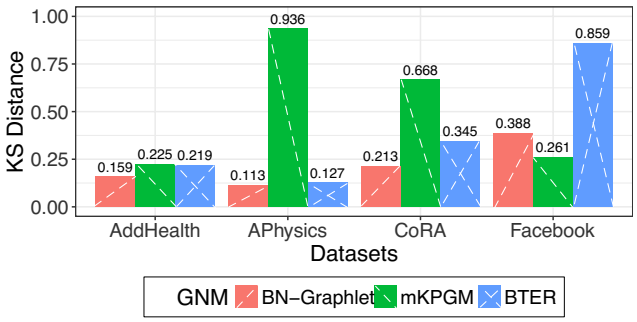


Figure 6: 3D-Kolgomorov-Smirnov distance across 4 datasets.

network (1155 nodes and 7884 edges), the astrophysics arXiv network (9,987 nodes and 25,973 edges), and Facebook wall postings (449,748 nodes and 1,016,621 edges).

We compared the *BN-Graphlet* with BTER and mKPGM. Figure 6 shows the 3-dimensional Kolmogorov-Smirnov ($KS_{3D}$) distance of the learned to the true network characteristics: hop-plot, degree, and clustering coefficient. In all datasets, *BN-Graphlet* obtains the lowest error, except in Facebook where its error is the second lowest; showing it can jointly model the hop-plot, degree, and clustering coefficient of real networks consistently better than BTER or mKPGM.

## 5 Related Work and Discussion

GNMs have been developed in various ways, using different motivations and frameworks. There has been some work to identify the common features of GNMs. For example, [Jacobs and Clauset, 2014] provides an overview of different families of GNMs and gives some important insights about relationships among them. [Rohe *et al.*, 2011] considers the relation between the latent space model [Hoff *et al.*, 2002] and stochastic block models [Holland *et al.*, 1983]. However, these works do not consider complex GNMs with a hierarchical sampling process involving intermediate latent variables.

Previous work have used BNs to create GNMs [Schein *et al.*, 2016; Liang *et al.*, 2016; Ho *et al.*, 2011; Neiswanger *et al.*, 2014]. However, these works proposed a specific BN with the purpose of solving a specific network problem (e.g., node actions, link predictions, etc). Our contribution is to propose a general representation that is not dependent on any dataset, problem, or BN topology, but rather only on an assumption of conditional independence during sampling. Using this minimal building block, we have proposed a sampling method that is universal across GNMs that fulfill this assumption.

Also related to our work is *lifted inference*. Lifted inference algorithms identify and exploit abundant symmetries in graphical models, in order to avoid repeated computations and speed up probabilistic inference [Koller and Pfeffer, 1997; Poole, 2003; Jha *et al.*, 2010; Van den Broeck and Darwiche, 2013; Mittal *et al.*, 2015; Sen *et al.*, 2009]. Lifted Importance Sampling (LIS) was proposed in relational learning for probabilistic inference [Gogate and Domingos, 2011], and extended by [Gogate *et al.*, 2012]. We note that their task is different since the symmetries are exploited to improve the precision of LIS (i.e., reduce variance) while our work exploits parametric symmetries for efficient network sampling, which provides guaranteed performance (both time complexity and correctness) for GNMs. [Venugopal and Gogate, 2014] showed a way to deal with situations where symmetries are broken. As in their work, we show ways to find symmetries but in the realm of social network models.

In general, like these existing lifted-techniques, our *lifted sampling* identifies structural patterns in GNMs to sample a set of random variables as a group rather than treating each random variable separately. This leads to an efficient and correct sampling method. Unlike existing lifted-techniques, we use symmetries to create a meta-model that encodes existing social network models and allows for creation of new ones.

In our work, we take advantage of the symmetries in the intrinsic graphical model of the GNMs for efficient sampling and then exploit the hierarchical representation to construct a generative model that produces sampled graphs with characteristics that are not easily achieved with current GNM methods. In the experiments, we show the range of networks that can be modeled with this new generative model. To our knowledge, our work is one of the first applications of lifted techniques that exploits the symmetries existing in social network models.

## 6 Conclusion

In this paper, we showed how hierarchical GNMs that are specified procedurally, can be represented as generally as BNs. Our unifying BN view provides the following advantages: (1) *universal representation*, which can highlight the similarities and differences between GNMs, and (2) *efficient*, provably correct and universal sampling, due to the adherence of the algorithm to a directed BN structure and an efficient sampling algorithm. The representation also facilitates the creation of new GNMs. To illustrate the benefits of the BN representation and associated sampling method, we proposed the hierarchical *BN-Graphlet* model and showed it more accurately captures the structure of four real-world networks.

## Acknowledgements

We thank the anonymous reviewers for their useful comments. This research is supported by NSF under contract numbers: IIS-1546488, IIS-1618690, CCF-0939370, and by "CONICYT + PAI/Concurso nacional de apoyo al retorno de investigadores/as desde el extranjero, convocatoria 2014 + folio 82140043."

## References

[Benson *et al.*, 2014] Austin R. Benson, Carlos Riquelme, and Sven Schmit. Learning multifractal structure in large networks. In *Proceedings of the 20th ACM SIGKDD International Conference on KDD*, 2014.

[Boutilier *et al.*, 1996] Craig Boutilier, NirFriedman, Moises Goldszmidt, and Daphne Koller. Context-specific independence in bayesian networks. In *Proceedings of the 12th International Conference on UAI*, pages 115–123, 1996.

[Chung and Lu, 2002] F. Chung and L. Lu. The average distances in random graphs with given expected degrees. *PNAS*, 99(25):15879–15882, 2002.

[Erdos and Renyi, 1960] P. Erdos and A. Renyi. On the evolution of random graphs. In *Publication of the Mathematical Institute of the Hungarian Academy of Sciences*, pages 17–61, 1960.

[Gogate and Domingos, 2011] Vibhav Gogate and Pedro M. Domingos. Probabilistic theorem proving. *CoRR*, abs/1202.3724, 2011.

[Gogate *et al.*, 2012] Vibhav Gogate, Abhay Jha, and Deepak Venugopal. Advances in lifted importance sampling. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, pages 1910–1916, 2012.

[Ho *et al.*, 2011] Qirong Ho, Ankur P. Parikh, Le Song, and Eric P. Xing. Multiscale community blockmodel for network exploration. In *AISTATS*, 2011.

[Hoff *et al.*, 2002] P. Hoff, A. Raftery, and M. Handcock. Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97:1090–1098, 2002.

[Hoff, 2008] Peter D. Hoff. Multiplicative latent factor models for description and prediction of social networks. *Computational and Mathematical Organization Theory*, 15(4), 2008.

[Holland *et al.*, 1983] Paul W. Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 5(2):109 – 137, 1983.

[Jacobs and Clauset, 2014] Abigail Z. Jacobs and Aaron Clauset. A unified view of generative models for networks: models, methods, opportunities, and challenges. In *arXiv stat*, 2014.

[Jha *et al.*, 2010] Abhay Jha, Vibhav Gogate, Alexandra Meliou, and Dan Suciu. Lifted inference seen from the other side : The tractable features. In *Advances in NIPS*. 2010.

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

[Koller and Pfeffer, 1997] Daphne Koller and Avi Pfeffer. Object-oriented bayesian networks. In *Proceedings of the Thirteenth Conference on UAI*, UAI'97, pages 302–313, San Francisco, CA, USA, 1997.

[Larremore *et al.*, 2014] Daniel B. Larremore, Aaron Clauset, and Abigail Z. Jacobs. Efficiently inferring community structure in bipartite networks. *Phys. Rev. E*, 90:012805, Jul 2014.

[Leskovec *et al.*, 2010] Jure Leskovec, Deepayan Chakrabarti, Jon Kleinberg, Christos Faloutsos, and Zoubin Ghahramani. Kronecker graphs: An approach to modeling networks. *JMLR*, 11(Feb):985–1042, 2010.

[Liang *et al.*, 2016] Dawen Liang, Laurent Charlin, James McInerney, and David M. Blei. Modeling user exposure in recommendation. In *Proceedings of the 25th International Conference on World Wide Web*, 2016.

[Mittal *et al.*, 2015] Happy Mittal, Anuj Mahajan, Vibhav G Gogate, and Parag Singla. Lifted inference rules with constraints. In *Advances in NIPS*. 2015.

[Moreno *et al.*, 2013] S. Moreno, J. Neville, and S. Kirshner. Learning mixed Kronecker product graph models with simulated method of moments. In *19th ACM SIGKDD International Conference on KDD*, 2013.

[Moreno *et al.*, 2014] S. Moreno, J. Pfeiffer III, S. Kirshner, and J. Neville. A scalable method for exact sampling from Kronecker family models. In *IEEE 14th ICDM*, Dec 2014.

[Neiswanger *et al.*, 2014] W. Neiswanger, C. Wang, Q. Ho, and E. P. Xing. Modeling citation networks using latent random offsets. In *Proceedings of 30th Conference on UAI*, pages 485–492, 2014.

[Peixoto, 2014] Tiago P. Peixoto. Hierarchical block structures and high-resolution model selection in large networks. *Phys. Rev. X*, 4:011047, Mar 2014.

[Pfeffer *et al.*, 1999] Avi Pfeffer, Daphne Koller, Brian Milch, and Ken T. Takusagawa. Spook: A system for probabilistic object-oriented knowledge representation. In *Proceedings of the 15th Conference on UAI*, pages 541–550, 1999.

[Poole and Zhang, 2003] David Poole and Nevin Lianwen Zhang. Exploiting contextual independence in probabilistic inference. *J. Artif. Intell. Res.(JAIR)*, 18:263–313, 2003.

[Poole, 2003] David Poole. First-order probabilistic inference. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, 2003.

[Rohe *et al.*, 2011] Karl Rohe, Sourav Chatterjee, and Bin Yu. Spectral clustering and the high-dimensional stochastic blockmodel. *Ann. Statist.*, 39(4):1878–1915, 08 2011.

[Schein *et al.*, 2016] A. Schein, M. Zhou, D. M. Blei, and H. Wallach. Bayesian Poisson Tucker decomposition for learning the structure of international relations. In *ICML*, 2016.

[Sen *et al.*, 2009] Prithviraj Sen, Amol Deshpande, and Lise Getoor. Bisimulation-based approximate lifted inference. In *Proceedings of the 25th Conference on UAI*, pages 496–505, Arlington, VA, United States, 2009.

[Seshadhri *et al.*, 2012] C. Seshadhri, Tamara Kolda, and Ali Pinar. Community structure and scale-free collections of Erdős-Rényi graphs. *Physical Review E*, 85(5), 2012.

[Seshadhri *et al.*, 2013] C. Seshadhri, Ali Pinar, and Tamara Kolda. An in-depth analysis of stochastic Kronecker graphs. *Journal of the ACM*, 60(2), 2013.

[Van den Broeck and Darwiche, 2013] Guy Van den Broeck and Adnan Darwiche. On the complexity and approximation of binary evidence in lifted inference. In *Advances in NIPS*, 2013.

[Venugopal and Gogate, 2014] Deepak Venugopal and Vibhav Gogate. Evidence-based clustering for scalable inference in markov logic. In *Proceedings of ECML PKDD*, pages 258–273, 2014.

[Yun and Vishwanathan, 2012] Hyokun Yun and S. V. N. Vishwanathan. Quilting stochastic kronecker product graphs to generate multiplicative attribute graphs. In *AISTATS*, pages 1389–1397, 2012.