EXACT REPRESENTATION OF SPARSE NETWORKS WITH SYMMETRIC NONNEGATIVE EMBEDDINGS

Anonymous authors
Paper under double-blind review

ABSTRACT

Many models for undirected graphs are based on factorizing the graph's adjacency matrix; these models find a vector representation of each node such that the predicted probability of a link between two nodes increases with the similarity (dot product) of their associated vectors. Recent work has shown that these models are unable to capture key structures in real-world graphs, particularly heterophilous structures, wherein links occur between dissimilar nodes. In contrast, a factorization with two vectors per node, based on logistic principal components analysis (LPCA), has been proven not only to represent such structures, but also to provide exact low-rank factorization of any graph with bounded max degree. However, this bound has limited applicability to real-world networks, which often have power law degree distributions with high max degree. Further, the LPCA model lacks interpretability since its asymmetric factorization does not reflect the undirectedness of the graph. We address the above issues in two ways. First, we prove a new bound for the LPCA model in terms of *arboricity* rather than max degree; this greatly increases the bound's applicability to many sparse real-world networks. Second, we propose an alternative graph model whose factorization is symmetric and nonnegative, which allows for link predictions to be interpreted in terms of node clusters. We show that the bounds for exact representation in the LPCA model extend to our new model. On the empirical side, our model is optimized effectively on real-world graphs with gradient descent on a cross-entropy loss. We demonstrate its effectiveness on a variety of foundational tasks, such as community detection and link prediction.

1 INTRODUCTION

Graphs naturally arise in data from a variety of fields including sociology (Mason & Verwoerd, 2007), biology (Scott, 1988), and computer networking (Bonato, 2004). A key underlying task in machine learning for graph data is forming models of graphs which can predict edges between nodes, form useful representations of nodes, and reveal interpretable structure in the graph, such as detecting clusters of nodes. Many graph models fall under the framework of edge-independent graph generative models, which can output the probabilities of edges existing between any pair of nodes. The parameters of such models can be trained iteratively on the network, or some fraction of the network which is known, in the link prediction task, i.e., by minimizing a predictive loss. To choose among these models, one must consider two criteria: 1) whether the model can express structures of interest in the graph, 2) whether the model expresses these structure in an interpretable way.

Expressiveness of low-dimensional embeddings As real-world graphs are high-dimensional objects, graph models generally compress information about the graph. Such models are exemplified by the family of dot product models, which associate each node with a real-valued "embedding" vector; the predicted probability of a link between two nodes increases with the similarity of their embedding vectors. These models can alternatively be seen as factorizing the graph's adjacency matrix to approximate it with a low-rank matrix. Recent work of Seshadhri et al. (2020) has shown that dot product models are limited in their ability to model common structures in real-world graphs, such as triangles incident only on low-degree nodes. In response, Chanpuriya et al. (2020) showed that with the logistic principal components analysis (LPCA) model, which has two embeddings per node (i.e., using the dot product of the 'left' embedding of one node and the 'right' embedding of another), not only can such structures be represented, but further, any graph can be exactly represented

with embedding vectors whose lengths are linear in the max degree of the graph. There are two keys to this result. First is the presence of a nonlinear linking function in the LPCA model; since adjacency matrices are generally not low-rank, exact low-rank factorization is generally impossible without a linking function. Second is that having two embeddings rather than one allows for expression of non-positive semidefinite (PSD) matrices. As discussed in Peysakhovich & Bottou (2021) that the single-embedding models can only represent PSD matrices precludes representation of 'heterophilous' structures in graphs; heterophilous structures are those wherein dissimilar nodes are linked, in contrast to more intuitive 'homophilous' linking between similar nodes.

Interpretability and node clustering Beyond being able to capture a given network accurately, it is often desirable for a graph model to form interpretable representations of nodes and to produce edge probabilities in an interpretable fashion. Dot product models can achieve this by restricting the node embeddings to be nonnegative. Nonnegative factorization has long been used to decompose data into parts (Donoho & Stodden, 2003). In the context of graphs, this entails decomposing the set of nodes of the network into clusters or communities. In particular, each entry of the nonnegative embedding vector of a node represents the intensity with which the node participates in a community. This allows the edge probabilities output by dot product models to be interpretable in terms of coparticipation in communities. Depending on the model, these vectors may have restrictions such as a sum-to-one requirement, meaning the node is assigned a categorical distribution over communities. The least restrictive and most expressive case is that of soft assignments to overlapping communities, where the entries can vary totally independently. In such models, which include the BIGCLAM model of Yang & Leskovec (2013), the output of the dot product may be mapped through a nonlinear link function (as in LPCA) to produce a probability for each edge, i.e., to ensure the values lie in $[0, 1]$.

Heterophily: Motivating example To demonstrate how heterophily can manifest in networks, as well as how models which assume homophily can fail to represent such networks, we provide a simple synthetic example. Suppose we have a graph of matches between users of a mostly heterosexual dating app, and the users each come from one of ten cities. Members from the same city are likely to match with each other; this typifies homophily, wherein links occur between similar nodes. Furthermore, users having the same gender are are unlikely to match with each other; this typifies heterophily. (Note that a mostly homosexual dating app would not exhibit heterophily in this sense.) Figure 1 shows an instantiation of such an adjacency matrix with 1000 nodes, which are randomly assigned to man or woman and to one of the ten cities. We recreate this network with our proposed embedding model and with BIGCLAM, which explicitly assumes homophily. (It is far from alone in this assumption; see Li et al. (2018) for a recent example, along with more examples and further discussion in Section 3.) We also compare with the SVD of the adjacency matrix, which outputs the best (lowest Frobenius error) low-rank approximation that is possible without a nonlinear linking function. Since SVD lacks nonnegativity constraints on the factors, we do not expect intepretability. In Figure 1, we show how BIGCLAM captures only the ten communities based on city, i.e., only the homophilous structure, and fails to capture the heterophilous distinction between men and women. We also plot the error of the reconstructions as the embedding length increases. There are $10 \cdot 2 = 20$ different kinds of nodes, meaning the expected adjacency matrix is rank-20, and our model maintains the lowest error up to this embedding length; by contrast, BIGCLAM is unable to decrease error after capturing city information with length-10 embeddings. In Figure 3, we visualize the features generated by the three methods, i.e., the factors returned by each factorization. Our model's factors captures the relevant latent structure in an interpretable way. By contrast, SVD's factors are harder to interpret, and BIGCLAM does not represent the heterophilous structure.

Summary of main contributions The key contributions of this work are as follows:

- We prove that the LPCA model admits exact low-rank factorizations of graphs with bounded *arboricity*, which is the minimum number of forests into which a graph's edges can be partitioned. By the Nash-Williams theorem, arboricity is a measure of a graph's density in that, letting S denote an induced subgraph and n_S and m_S denote the number of nodes and edges in S, arboricity is the maximum over all subgraphs S of $\lceil \frac{m_S}{n_S - 1} \rceil$. Our result is more applicable to real-world graphs than the prior one for graphs with bounded max degree, since sparsity is a common feature of real networks, whereas low max degree is not.

- We introduce a graph model which is both highly expressive and interpretable. Our model incorporates two embeddings per node and a nonlinear linking function, and hence is able to express both heterophily and overlapping communities. At the same time, our model is based on symmetric nonnegative matrix factorization, so it outputs link probabilities which are interpretable in terms of the communities it detects.
- We show how any graph with a low-rank factorization in the LPCA model also admits a low-rank factorization in our community-based model. This means that the guarantees on low-rank representation for bounded max degree and arboricity also apply to our model.
- In experiments, we show that our method is competitive with and often outperforms other comparable models on real-world graphs in terms of representing the network, doing interpretable link prediction, and detecting communities that align with ground-truth.

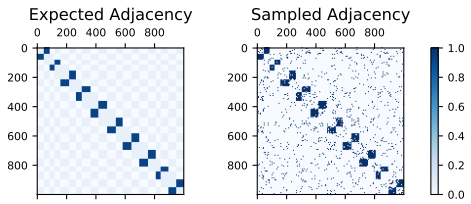

Figure 1: The motivating synthetic graph. The expected adjacency matrix (left) and the sampled matrix (right); the latter, which is passed to the training algorithms, is produced by treating the entries of the former as parameters of Bernoulli distributions and sampling. The network is approximately a union of ten bipartite graphs, each of which correspond to men and women in one of the ten cities.

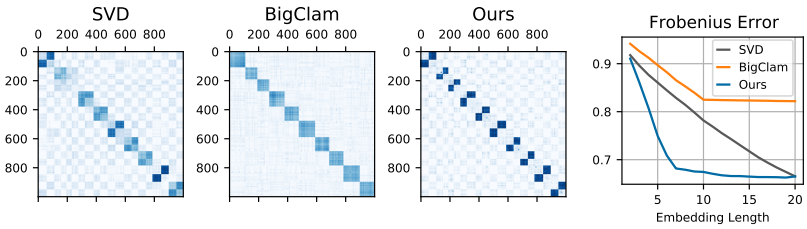

Figure 2: Left: Reconstructions of the motivating synthetic graph of Figure 1 with SVD, BIGCLAM, and our model, using 12 communities or singular vectors. Note the lack of the small diagonal structure in BIGCLAM's reconstruction; this corresponds to its inability to capture the heterophilous interaction between men and women. Right: Frobenius error when reconstructing the motivating synthetic graph of Figure 1 with SVD, BIGCLAM, and our model, as the embedding length is varied. The error is normalized by the sum of the true adjacency matrix (i.e., twice the number of edges).

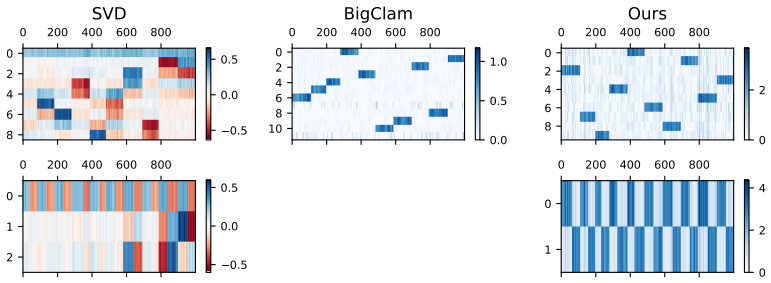

Figure 3: Factors resulting from decomposition of the motivating synthetic graph of Figure 1 with the three models, using 12 communities or singular vectors. The top/bottom rows represent the positive/negative eigenvalues corresponding to homophilous/heterophilous communities (note that BIGCLAM does not include the latter). The homophilous factors from BIGCLAM and our model reflect the 10 cities, and the heterophilous factor from our model reflect men and women. The factors from SVD are harder to interpret. Note that the order of the communities in the factors is arbitrary.

2 COMMUNITY-BASED GRAPH FACTORIZATION MODEL

Consider the set of undirected, unweighted graphs on n nodes, i.e., the set of graphs with symmetric adjacency matrices in $\{0,1\}^{n\times n}$. We propose an edge-independent generative model for such graphs. Given nonnegative parameter matrices $\boldsymbol{B} \in \mathbb{R}_+^{n\times k_B}$ and $\boldsymbol{C} \in \mathbb{R}_+^{n\times k_C}$, we set the probability of an edge existing between nodes i and j to be the (i,j)-th entry of matrix $\tilde{\boldsymbol{A}}$:

$$\tilde{\boldsymbol{A}} := \sigma(\boldsymbol{B}\boldsymbol{B}^\top - \boldsymbol{C}\boldsymbol{C}^\top), \tag{1}$$

where σ is the logistic function. Here k_B, k_C are the number of homophilous/heterophilous clusters. Intuitively, if $\boldsymbol{b}_i \in \mathbb{R}_+^{k_B}$ is the i-th row of matrix $\boldsymbol{B}$, then $\boldsymbol{b}_i$ is the affinity of node i to each of the k_B homophilous communities. Similarly, $\boldsymbol{c}_i \in \mathbb{R}_+^{k_C}$ is the affinity of node i to the k_C heterophilous communities. As an equivalent statement, for each pair of nodes i and j, $\tilde{\boldsymbol{A}}_{i,j} := \sigma(\boldsymbol{b}_i\boldsymbol{b}_j^\top - \boldsymbol{c}_i\boldsymbol{c}_j^\top)$. We will soon discuss the precise interpretation of this model, but the idea is roughly similar to the attract-repel framework of Peysakhovich & Bottou (2021). When nodes i and j have similar 'attractive' $\boldsymbol{b}$ embeddings, i.e., when $\boldsymbol{b}_i\boldsymbol{b}_j^\top$ is high, the likelihood of an edge between them increases, hence why the $\boldsymbol{B}$ factor is homophilous. By contrast, the $\boldsymbol{C}$ factor is 'repulsive'/heterophilous since, when $\boldsymbol{c}_i\boldsymbol{c}_j^\top$ is high, the likelihood of an edge between i and j decreases.

Alternate expression We note that the model above can also be expressed in a form which normalizes cluster assignments and is more compact, in that it combines the homophilous and heterophilous cluster assignments. Instead of $\boldsymbol{B}$ and $\boldsymbol{C}$, this form uses a matrix $\boldsymbol{V} \in [0,1]^{n\times k}$ and a diagonal matrix $\boldsymbol{W} \in \mathbb{R}^{k\times k}$, where $k = k_B + k_C$ is the total number of clusters. In particular, let $\boldsymbol{m}_B$ and $\boldsymbol{m}_C$ be the vectors containing the maximums of each column of $\boldsymbol{B}$ and $\boldsymbol{C}$. By setting

$$\begin{aligned} \boldsymbol{V} &= \left(\boldsymbol{B} \times \operatorname{diag}\left(\boldsymbol{m}_B^{-1}\right); \quad \boldsymbol{C} \times \operatorname{diag}\left(\boldsymbol{m}_C^{-1}\right)\right) \\ \boldsymbol{W} &= \operatorname{diag}\left(\left(+\boldsymbol{m}_B^2; \quad -\boldsymbol{m}_C^2\right)\right), \end{aligned} \tag{2}$$

the constraint on $\boldsymbol{V}$ is satisfied. Further, $\boldsymbol{V}\boldsymbol{W}\boldsymbol{V}^\top = \boldsymbol{B}\boldsymbol{B}^\top - \boldsymbol{C}\boldsymbol{C}^\top$, so

$$\tilde{\boldsymbol{A}} := \sigma(\boldsymbol{B}\boldsymbol{B}^\top - \boldsymbol{C}\boldsymbol{C}^\top) = \sigma(\boldsymbol{V}\boldsymbol{W}\boldsymbol{V}^\top). \tag{3}$$

Here, if $\mathbf{v}_i \in [0,1]^k$ is the i-th row of matrix $\boldsymbol{V}$, then $\mathbf{v}_i$ is the soft (normalized) assignment of node i to the k communities. The diagonal entries of $\boldsymbol{W}$ represent the strength of the homophily (if positive) or heterophily (if negative) of the communities. For each entry, $\tilde{\boldsymbol{A}}_{i,j} = \sigma(\mathbf{v}_i\boldsymbol{W}\mathbf{v}_j^\top)$. We use these two forms interchangeably throughout this work.

Interpretation The edge probabilities output by this model have an intuitive interpretation. Recall that there are bijections between probability $p \in [0,1]$, odds $o = \frac{p}{1-p} \in [0,\infty)$, and logit $\ell = \log(o) \in (-\infty,+\infty)$. The logit of the link probability between nodes i and j is $\mathbf{v}_i^\top\boldsymbol{W}\mathbf{v}_j$, which is a summation of terms $\mathbf{v}_{ic}\mathbf{v}_{jc}\boldsymbol{W}_{cc}$ over all communities $c \in [k]$. If the nodes both fully participate in community c, that is, $\mathbf{v}_{ic} = \mathbf{v}_{jc} = 1$, then the edge logit is changed by $\boldsymbol{W}_{cc}$ starting from a baseline of 0, or equivalently, the odds of an edge is multiplied by $\exp(\boldsymbol{W}_{cc})$ starting from a baseline odds of 1; if either of the nodes participates only partially in community c, then the change in logit and odds is accordingly prorated. Homophily and heterophily also have a clear interpretation in this model: homophilous communities, which are expressed in $\boldsymbol{B}$, are those with $\boldsymbol{W}_{cc} > 0$, where two nodes both participating in the community increases the odds of a link, whereas communities with $\boldsymbol{W}_{cc} < 0$, which are expressed in $\boldsymbol{C}$, are heterophilous, and coparticipation decreases the odds of a link.

3 RELATED WORK

Community detection via interpretable factorizations There is extensive prior work on the community detection / node clustering problem (Schaeffer, 2007; Aggarwal & Wang, 2010; Nascimento & De Carvalho, 2011), perhaps the most well-known being the normalized cuts algorithm of Shi & Malik (2000), which produces a clustering based on the entrywise signs of an eigenvector of the graph Laplacian matrix. However, the clustering algorithms which are most relevant to our work are those based on non-negative matrix factorization (NMF) (Lee & Seung, 1999; Berry et al., 2007; Wang & Zhang, 2012; Gillis, 2020), many of which can be seen as integrating nonnegativity constraints into

the broader, well-studied *random dot product graph* (RDPG) model (Young & Scheinerman, 2007; Scheinerman & Tucker, 2010; Athreya et al., 2017; Gallagher et al., 2021; Marenco et al., 2022). One such algorithm is that of Yu et al. (2005), which approximately factors a graph's adjacency matrix $\boldsymbol{A} \in \{0,1\}^{n\times n}$ into two positive matrices $\boldsymbol{H}$ and $\boldsymbol{\Lambda}$, where $\boldsymbol{H} \in \mathbb{R}_+^{n\times k}$ is left-stochastic (i.e. each of its columns sums to 1) and $\boldsymbol{\Lambda} \in \mathbb{R}_+^{k\times k}$ is diagonal, such that $\boldsymbol{H}\boldsymbol{\Lambda}\boldsymbol{H}^\top \approx \boldsymbol{A}$. Here $\boldsymbol{H}$ represents a soft clustering of the n nodes into k clusters, while the diagonal entries of $\boldsymbol{\Lambda}$ represent the prevalence of edges within clusters. Note the similarity of the factorization to our model, save for the lack of a nonlinearity. Other NMF approaches include those of Ding et al. (2008), Yang et al. (2012), Kuang et al. (2012), and Kuang et al. (2015) (SYMNMF).

Modeling heterophily Much of the existing work on graph models has an underlying assumption of network homophily (Johnson et al., 2010; Noldus & Van Mieghem, 2015). There has been significant recent interest in the limitations of graph neural network (GNN) models (Duvenaud et al., 2015; Kipf & Welling, 2017; Hamilton et al., 2017) at addressing network heterophily (NT & Maehara, 2019; Zhu et al., 2020; Zheng et al., 2022), as well as proposed solutions (Pei et al., 2020; Yan et al., 2021), but relatively less work for more fundamental models such as those for clustering. Some existing NMF approaches to clustering do naturally model heterophilous structure in networks. For example, the Generalized RDPG (Rubin-Delanchy et al., 2017) allows for the representation of non-PSD adjacency matrices by using a "tri-factorization" like that of Yu et al. (2005), but removing the constraint that the central diagonal matrix must be nonnegative. The model of Nourbakhsh et al. (2014) goes further, removing all constraints on this matrix. Another example is the model in Miller et al. (2009), which is similar to ours, though it restricts the cluster assignment matrix $\boldsymbol{V}$ to be binary; additionally, their training algorithm is not based on gradient descent as ours is, and it does not scale to larger networks. More recently, Peysakhovich & Bottou (2021) propose a decomposition of the form $\boldsymbol{A} \approx \boldsymbol{D} + \boldsymbol{B}\boldsymbol{B}^\top - \boldsymbol{C}\boldsymbol{C}^\top$, where $\boldsymbol{D} \in \mathbb{R}^{n\times n}$ is diagonal and $\boldsymbol{B}, \boldsymbol{C} \in \mathbb{R}^{n\times k}$ are low-rank. Note that their decomposition does not include a nonlinear linking function, and their work does not pursue a clustering interpretation or investigate setting the factors $\boldsymbol{B}$ and $\boldsymbol{C}$ to be nonnegative.

Overlapping communities and exact embeddings Many models discussed above focus on the single-label clustering task and thus involve highly-constrained factorizations (e.g., sum-to-one conditions). We are interested in the closely related but distinct task of multi-label clustering, also known as overlapping community detection (Xie et al., 2013; Javed et al., 2018), which involves less constrained, more expressive factorizations. The BIGCLAM algorithm of Yang & Leskovec (2013) uses the following generative model for this task: the probability of a link between two nodes i and j is given by $1 - \exp(-\boldsymbol{f}_i \cdot \boldsymbol{f}_j)$, where $\boldsymbol{f}_i, \boldsymbol{f}_j \in \mathbb{R}_+^k$ represent the intensities with which the nodes participate in each of the k communities. Note that BIGCLAM assumes strict homophily of the communities: two nodes participating in the same community always increases the probability of a link. However, this model allows for expression of very dense intersections of communities, which the authors observe is generally a characteristic of real-world networks. To ensure that output entries are probabilities, BIGCLAM's factorization includes a nonlinear linking function (namely, $f(x) = 1 - e^x$), like our model and LPCA. Recent work outside clustering and community detection on graph generative models (Rendsburg et al., 2020; Chanpuriya et al., 2020) suggests that incorporating a nonlinear linking function can greatly increase the expressiveness of factorization-based graph models, to the point of being able to exactly represent a graph. This adds to a growing body of literature on expressiveness guarantees for embeddings on relational data (Sala et al., 2018; Bhattacharjee & Dasgupta, 2020; Boratko et al., 2021). Most relevant to our work, as previously discussed, Chanpuriya et al. (2020) provide a guarantee for exact low-rank representation of graphs with bounded max degree when using the LPCA factorization model. In this work, we provide a new such guarantee, except for bounded arboricity, which is more applicable to real-world networks, and extend these guarantees to our community-based factorization.

4 THEORETICAL RESULTS

We first restate the main result from Chanpuriya et al. (2020) on exact representation of graphs with bounded max degree using the logistic principal components analysis (LPCA) model, which reconstructs a graph $\boldsymbol{A} \in \{0,1\}^{n\times n}$ using logit factors $\boldsymbol{X}, \boldsymbol{Y} \in \mathbb{R}^{n\times k}$ via

$$\boldsymbol{A} \approx \sigma(\boldsymbol{X}\boldsymbol{Y}^\top). \tag{4}$$

Note that unlike our community-based factorization, the factors of the LPCA model are not nonnegative, and the factorization does not reflect the symmetry of the undirected graph's adjacency matrix. Regardless of the model's interpretability, the following theorem provides a significant guarantee on its expressiveness. We use the following notation: given a matrix $\boldsymbol{M}$, let $H(\boldsymbol{M})$ denote the matrix resulting from entrywise application of the Heaviside step function to $\boldsymbol{M}$, that is, setting all positive entries to 1, negative entries to 0, and zero entries to $1/2$.

Theorem 4.1 (Exact LPCA Factorization for Bounded-Degree Graphs, Chanpuriya et al. (2020))**.** *Let $\boldsymbol{A} \in \{0,1\}^{n \times n}$ be the adjacency matrix of a graph G with maximum degree c. Then there exist matrices $\boldsymbol{X}, \boldsymbol{Y} \in \mathbb{R}^{n \times (2c+1)}$ such that $\boldsymbol{A} = H(\boldsymbol{X}\boldsymbol{Y}^\top)$.*

This corresponds to arbitrarily small approximation error in the LPCA model (Equation 4) because, provided such factors $\boldsymbol{X}, \boldsymbol{Y}$ for some graph $\boldsymbol{A}$, we have that $\lim_{s \to \infty} \sigma\left(s\boldsymbol{X}\boldsymbol{Y}^\top\right) = H(\boldsymbol{X}\boldsymbol{Y}^\top) = \boldsymbol{A}$. That is, we can scale the factors larger to reduce the error to an arbitrary extent.

We expand on this result in two ways. First, give a new bound for exact embedding in terms of arboricity, rather than max degree. This increases the applicability to real-world networks, which often are sparse (i.e., low arboricity) and have right-skewed degree distributions (i.e., high max degree). Second, we show that any rank-k LPCA factorization can be converted to our model's symmetric nonnegative factorization with $O(k)$ communities. This extends the guarantees on the LPCA model's power for exact representation of graphs, both the prior guarantee in terms of max degree and our new one in terms of arboricity, to our community-based model as well. In Appendix A.1, we also introduce a natural family of graphs - Community Overlap Threshold (COT) graphs - for which our model's community-based factorization not only exactly represents the graph, but also must capture some latent structure to do so with sufficiently low embedding dimensionality.

Arboricity bound for exact representation We will use the following well-known fact: the rank of the entrywise product of two matrices is at most the product of their individual ranks, that is,

$$\text{rank}(\boldsymbol{X} \circ \boldsymbol{Y}) \leq \text{rank}(\boldsymbol{X}) \cdot \text{rank}(\boldsymbol{Y}).$$

Theorem 4.2 (Exact LPCA Factorization for Bounded-Arboricity Graphs)**.** *Let $\boldsymbol{A} \in \{0,1\}^{n \times n}$ be the adjacency matrix of an undirected graph G with arboricity α. Then there exist embeddings $\boldsymbol{X}, \boldsymbol{Y} \in \mathbb{R}^{n \times (4\alpha^2+1)}$ such that $\boldsymbol{A} = H(\boldsymbol{X}\boldsymbol{Y}^\top)$.*

Proof. Let the undirected graph $\boldsymbol{A}$ have arboricity α, i.e., the edges can be partitioned into α forests. We produce a directed graph $\boldsymbol{B}$ from $\boldsymbol{A}$ by orienting the edges in these forests so that each node's edges point towards its children. Now $\boldsymbol{A} = \boldsymbol{B} + \boldsymbol{B}^\top$, and every node in $\boldsymbol{B}$ has in-degree at most α.

Let $\boldsymbol{V} \in \mathbb{R}^{n \times 2\alpha}$ be the Vandermonde matrix with $\boldsymbol{V}_{t,j} = t^{j-1}$. For any $\boldsymbol{c} \in \mathbb{R}^{2\alpha}$, $[\boldsymbol{V}\boldsymbol{c}](t) = \sum_{j=1}^{2\alpha} \boldsymbol{c}(j) \cdot t^{j-1}$, that is, $\boldsymbol{V}\boldsymbol{c} \in \mathbb{R}^n$ is a degree-(2α) polynomial with coefficients $\boldsymbol{c}$ evaluated at the integers $t \in [n] = \{1, \ldots, n\}$. Let $\boldsymbol{b}_i$ be the ith column of $\boldsymbol{B}$. We seek to construct a polynomial such that for t with $\boldsymbol{b}_i(t) = 1$, $[\boldsymbol{V}\boldsymbol{c}_i](t) = 0$, and $[\boldsymbol{V}\boldsymbol{c}_i](t) < 0$ elsewhere; that is, when inputting an index $t \in [n]$ such that the tth node is an in-neighbor of the ith node, we want the polynomial to output 0, and for all other indices in $[n]$, we want it to have a negative output. Letting $N(i)$ denote the in-neighbors of the ith node, a simple instantiation of such a polynomial in t is $-1 \cdot \prod_{j \in N(i)} (t-j)^2$. Note that since all nodes have in-degree at most α, this polynomial's degree is at most 2α, and hence there exists a coefficient vector $\boldsymbol{c}_i \in \mathbb{R}^{2\alpha}$ encoding this polynomial.

Let $\boldsymbol{C} \in \mathbb{R}^{n \times 2\alpha}$ be the matrix resulting from stacking such coefficient vectors for each of the n nodes. Consider $\boldsymbol{P} = \boldsymbol{V}\boldsymbol{C} \in \mathbb{R}^{n \times n}$: $\boldsymbol{P}_{i,j}$ is 0 if $\boldsymbol{B}_{i,j} = 1$ and negative otherwise. Then $(\boldsymbol{P} \circ \boldsymbol{P}^\top)_{i,j}$ is 0 when either $\boldsymbol{B}_{i,j} = 1$ or $(\boldsymbol{B}^\top)_{i,j} = 1$ and positive otherwise; equivalently, since $\boldsymbol{A} = \boldsymbol{B} + \boldsymbol{B}^\top$, $(\boldsymbol{P} \circ \boldsymbol{P}^\top)_{i,j} = 0$ iff $\boldsymbol{A}_{i,j} = 1$. Take any positive ϵ less than the smallest positive entry of $\boldsymbol{P} \circ \boldsymbol{P}^\top$. Letting $\boldsymbol{J}$ be an all-ones matrix, define $\boldsymbol{M} = \epsilon\boldsymbol{J} - (\boldsymbol{P} \circ \boldsymbol{P}^\top)$. Note that $\boldsymbol{M}_{i,j} > 0$ if $\boldsymbol{A} = 1$ and $\boldsymbol{M}_{i,j} < 0$ if $\boldsymbol{A} = 0$, that is, $\boldsymbol{M} = H(\boldsymbol{A})$ as desired. Since $\text{rank}(\boldsymbol{J}) = 1$ and $\text{rank}(\boldsymbol{P}) \leq 2\alpha$, by the bound on the rank of entrywise products of matrices, the rank of $\boldsymbol{M}$ is at most $(2\alpha)^2 + 1$. ■

Exact representation with community factorization LPCA factors $\boldsymbol{X}, \boldsymbol{Y} \in \mathbb{R}^{n \times k}$ can be processed into nonnegative factors $\boldsymbol{B} \in \mathbb{R}_+^{n \times k_B}$ and $\boldsymbol{C} \in \mathbb{R}_+^{n \times k_C}$ such that $k_B + k_C = 6k$ and

$$\boldsymbol{B}\boldsymbol{B}^\top - \boldsymbol{C}\boldsymbol{C}^\top = \tfrac{1}{2}\left(\boldsymbol{X}\boldsymbol{Y}^\top + \boldsymbol{Y}\boldsymbol{X}^\top\right). \tag{5}$$

Observe that the left-hand side can only represent symmetric matrices, but $\boldsymbol{X}\boldsymbol{Y}^\top$ is not necessarily symmetric even if $H(\boldsymbol{X}\boldsymbol{Y}^\top) = \boldsymbol{A}$ for a symmetric $\boldsymbol{A}$. For this reason, we use a symmetrization: let $\boldsymbol{L} = \frac{1}{2}\left(\boldsymbol{X}\boldsymbol{Y}^\top + \boldsymbol{Y}\boldsymbol{X}^\top\right)$. Note that $H(\boldsymbol{L}) = H(\boldsymbol{X}\boldsymbol{Y}^\top)$, so if $\boldsymbol{X}\boldsymbol{Y}^\top$ constitutes an exact representation of $\boldsymbol{A}$ in that $H(\boldsymbol{X}\boldsymbol{Y}^\top) = \boldsymbol{A}$, so too do both expressions for $\boldsymbol{L}$ in Equation 5. Pseudocode for the procedure of constructing $\boldsymbol{B}, \boldsymbol{C}$ given $\boldsymbol{X}, \boldsymbol{Y}$ is given in Algorithm 1. The concept of this algorithm is to first separate the logit matrix $\boldsymbol{L}$ into a sum and difference of rank-1 components via eigendecomposition. Each of these components can be written as $+\mathbf{v}\mathbf{v}^\top$ or $-\mathbf{v}\mathbf{v}^\top$ with $\mathbf{v} \in \mathbb{R}^n$, where the sign depends on the sign of the eigenvalue. Each component is then separated into a sum and difference of three outer products of nonnegative vectors, via Lemma 4.3 below.

Lemma 4.3. *Let $\phi : \mathbb{R} \to \mathbb{R}$ denote the ReLU function, i.e., $\phi(z) = \max\{z, 0\}$. For any vector $\mathbf{v}$,*

$$\mathbf{v}\mathbf{v}^\top = 2\phi(\mathbf{v})\phi(\mathbf{v})^\top + 2\phi(-\mathbf{v})\phi(-\mathbf{v})^\top - |\mathbf{v}||\mathbf{v}|^\top.$$

Proof. Take any $\mathbf{v} \in \mathbb{R}^k$. Then

$$\begin{aligned}
\mathbf{v}\mathbf{v}^\top &= (\phi(\mathbf{v}) - \phi(-\mathbf{v})) \cdot (\phi(\mathbf{v})^\top - \phi(-\mathbf{v})^\top) \\
&= \phi(\mathbf{v})\phi(\mathbf{v})^\top + \phi(-\mathbf{v})\phi(-\mathbf{v})^\top - \phi(\mathbf{v})\phi(-\mathbf{v})^\top - \phi(-\mathbf{v})\phi(\mathbf{v})^\top \\
&= 2\phi(\mathbf{v})\phi(\mathbf{v})^\top + 2\phi(-\mathbf{v})\phi(-\mathbf{v})^\top - (\phi(\mathbf{v}) + \phi(-\mathbf{v})) \cdot (\phi(\mathbf{v}) + \phi(-\mathbf{v}))^\top \\
&= 2\phi(\mathbf{v})\phi(\mathbf{v})^\top + 2\phi(-\mathbf{v})\phi(-\mathbf{v})^\top - |\mathbf{v}||\mathbf{v}|^\top,
\end{aligned}$$

where the first step follows from $\mathbf{v} = \phi(\mathbf{v}) - \phi(-\mathbf{v})$, and the last step from $|\mathbf{v}| = \phi(\mathbf{v}) + \phi(-\mathbf{v})$. ■

Algorithm 1 follows from Lemma 4.3 and constitutes a constructive proof of the following theorem:

Theorem 4.4 (Exact Community Factorization from Exact LPCA Factorization)**.** *Given a symmetric matrix $\boldsymbol{A} \in \{0, 1\}$ and $\boldsymbol{X}, \boldsymbol{Y} \in \mathbb{R}^{n \times k}$ such that $\boldsymbol{A} = H(\boldsymbol{X}\boldsymbol{Y}^\top)$, there exist nonnegative matrices $\boldsymbol{B} \in \mathbb{R}_+^{n \times k_B}$ and $\boldsymbol{C} \in \mathbb{R}_+^{n \times k_C}$ such that $k_B + k_C = 6k$ and $\boldsymbol{A} = H(\boldsymbol{B}\boldsymbol{B}^\top - \boldsymbol{C}\boldsymbol{C}^\top)$.*

Algorithm 1 Converting LPCA Factorization to Community Factorization

input logit factors $\boldsymbol{X}, \boldsymbol{Y} \in \mathbb{R}^{n \times k}$
output $\boldsymbol{B} \in \mathbb{R}_+^{n \times k_B}$ and $\boldsymbol{C} \in \mathbb{R}_+^{n \times k_C}$ such that $k_B + k_C = 6k$ and
$\boldsymbol{B}\boldsymbol{B}^\top - \boldsymbol{C}\boldsymbol{C}^\top = \frac{1}{2}\left(\boldsymbol{X}\boldsymbol{Y}^\top + \boldsymbol{Y}\boldsymbol{X}^\top\right)$

1: Set $\boldsymbol{Q} \in \mathbb{R}^{n \times 2k}$ and $\boldsymbol{\lambda} \in \mathbb{R}^{2k}$ by truncated eigendecomposition such that
$\boldsymbol{Q} \times \mathrm{diag}(\boldsymbol{\lambda}) \times \boldsymbol{Q}^\top = \frac{1}{2}(\boldsymbol{X}\boldsymbol{Y}^\top + \boldsymbol{Y}\boldsymbol{X}^\top)$
2: $\boldsymbol{B}^* \leftarrow \boldsymbol{Q}^+ \times \mathrm{diag}(\sqrt{+\boldsymbol{\lambda}^+})$, where $\boldsymbol{\lambda}^+, \boldsymbol{Q}^+$ are the positive eigenvalues/vectors
3: $\boldsymbol{C}^* \leftarrow \boldsymbol{Q}^- \times \mathrm{diag}(\sqrt{-\boldsymbol{\lambda}^-})$, where $\boldsymbol{\lambda}^-, \boldsymbol{Q}^-$ are the negative eigenvalues/vectors
4: $\boldsymbol{B} \leftarrow \left(\sqrt{2}\phi(\boldsymbol{B}^*); \quad \sqrt{2}\phi(-\boldsymbol{B}^*); \quad |\boldsymbol{C}^*|\right)$ ▷ ϕ and $|\cdot|$ are entrywise ReLU and absolute value
5: $\boldsymbol{C} \leftarrow \left(\sqrt{2}\phi(\boldsymbol{C}^*); \quad \sqrt{2}\phi(-\boldsymbol{C}^*); \quad |\boldsymbol{B}^*|\right)$
6: **return** $\boldsymbol{B}, \boldsymbol{C}$

As stated in the introduction to this section, Theorem 4.4 extends any upper bound on the exact factorization dimensionality from the LPCA model to our community-based model. That is, up to a constant factor, the bound in terms of max degree from Theorem 4.1 and the bound in terms of arboricity from Theorem 4.2 also apply to our model; for brevity, we state just the latter here.

Corollary 4.5 (Exact Community Factorization for Bounded-Arboricity Graphs)**.** *Let $\boldsymbol{A} \in \{0, 1\}^{n \times n}$ be the adjacency matrix of an undirected graph G with arboricity α. Then there exist nonnegative embeddings $\boldsymbol{B} \in \mathbb{R}_+^{n \times k_B}$ and $\boldsymbol{C} \in \mathbb{R}_+^{n \times k_C}$ such that $k_B + k_C = 6(4\alpha^2 + 1)$ and $\boldsymbol{A} = H(\boldsymbol{B}\boldsymbol{B}^\top - \boldsymbol{C}\boldsymbol{C}^\top)$.*

Note that Corollary 4.5 is purely a statement about the capacity of our model; Theorem 4.2 stems from a constructive proof based on polynomial interpolation, and therefore so too does this corollary. We do not expect this factorization to be informative about the graph's latent structure. In the following Section 5, we will fit the model with an entirely different algorithm for downstream applications.

5 EXPERIMENTS

We now present a training algorithm to fit our model, then evaluate our method on a benchmark of five real-world networks. These are fairly common small to mid-size datasets ranging from around 1K to 10K nodes; for brevity, we defer the statistics and discussion of these datasets, including how some of them exhibit heterophily, to Appendix A.2.

5.1 TRAINING ALGORITHM

Given an input graph $\boldsymbol{A} \in \{0,1\}^{n\times n}$, we find low-rank nonnegative matrices $\boldsymbol{B}$ and $\boldsymbol{C}$ such that the model produces $\tilde{\boldsymbol{A}} = \sigma(\boldsymbol{B}\boldsymbol{B}^\top - \boldsymbol{C}\boldsymbol{C}^\top) \in (0,1)^{n\times n}$ as in Equation 1 which approximately matches $\boldsymbol{A}$. In particular, we train the model to minimize the sum of binary cross-entropies of the link predictions over all pairs of nodes:

$$R = -\sum \left(\boldsymbol{A} \log(\tilde{\boldsymbol{A}}) + (1-\boldsymbol{A}) \log(1-\tilde{\boldsymbol{A}}) \right), \tag{6}$$

where $\sum$ denotes the scalar summation of all entries in the matrix. We fit the parameters by gradient descent over this loss, as well as L_2 regularization of the factors $\boldsymbol{B}$ and $\boldsymbol{C}$, subject to the nonnegativity of $\boldsymbol{B}$ and $\boldsymbol{C}$. This algorithm is fairly straightforward; pseudocode is given in Algorithm 2. This is quite similar to the training algorithm of Chanpuriya et al. (2020), but in contrast to that work, which only targets an exact fit, we explore the expression of graph structure in the factors and their utility in downstream tasks. Regularization of the factors is implemented to this end to avoid overfitting. Though we outline a non-stochastic version of the training algorithm, it generalizes straightforwardly to a stochastic version, i.e., by sampling links and non-links for the loss function.

Algorithm 2 Fitting the Constrained Model

input adjacency matrix $\boldsymbol{A} \in \{0,1\}^{n\times n}$, regularization weight $\lambda \geq 0$, number of iterations I, number of homophilous/heterophilous communities k_B/k_C

output fitted factors $\boldsymbol{B} \in \mathbb{R}_+^{n\times k_B}$ and $\boldsymbol{C} \in \mathbb{R}_+^{n\times k_C}$ such that $\sigma(\boldsymbol{B}\boldsymbol{B}^\top - \boldsymbol{C}\boldsymbol{C}^\top) \approx \boldsymbol{A}$

1: Initialize $\boldsymbol{B}, \boldsymbol{C}$ by setting entries to independent samples of $\text{Unif}(0, 1/\sqrt{k_B})$, $\text{Unif}(0, 1/\sqrt{k_C})$
2: **for** $i \leftarrow 1$ to I **do**
3: $\quad \tilde{\boldsymbol{A}} \leftarrow \sigma(\boldsymbol{B}\boldsymbol{B}^\top - \boldsymbol{C}\boldsymbol{C}^\top)$
4: $\quad R \leftarrow -\sum \left(\boldsymbol{A} \log(\tilde{\boldsymbol{A}}) + (1-\boldsymbol{A}) \log(1-\tilde{\boldsymbol{A}}) \right)$
5: $\quad R \leftarrow R + \lambda \left(\|\boldsymbol{B}\|_F^2 + \|\boldsymbol{C}\|_F^2 \right)$
6: $\quad$ Calculate $\partial_{\boldsymbol{B},\boldsymbol{C}} R$ via differentiation through Steps 2 to 4
7: $\quad$ Update $\boldsymbol{B}, \boldsymbol{C}$ to minimize R using $\partial_{\boldsymbol{B},\boldsymbol{C}} R$, subject to $\boldsymbol{B}, \boldsymbol{C} \geq 0$
8: **end for**
9: **return** $\boldsymbol{B}, \boldsymbol{C}$

Implementation details Our implementation uses PyTorch (Paszke et al., 2019) for automatic differentiation and minimizes loss using the SciPy (Jones et al., 2001) implementation of the L-BFGS (Liu & Nocedal, 1989; Zhu et al., 1997) algorithm with default hyperparameters and up to a max of 200 iterations of optimization. We set regularization weight $\lambda = 10$ as in Yang & Leskovec (2013). We include code in the form of a Jupyter notebook (Pérez & Granger, 2007) demo. The code also contains stochastic (i.e., more scalable) versions of the optimization functions.

5.2 RESULTS

Expressiveness First, we investigate the expressiveness of our generative model, that is, the fidelity with which it can reproduce an input network. In Section 1, we used a simple synthetic network to show that our model is more expressive than others due to its ability to represent heterophilous structures in addition to homophilous ones. We now evaluate the expressiveness of our model on real-world networks. As with the synthetic graph, we fix the number of communities or singular vectors, fit the model, then evaluate the reconstruction error. In Figure 4, we compare the results of our model with those of SVD, BIGCLAM (which is discussed in detail in Section 3), and SYMNMF (Kuang et al., 2015). SYMNMF simply factors the adjacency matrix as $\boldsymbol{A} \approx \boldsymbol{H}\boldsymbol{H}^\top$, where $\boldsymbol{H} \in \mathbb{R}_+^{n\times k}$; note that, like SVD, SYMNMF does not necessarily output a matrix whose entries are probabilities (i.e., bounded in $[0,1]$), and hence it is not a graph generative model like ours and BIGCLAM.

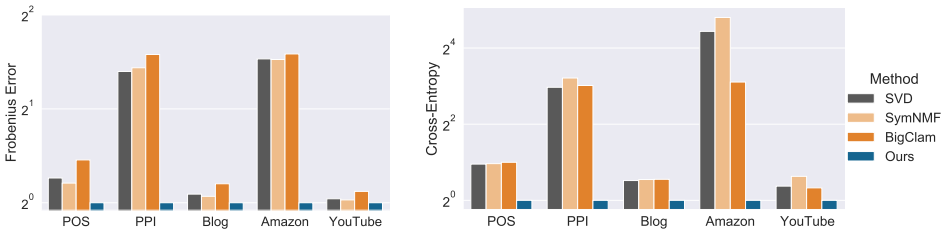

Figure 4: Reconstruction error on real-world networks, relative to our model's error.

For each method, we fix the number of communities or singular vectors at the ground-truth number. For this experiment only, we are not concerned with learning the latent structure of the graph; the only goal is accurate representation of the network with limited parameters. So, for a fair comparison with SVD, we do not regularize the training of the other methods. Our method consistently has the lowest reconstruction error, both in terms of Frobenius error and entrywise cross-entropy (Equation 6). Interestingly, we find the most significant improvement exactly on the three datasets which have been noted to exhibit significant heterophily: POS, PPI, and AMAZON.

Similarity to ground-truth communities To assess the interpretability of clusters generated by our method, we evaluate the similarity of these clusters to ground-truth communities (i.e., class labels), and we compare other methods for overlapping clustering. We additionally compare to another recent but non-generative approach, the VGRAPH method of Sun et al. (2019), which is based on link clustering; the authors found their method to generally achieve state-of-the-art results in this task. For all methods, we set the number of communities to be detected as the number of ground-truth communities. We report F1-Score as computed in Yang & Leskovec (2013). See Figure 5 (left): the performance of our method is competitive with SYMNMF, BIGCLAM, and vGraph.

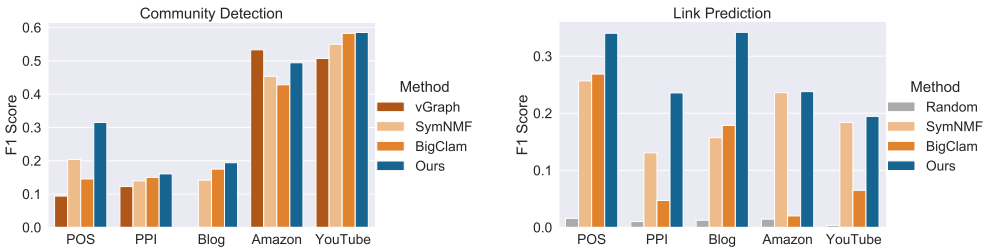

Figure 5: Left: Similarity of recovered communities to ground-truth labels of real-world datasets. We are unable to run the authors' implementation of VGRAPH on BLOG with limited memory. Right: Accuracy of link prediction on real-world datasets.

Interpretable link prediction We assess the predictive power of our generative model on the link prediction task. As discussed in Section 2, the link probabilities output by our model are interpretable in terms of a clustering of nodes that it generates; we compare results with our method to those with other models which permit similar interpretation, namely BIGCLAM and SYMNMF. We randomly select 10% of node pairs to hold out, fit the models on the remaining 90%, then use the trained models to predict links between node pairs in the held out 10%. As a baseline, we also show results for randomly predicting link or no link with equal probability. See Figure 5 (right). The performance of our method is competitive with or exceeds that of the other methods in terms of F1 Score.

6 CONCLUSION

We introduce a community-based graph generative model based on symmetric nonnegative matrix factorization which is capable of representing both homophily and heterophily. We expand on prior work guaranteeing exact representation of bounded max degree graphs with a new, more applicable guarantee for bounded arboricity graphs, and we show that both the prior bound and our new one apply to our more interpretable graph model. We illustrate our model's capabilities with experiments on a synthetic motivating example. Experiments on real-world networks show its effectiveness on several key tasks. More broadly, our results suggest that incorporating heterophily into models and methods for networks can improve both theoretical grounding and overall empirical performance, while maintaining simplicity and interpretability. A deeper understanding of the expressiveness of both nonnegative and arbitrary low-rank logit models for graphs is an interesting future direction.

REFERENCES

Charu C Aggarwal and Haixun Wang. A survey of clustering algorithms for graph data. In *Managing and Mining Graph Data*, pp. 275–301. Springer, 2010.

Avanti Athreya, Donniell E Fishkind, Minh Tang, Carey E Priebe, Youngser Park, Joshua T Vogelstein, Keith Levin, Vince Lyzinski, and Yichen Qin. Statistical inference on random dot product graphs: a survey. *The Journal of Machine Learning Research*, 18(1):8393–8484, 2017.

Michael W Berry, Murray Browne, Amy N Langville, V Paul Pauca, and Robert J Plemmons. Algorithms and applications for approximate nonnegative matrix factorization. *Computational Statistics & Data Analysis*, 52(1):155–173, 2007.

Robi Bhattacharjee and Sanjoy Dasgupta. What relations are reliably embeddable in euclidean space? In *Algorithmic Learning Theory*, pp. 174–195. PMLR, 2020.

Anthony Bonato. A survey of models of the web graph. In *Workshop on Combinatorial and Algorithmic Aspects of Networking*, pp. 159–172. Springer, 2004.

Michael Boratko, Dongxu Zhang, Nicholas Monath, Luke Vilnis, Kenneth L Clarkson, and Andrew McCallum. Capacity and bias of learned geometric embeddings for directed graphs. *Advances in Neural Information Processing Systems*, 34:16423–16436, 2021.

Bobby-Joe Breitkreutz, Chris Stark, Teresa Reguly, Lorrie Boucher, Ashton Breitkreutz, Michael Livstone, Rose Oughtred, Daniel H Lackner, Jürg Bähler, Valerie Wood, et al. The biogrid interaction database: 2008 update. *Nucleic acids research*, 36(suppl_1):D637–D640, 2007.

Sudhanshu Chanpuriya, Cameron Musco, Konstantinos Sotiropoulos, and Charalampos Tsourakakis. Node embeddings and exact low-rank representations of complex networks. *Advances in Neural Information Processing Systems*, 33, 2020.

Chris Ding, Tao Li, and Michael I Jordan. Nonnegative matrix factorization for combinatorial optimization: Spectral clustering, graph matching, and clique finding. In *2008 Eighth IEEE International Conference on Data Mining*, pp. 183–192. IEEE, 2008.

David L. Donoho and Victoria Stodden. When does non-negative matrix factorization give a correct decomposition into parts? In *Advances in Neural Information Processing Systems 16*, pp. 1141–1148. MIT Press, 2003.

David Duvenaud, Dougal Maclaurin, Jorge Aguilera-Iparraguirre, Rafael Gómez-Bombarelli, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P. Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in Neural Information Processing Systems 28*, pp. 2224–2232, 2015.

Jacob G Foster, David V Foster, Peter Grassberger, and Maya Paczuski. Edge direction and the structure of networks. *Proceedings of the National Academy of Sciences*, 107(24):10815–10820, 2010.

Ian Gallagher, Andrew Jones, and Patrick Rubin-Delanchy. Spectral embedding for dynamic networks with stability guarantees. *Advances in Neural Information Processing Systems*, 34:10158–10170, 2021.

Nicolas Gillis. *Nonnegative Matrix Factorization*. SIAM, 2020.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Takeshi Hase, Yoshihito Niimura, and Hiroshi Tanaka. Difference in gene duplicability may explain the difference in overall structure of protein-protein interaction networks among eukaryotes. *BMC Evolutionary Biology*, 10(1):1–15, 2010.

Muhammad Aqib Javed, Muhammad Shahzad Younis, Siddique Latif, Junaid Qadir, and Adeel Baig. Community detection in networks: A multidisciplinary review. *Journal of Network and Computer Applications*, 108:87–111, 2018.

Samuel Johnson, Joaquín J Torres, J Marro, and Miguel A Munoz. Entropic origin of disassortativity in complex networks. *Physical Review Letters*, 104(10):108702, 2010.

Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001. URL http://www.scipy.org/.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *International Conference on Learning Representations*, 2017.

Da Kuang, Chris Ding, and Haesun Park. Symmetric nonnegative matrix factorization for graph clustering. In *Proceedings of the 2012 SIAM International Conference on Data Mining*, pp. 106–117. SIAM, 2012.

Da Kuang, Sangwoon Yun, and Haesun Park. Symnmf: nonnegative low-rank approximation of a similarity matrix for graph clustering. *Journal of Global Optimization*, 62(3):545–574, 2015.

Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

Ye Li, Chaofeng Sha, Xin Huang, and Yanchun Zhang. Community detection in attributed graphs: An embedding approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Dong C Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1-3):503–528, 1989.

Bernardo Marenco, Paola Bermolen, Marcelo Fiori, Federico Larroca, and Gonzalo Mateos. Online change point detection for weighted and directed random dot product graphs. *IEEE Transactions on Signal and Information Processing over Networks*, 8:144–159, 2022.

Oliver Mason and Mark Verwoerd. Graph theory and networks in biology. *IET Systems Biology*, 1 (2):89–119, 2007.

Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, 27(1):415–444, 2001.

Kurt T. Miller, Thomas L. Griffiths, and Michael I. Jordan. Nonparametric latent feature models for link prediction. In *Advances in Neural Information Processing Systems 22*, pp. 1276–1284. Curran Associates, Inc., 2009.

Maria CV Nascimento and Andre CPLF De Carvalho. Spectral methods for graph clustering–a survey. *European Journal of Operational Research*, 211(2):221–231, 2011.

Mark EJ Newman. Assortative mixing in networks. *Physical Review Letters*, 89(20):208701, 2002.

Rogier Noldus and Piet Van Mieghem. Assortativity in complex networks. *Journal of Complex Networks*, 3(4):507–542, 2015.

Farshad Nourbakhsh, Samuel Rota Bulo, and Marcello Pelillo. A matrix factorization approach to graph compression. In *2014 22nd International Conference on Pattern Recognition*, pp. 76–81. IEEE, 2014.

Hoang NT and Takanori Maehara. Revisiting graph neural networks: All we have is low-pass filters. *arXiv preprint arXiv:1905.09550*, 2019.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019.

Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-gcn: Geometric graph convolutional networks. In *8th International Conference on Learning Representations, ICLR 2020*, 2020.

Fernando Pérez and Brian E. Granger. IPython: a system for interactive scientific computing. *Computing in Science and Engineering*, 9(3):21–29, May 2007. ISSN 1521-9615. doi: 10.1109/MCSE.2007.53. URL https://ipython.org.

Alexander Peysakhovich and Leon Bottou. An attract-repel decomposition of undirected networks. *arXiv preprint arXiv:2106.09671*, 2021.

Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, and Jie Tang. Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pp. 459–467. ACM, 2018.

Luca Rendsburg, Holger Heidrich, and Ulrike Von Luxburg. Netgan without gan: From random walks to low-rank approximations. In *International Conference on Machine Learning*, pp. 8073–8082. PMLR, 2020.

Patrick Rubin-Delanchy, Joshua Cape, Minh Tang, and Carey E Priebe. A statistical interpretation of spectral embedding: the generalised random dot product graph. *arXiv preprint arXiv:1709.05506*, 2017.

Frederic Sala, Chris De Sa, Albert Gu, and Christopher Ré. Representation tradeoffs for hyperbolic embeddings. In *International Conference on Machine Learning*, pp. 4460–4469. PMLR, 2018.

Satu Elisa Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27–64, 2007.

Edward R Scheinerman and Kimberly Tucker. Modeling graphs using dot product representations. *Computational statistics*, 25(1):1–16, 2010.

John Scott. Social network analysis. *Sociology*, 22(1):109–127, 1988.

C Seshadhri, Aneesh Sharma, Andrew Stolman, and Ashish Goel. The impossibility of low-rank representations for triangle-rich complex networks. *Proceedings of the National Academy of Sciences*, 117(11):5631–5637, 2020.

Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

Fan-Yun Sun, Meng Qu, Jordan Hoffmann, Chin-Wei Huang, and Jian Tang. vgraph: A generative model for joint community detection and node representation learning. In *Advances in Neural Information Processing Systems 32*, pp. 512–522, 2019.

Lei Tang and Huan Liu. Relational learning via latent social dimensions. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 817–826. ACM, 2009.

Yu-Xiong Wang and Yu-Jin Zhang. Nonnegative matrix factorization: A comprehensive review. *IEEE Transactions on Knowledge and Data Engineering*, 25(6):1336–1353, 2012.

Jierui Xie, Stephen Kelley, and Boleslaw K. Szymanski. Overlapping community detection in networks: The state-of-the-art and comparative study. *ACM Computing Surveys*, 45(4):43:1–43:35, 2013.

Yujun Yan, Milad Hashemi, Kevin Swersky, Yaoqing Yang, and Danai Koutra. Two sides of the same coin: Heterophily and oversmoothing in graph convolutional neural networks. *arXiv preprint arXiv:2102.06462*, 2021.

Jaewon Yang and Jure Leskovec. Overlapping community detection at scale: a nonnegative matrix factorization approach. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, pp. 587–596, 2013.

Jaewon Yang and Jure Leskovec. Defining and evaluating network communities based on ground-truth. *Knowledge and Information Systems*, 42(1):181–213, 2015.

Zhirong Yang, Tele Hao, Onur Dikmen, Xi Chen, and Erkki Oja. Clustering by nonnegative matrix factorization using graph random walk. In *Advances in Neural Information Processing Systems*, pp. 1079–1087, 2012.

Stephen J Young and Edward R Scheinerman. Random dot product graph models for social networks. In *International Workshop on Algorithms and Models for the Web-Graph*, pp. 138–149. Springer, 2007.

Kai Yu, Shipeng Yu, and Volker Tresp. Soft clustering on graphs. In *Advances in Neural Information Processing Systems*, pp. 1553–1560, 2005.

Xin Zheng, Yixin Liu, Shirui Pan, Miao Zhang, Di Jin, and Philip S Yu. Graph neural networks for graphs with heterophily: A survey. *arXiv:2202.07082*, 2022.

Ciyou Zhu, Richard H Byrd, Peihuang Lu, and Jorge Nocedal. Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software (TOMS)*, 23(4):550–560, 1997.

Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs. *Advances in Neural Information Processing Systems*, 33, 2020.

Katharina Anna Zweig. Are word-adjacency networks networks? In *Towards a theoretical framework for analyzing complex linguistic networks*, pp. 153–163. Springer, 2016.

A APPENDIX

A.1 COT GRAPH EXACT REPRESENTATION

As a theoretical demonstration of the capability of our model to learn latent structure, we additionally show that our model can exactly represent a natural family of graphs, which exhibits both homophily and heterophily, with small k and interpretably. The family of graphs is specified below in Definition 1; roughly speaking, nodes in such graphs share an edge iff they coparticipate in some number of homophilous communities and don't coparticipate in a number of heterophilous communities. For example, the motivating graph described in Section 1 would be an instance of such a graph if an edge occurs between two users iff the two users are from the same city and have different genders.

Definition 1 (Community Overlap Threshold (COT) Graph)**.** *An unweighted, undirected graph whose edges are determined by an overlapping clustering and a "thresholding" integer $t \in \mathbb{Z}$ as follows: for each vertex i, there are two latent binary vectors $\boldsymbol{b}_i \in \{0,1\}^{k_b}$ and $\boldsymbol{c}_i \in \{0,1\}^{k_c}$, and there is an edge between vertices i and j iff $\boldsymbol{b}_i \cdot \boldsymbol{b}_j - \boldsymbol{c}_i \cdot \boldsymbol{c}_j \geq t$.*

Theorem A.1 (Compact Representation of COT Graphs)**.** *Suppose $\boldsymbol{A}$ is the adjacency matrix of a COT graph on n nodes with latent vectors $\boldsymbol{b}_i \in \{0,1\}^{k_b}$ and $\boldsymbol{c}_i \in \{0,1\}^{k_c}$ for $i \in \{1, 2, \ldots, n\}$. Let $k = k_b + k_c$. Then, for any $\epsilon > 0$, there exist $\boldsymbol{V} \in [0,1]^{n \times (k+1)}$ and diagonal $\boldsymbol{W} \in \mathbb{R}^{(k+1)\times(k+1)}$ such that $\left\|\sigma(\boldsymbol{V}\boldsymbol{W}\boldsymbol{V}^\top) - \boldsymbol{A}\right\|_F < \epsilon$.*

Proof. Let t be the thresholding integer of the graph, and let the rows of $\boldsymbol{B} \in \{0,1\}^{n \times k_b}$ and $\boldsymbol{C} \in \{0,1\}^{n \times k_c}$ contain the vectors $\boldsymbol{b}$ and $\boldsymbol{c}$ of all nodes. Via Equation 2, we can find $\boldsymbol{V}^* \in [0,1]^{n \times k}$ and diagonal $\boldsymbol{W}^* \in \mathbb{R}^{k \times k}$ such that $\boldsymbol{V}^*\boldsymbol{W}^*\boldsymbol{V}^{*\top} = \boldsymbol{B}\boldsymbol{B}^\top - \boldsymbol{C}\boldsymbol{C}^\top$. Now let

$$\boldsymbol{V} = \begin{pmatrix} \boldsymbol{V}^* & \mathbf{1} \end{pmatrix} \qquad \boldsymbol{W} = \begin{pmatrix} \boldsymbol{W}^* & 0 \\ 0 & \frac{1}{2} - t \end{pmatrix}.$$

Then $(\boldsymbol{V}\boldsymbol{W}\boldsymbol{V}^\top)_{ij} = \boldsymbol{b}_i \cdot \boldsymbol{b}_j - \boldsymbol{c}_i \cdot \boldsymbol{c}_j + \frac{1}{2} - t$. Hence $(\boldsymbol{V}\boldsymbol{W}\boldsymbol{V}^\top)_{ij} > 0$ iff $\boldsymbol{b}_i \cdot \boldsymbol{b}_j - \boldsymbol{c}_i \cdot \boldsymbol{c}_j > t - \frac{1}{2}$, which is true iff $\boldsymbol{A}_{ij} = 1$ by the assumption on the graph. Similarly, $(\boldsymbol{V}\boldsymbol{W}\boldsymbol{V}^\top)_{ij} < 0$ iff $\boldsymbol{A}_{ij} = 0$. It follows that

$$\lim_{s\to\infty} \sigma\left(\boldsymbol{V}(s\boldsymbol{W})\boldsymbol{V}^\top\right) = \lim_{s\to\infty} \sigma\left(s\boldsymbol{V}\boldsymbol{W}\boldsymbol{V}^\top\right) = \boldsymbol{A}. \qquad \blacksquare$$

A.2 DATASET DESCRIPTIONS

As stated in Section 5, we now briefly describe the five real-world datasets. Statistics for these datasets are given in Table 1.

Table 1: Statistics of datasets used in our experiments. As in Sun et al. (2019), for YOUTUBE and AMAZON, we take only nodes which participate in at least one of the largest 5 ground-truth communities. Note that degeneracy is an upper bound on arboricity.

Name	Reference	Nodes	Edges	Labels	Max Degree	Degeneracy
BLOG	Tang & Liu (2009)	10,312	333,983	39	3,992	114
YOUTUBE	Yang & Leskovec (2015)	5,346	24,121	5	628	19
POS	Qiu et al. (2018)	4,777	92,406	40	3,644	49
PPI	Breitkreutz et al. (2007)	3,852	76,546	50	593	29
AMAZON	Yang & Leskovec (2015)	794	2,109	5	29	6

BLOG is a social network of relationships between online bloggers; the node labels represent interests of the bloggers. Similarly, **YOUTUBE** is a social network of YouTube users, and the labels represent groups that the users joined.

POS is a word co-occurrence network: nodes represent words, and there are edges between words which are frequently adjacent in a section of the Wikipedia corpus. Each node label represents the part-of-speech of the word. **PPI** is a subgraph of the protein-protein interaction network for Homo Sapiens. Labels represent biological states. Finally, **AMAZON** is a co-purchasing network: nodes represent products, and there are edges between products which are frequently purchased together. Labels represent categories of products.

While social networks like the former two in this list are generally dominated by homophily (McPherson et al., 2001), the latter three should exhibit significant heterophily. For co-purchasing networks like AMAZON, depending on the product, two of the same kind of product are generally not co-purchased, e.g., Pepsi and Coke, as discussed in Peysakhovich & Bottou (2021). Though less intuitively accessible, there is also prior discussion of disassortativity in word adjacencies (Foster et al., 2010; Zweig, 2016), as well as in PPI networks (Newman, 2002; Hase et al., 2010).