GRAPH NEURAL NETWORKS FOR COMMUNITY DETECTION ON SPARSE GRAPHS

Luana Ruiz*, Ningyuan (Teresa) Huang†, Soledad Villar†

ABSTRACT

Spectral methods provide consistent estimators for community detection in dense graphs. However, their performance deteriorates as the graphs become sparser. In this work we consider a random graph model that can produce graphs at different levels of sparsity, and we show that graph neural networks can outperform spectral methods on sparse graphs. We illustrate the results with numerical examples in both synthetic and real graphs.

***Index Terms*—** Graph neural networks, community detection, spectral embedding

1. INTRODUCTION

Community detection is a fundamental problem in network science and highly relevant to graph signal processing [1, 2, 3]. Community in a graph refers to a group of nodes that are similar in terms of their connectivity structure and their attributes. Detecting communities reveals important graph structures which can be exploited in a variety of applications, including human neuroimaging [4], network protocol design [5], and social networks [6]. Numerous approaches have been proposed for community detection, including graph neural networks (GNNs) [7]. See, e.g., [8, 9] for a survey and references therein.

In this paper, we focus on supervised community detection with GNNs. We compare GNNs with spectral embeddings, a class of established statistical methods. Spectral embeddings (SEs) have nice theoretical guarantees in random graph models, but can be computationally intensive in large graphs and brittle in sparse graphs [10]. On the other hand, GNNs are deep convolutional architectures for graph data [11, 12] enjoying desirable mathematical properties such as stability [13] and transferability [14], and showing remarkable empirical performance in a variety of problems on large-scale, sparse graphs [15, 16, 17].

Motivated by these observations, we seek out to understand the theoretical underpinnings of the differences in behavior observed between GNNs and SEs in community detection on dense and sparse graphs. To this end, we propose a random graph model that can generate graphs with variable sparsity (Def. 2). We then use it to prove that SEs degrade with graph sparsity (Thm. 1), and to explain why GNNs perform consistently well in sparse graphs (Thm. 2). These findings are further demonstrated empirically through numerical experiments on both synthetic and real-world graphs (Sec. 4).

*LR is with the Simons-Berkeley Inst. and is supported by a Simons Research Fellowship.. †NH and SV are with the AMS Dept. and MINDS at Johns Hopkins University. SV is partially supported by ONR N00014-22-1-2126, NSF CISE 2212457, an AI2AI Amazon research award, and the NSF–Simons Research Collaboration on the Mathematical and Scientific Foundations of Deep Learning (MoDL) (NSF DMS 2031985).

2. PRELIMINARY DEFINITIONS

A graph $\mathbf{G}$ is a triplet $\mathbf{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ where $\mathcal{V} = \{1, \dots, N\}$ is the node set, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ the edge set, and $\mathcal{W} : \mathcal{E} \to \mathbb{R}$ a function assigning edge weights. We focus on unweighted, undirected and connected graphs $\mathbf{G}$, so that $\mathcal{W} : \mathcal{E} \to \{0, 1\}$, $\mathcal{W}(i, j) = \mathcal{W}(j, i)$ for all i, j and there is a single connected component. We represent the graph $\mathbf{G}$ by its adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$, defined as $[\mathbf{A}]_{ij} = \mathcal{W}(i, j)$ if $(i, j) \in \mathcal{E}$ and 0 otherwise. Since $\mathbf{A}$ is symmetric, it can be diagonalized as $\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top$. The diagonal elements of $\mathbf{\Lambda}$ are the eigenvalues $\lambda_i \in \mathbb{R}$, $|\lambda_1| \geq \dots \geq |\lambda_N|$, and the columns of $\mathbf{V}$ the corresponding eigenvectors $\mathbf{v}_i$, $1 \leq i \leq N$.

We assume that the nodes of $\mathbf{G}$ can carry data, which is represented in the form of *graph signals* [3, 18]. A graph signal is a vector $\mathbf{x} \in \mathbb{R}^N$ where $[\mathbf{x}]_i$ is the value of the signal of the node i. More generally, graphs can also carry D-dimensional signals $\mathbf{X} \in \mathbb{R}^{N \times D}$, where each column of $\mathbf{X}$, denoted $\mathbf{x}^d$, is a *node feature*.

Community detection on $\mathbf{G}$ consists of clustering nodes $i \in \mathcal{V}$ into K communities. The goal of community detection is thus to obtain a graph signal $\mathbf{Y} \in [0, 1]^{N \times K}$ where each row $[\mathbf{Y}]_{i\cdot}$ represents the *community assignment* of node i (potentially overlapping [19]). In this paper, we assume non-overlapping communities, so that $[\mathbf{Y}]_{i\cdot} = \texttt{one-hot}(k)$ (i.e., $[\mathbf{Y}]_{ij} = 1$ for $j = k$ and 0 for $j \neq k$) implies that node i is in community k.

There are many variants of community detection [10]. For example, the number of communities K may or may not be predefined [20], and the problem can be solved in an unsupervised or supervised manner [21]. In this paper, we assume that K is given and solve the problem with supervision. Formally, given a graph $\mathbf{G}$ and a signal $\mathbf{X}$, and a true community assignment matrix $\mathbf{Y}$, we fix a training set consisting of a subset $\mathcal{T} = \{i_1, \dots, i_M\} \subset \mathcal{V}$ of the graph nodes. This training set is used to define a node selection matrix $\mathbf{M}_\mathcal{T} \in \{0, 1\}^{M \times N}$ where $[\mathbf{M}_\mathcal{T}]_{ij} = 1$ only for $i = m$, $j = i_m$, and the masked input signal $\mathbf{X}_\mathcal{T} \in \mathbb{R}^{N \times D}$ where $[\mathbf{X}_\mathcal{T}]_{i\cdot} = [\mathbf{X}]_{i\cdot}$ for $i \in \mathcal{T}$ and 0 otherwise. We then use $\mathcal{T}$ to solve the following optimization problem

$$\min_f \ell(\mathbf{M}_\mathcal{T}\mathbf{Y}, \mathbf{M}_\mathcal{T} f(\mathbf{A}, \mathbf{X}_\mathcal{T})) \tag{1}$$

where $\ell : \mathbb{R}^{M \times K} \times \mathbb{R}^{M \times K} \to \mathbb{R}$ is a loss and $f : \mathbb{R}^{N \times N} \times \mathbb{R}^{N \times D} \to \mathbb{R}^{N \times K}$ is a parametric function.

Typically, the function f is parametrized as

$$f = c \circ \phi \tag{2}$$

where c is a classifier and ϕ is an embedding. We will consider the case where the embedding is obtained via SEs in Sec. 2.1, and via GNNs in Sec. 2.2.

2.1. Stochastic Block Model and Spectral Embeddings

The canonical statistical model for graphs with communities is the stochastic block model (SBM).

Definition 1 (Stochastic Block Model)**.** A SBM graph with K communities is defined as a graph $\mathbf{G}$ with adjacency matrix $\mathbf{A} \in \{0,1\}^{N\times N}$ given by

$$\mathbf{A} \sim \text{Ber}(\mathbf{P}),\ \ \mathbf{P} = \mathbf{Y}\mathbf{B}\mathbf{Y}^\top$$

where $\mathbf{Y} \in \{0,1\}^{N\times K}$ is the community assignment matrix $\mathbf{Y}_{i\cdot} = \texttt{one-hot}(k)$, and $\mathbf{B} \in [0,1]^{K\times K}$ is a full-rank matrix representing the block connection probability.

Spectral methods for community detection are inspired by the spectral decomposition of the SBM. Consider for example the case where $K = 2$, $\mathbf{B} = [p\ q; q\ p]$, $p \neq q$, and the communities are balanced, i.e., N is even and both communities have size $N/2$. Relabeling $\mathcal{V}$ so that the first $N/2$ nodes belong to the first community and the remaining $N/2$ to the second, we see that the eigenvectors of $\mathbb{E}\mathbf{A} \equiv \mathbf{P}$, the expected adjacency, are given by

$$[\mathbf{v}_1(\mathbb{E}\mathbf{A})]_i = \frac{1}{\sqrt{N}},\ [\mathbf{v}_2(\mathbb{E}\mathbf{A})]_i = \begin{cases} -1/\sqrt{N},\ i \leq N/2 \\ +1/\sqrt{N},\ i > N/2. \end{cases} \tag{3}$$

For a graph $\mathbf{G}$ sampled from this model, with sufficiently large N and mild assumptions on p, q, we can thus expect the eigenvector $\mathbf{v}_2(\mathbf{A})$ to provide a good estimate of its community structure, i.e., $\mathbf{v}_k(\mathbf{A}) \approx \mathbf{v}_k(\mathbb{E}\mathbf{A}), k \in \{1,2\}$.

Real-world graphs $\mathbf{G}$ have more intricate sparsity patterns than the SBM, but it is reasonable to assume that if the graph $\mathbf{G}$ has two balanced communities, for some permutation of the nodes its adjacency matrix $\mathbf{A}$ can be approximately written as $\mathbf{A} = \mathbf{A}_{\text{SBM}} + \mathbf{E}$, where $\mathbf{A}_{\text{SBM}}$ is as in Def. 1 and $\mathbf{E}$ can be seen as a perturbation satisfying $\|\mathbf{E}\|_2 < \|\mathbf{A}_{\text{SBM}}\|_2$. As such, the first two eigenvectors of $\mathbf{A}$ still "embed" community information. More generally, in graphs $\mathbf{G}$ with $K > 2$ balanced communities, the community information is "embedded" in the first K eigenvectors. Based on this observation, the order-K *spectral embedding* (SE) of a graph $\mathbf{G}$ is defined as

$$\phi_{\text{SE}}(\mathbf{A}) = [\mathbf{v}_1\ \mathbf{v}_2\ \ldots\ \mathbf{v}_{K-1}\ \mathbf{v}_K] = \mathbf{V}_K, \tag{4}$$

i.e., as the concatenation of the first K eigenvectors of $\mathbf{A}$. Variants of SE tailored for sparse graphs propose replacing $\mathbf{A}$ with other graph operators, such as the normalized adjacency matrix $\tilde{\mathbf{A}} := \mathbf{D}^{-0.5}\mathbf{A}\mathbf{D}^{-0.5}$ where $\mathbf{D}$ is the degree matrix [22], the non-backtracking operator [10], etc.

Note that ϕ_{SE} is nonparametric; it can be obtained directly from the graph without node label supervision. When we use spectral embeddings in (2), the only parameters that are learned are those of the classifier c. E.g., choosing a linear classifier yields a simple parameterization of f as $f(\mathbf{A}) = c \circ \phi_{\text{SE}}(\mathbf{A}) = \texttt{softmax}(\mathbf{V}_K\mathbf{C})$ where $\mathbf{C} \in \mathbb{R}^{K\times K}$ is learned. More generally, it is possible to use embeddings $\phi_{\text{SE}}(\mathbf{A}) = \mathbf{V}_{\tilde{K}}$ with $\tilde{K} > K$, i.e., with a larger number of eigenvectors than that of communities, in which case $\mathbf{C} \in \mathbb{R}^{\tilde{K}\times K}$.

An important observation to make is that ϕ_{SE} (and so f) do not need to depend on $\mathbf{X}$, but if such node features are available, they can be incorporated into the spectral embedding in different ways, e.g., [23, 24, 25, 26, 27]. We consider an approach similar to [25], by first embedding the node feature covariance and concatenating it with the spectral embedding. More precisely, let $\mathbf{V}'_\kappa$ be the first κ eigenvectors of the covariance matrix $\mathbf{X}\mathbf{X}^\top$, then the *feature-aware* spectral embedding is defined as

$$\phi_{\text{SE}}(\mathbf{A}; \mathbf{X}) = [\mathbf{V}_K\ \ \mathbf{V}'_\kappa]. \tag{5}$$

2.2. Graph Neural Networks

Given a graph $\mathbf{G}$ with adjacency matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$ and a graph signal $\mathbf{x} \in \mathbb{R}^N$, a graph convolution (or filter) is given by [28]

$$\mathbf{u} = \sum_{k=0}^{K-1} h_k \mathbf{A}^k \mathbf{x} \tag{6}$$

where $h_0, \ldots, h_{K-1}$ are the filter coefficients or taps. More generally, if $\mathbf{X} \in \mathbb{R}^{N\times D}$ and $\mathbf{U} \in \mathbb{R}^{N\times G}$ have D and G features respectively, we write

$$\mathbf{U} = \sum_{k=0}^{K-1} \mathbf{A}^k \mathbf{X} \mathbf{H}_k \tag{7}$$

where the filter parameters are now collected in the matrices $\mathbf{H}_0, \ldots, \mathbf{H}_{K-1} \in \mathbb{R}^{D\times G}$.

Graph neural networks (GNNs) are deep convolutional architectures where each layer composes a graph convolution (7) and a pointwise nonlinearity $[\sigma(\mathbf{U})]_{ij} = \sigma([\mathbf{U}]_{ij})$, e.g., the ReLU or the sigmoid. The ℓth layer of a GNN can thus be written as

$$\mathbf{X}_\ell = \sigma\left(\sum_{k=0}^{K-1} \mathbf{A}^k \mathbf{X}_{\ell-1} \mathbf{H}_{\ell k}\right) \tag{8}$$

where $\mathbf{X}_{\ell-1} \in \mathbb{R}^{N\times F_{\ell-1}}$ and $\mathbf{X}_\ell \in \mathbb{R}^{N\times F_\ell}$ are the input and output to this layer with $F_{\ell-1}$ and F_ℓ features each. If the GNN has L layers, its input and output are $\mathbf{X}_0 = \mathbf{X} \in \mathbb{R}^{N\times F_0}$ and $\mathbf{X}_L \in \mathbb{R}^{N\times F_L}$.

The GNN in (8) may be used to parametrize ϕ in (2), in which case we define the *GNN embedding*

$$\phi_{\text{GNN}}(\mathbf{A}, \mathbf{X}) = \mathbf{X}_L. \tag{9}$$

Note that, unlike the spectral embedding (4), (9) is parametric on $\{\mathbf{H}_{\ell k}\}_{\ell,k}$ and always needs an input signal $\mathbf{X}$ (if an input signal is not available, $\mathbf{X}$ may be a random signal, for example). A typical parametrization of f for GNN embeddings is $f(\mathbf{A}, \mathbf{X}) = c \circ \phi_{\text{GNN}}(\mathbf{A}, \mathbf{X}) = \texttt{softmax}(\mathbf{X}_L\mathbf{C})$ where $\mathbf{C} \in \mathbb{R}^{F_L\times K}$ is a linear classifier over F_L node features. This is equivalent to a $L+1$-layer GNN with $K = 1$ and softmax nonlinearity in the last layer.

3. MAIN RESULTS

In the following, we introduce a random graph model for both dense and sparse graphs. We use this model to prove a result that helps explain the limitations of spectral embeddings on sparse graphs. We then show that under mild assumptions on both the graph and the input signal, GNNs give access to entire spectrum, and thus can learn embeddings that are more expressive than spectral embeddings.

3.1. A Graph Model for Dense and Sparse Graphs

Def. 2 introduces a random graph model allowing to model graphs with varying levels of sparsity according to a sparsity parameter γ.

Definition 2 (Dense-Sparse Graph Model (DSGM))**.** A DSGM graph with kernel $\mathbf{W}$ and sparsity parameter γ is defined as a graph $\mathbf{G}$ with adjacency matrix $\mathbf{A} \in \{0,1\}^{N\times N}$ given by

$$[\mathbf{A}]_{ij} = [\mathbf{A}]_{ji} \sim \text{Ber}(\mathbf{W}(u_i, u_j)),\ u_i = \begin{cases} u_{i-1} + \gamma,\ 2 \leq i \leq N \\ -\lfloor \frac{n}{2} \rfloor \gamma + \frac{\gamma}{2},\ i = 1 \end{cases}$$

where $\mathbf{W} : \mathbb{R}^2 \to [0,1]$ is symmetric, $\|\mathbf{W}\|_{L^2} < \infty$, and $\gamma > 0$.

This model allows sampling both dense and sparse graphs because, since $\mathbf{W}$ has vanishing tails (or can be mapped to a kernel that does by some measure-preserving transformation), for a fixed N the graph is sparser for larger γ.

The kernel $\mathbf{W}$ defines a self-adjoint Hilbert Schmidt operator. Hence, it has a real spectrum given by

$$\int_{-\infty}^{\infty} \mathbf{W}(u,v)\varphi_i(u)du = \lambda_i \varphi_i(v) \tag{10}$$

where the eigenvalues λ_i are countable and the eigenfunctions φ_i form an orthonormal basis of L^2. By convention, the eigenvalues are ordered as $|\lambda_1| \geq |\lambda_2| \geq \ldots$. Moreover, $|\lambda_i| \leq \infty$ for all i, and $\lambda_i \to 0$ as $i \to \infty$ with zero being the only accumulation point.

We further introduce the notion of a kernel induced by a graph, which will be useful in future derivations. For $N \geq 2$, the kernel induced by the graph $\mathbf{G}_N$ with adjacency $\mathbf{A}_N$ and sparsity parameter γ is defined as

$$\mathbf{W}_N(u,v) = \sum_{i=1}^{N-1}\sum_{j=1}^{N-1} [\mathbf{A}_N]_{ij}\mathbb{I}(u \in I_i)\mathbb{I}(v \in I_j) \tag{11}$$

where $I_i = [u_i, u_{i+1})$ for $1 \leq i \leq N-2$, $I_{N-1} = [u_{N-1}, u_N]$, and u_i is as in Def. 2.

3.2. Limitations of Spectral Embeddings

To discuss community detection on graphs sampled from a DSGM (Def. 2), we assume that the kernel $\mathbf{W}$ exhibits community structure. For simplicity, we focus on 2 communities but the discussions can be easily extended to K communities. Inspired by the degree-corrected SBM [29, 30], in Def. 3 we introduce the degree-corrected stochastic block kernel (SBK) as the canonical kernel for DSGMs with 2 balanced communities. This model is suitable to model sparse graphs and well-studied in the spectral embedding literature [30, 22]. To ensure that models based on these kernels are valid DSGMs, we restrict attention to finite-energy degree functions θ.

Definition 3 (Degree-Corrected SBK)**.** The degree-corrected SBK with 2 communities is given by

$$\mathbf{W}(u,v) = \begin{cases} \theta(u)\,\theta(v)\,p, & uv \geq 0 \\ \theta(u)\,\theta(v)\,q, & uv < 0 \end{cases}$$

where $\theta : \mathbb{R} \to [0,1]$, $\theta \in L^2$, is the degree function. The true community assignment is $Y(u) = [1\ 0]\mathbb{I}(u \geq 0) + [0\ 1]\mathbb{I}(u < 0)$, which is independent of θ.

It is not difficult to see that the first 2 eigenfunctions of $\mathbf{W}$ in Def. 3 reveal the community structure [1]. For graphs $\mathbf{G}_N$ sampled as in Def. 2 from the DSGM with degree-corrected kernel as in Def. 3, the true community assignment is given by $[\mathbf{Y}]_{i\cdot} = Y(u_i)$ for $1 \leq i \leq N$. As such, the quality of the estimate of the community assignment given by the first 2 (or, more generally, the first K) eigenvectors of $\mathbf{G}_N$ will depend on both (i) how close the eigenvalues $\lambda_k(\mathbf{G}_N)$ are to the kernel eigenvalues $\lambda_k(\mathbf{W})$ (as this can affect their ordering) and (ii) how close the eigenvectors $\mathbf{v}_k$ are to the eigenfunctions φ_k. These differences are upper bounded by Thm. 1.

Theorem 1 (Eigenvalue and eigenvector concentration)**.** Let $\mathbf{G}_N$ be a graph sampled from the DSGM in Def. 2, where N satisfies [14, Ass. AS4]. Let $c \leq \lfloor N/2 \rfloor \gamma - \gamma/2$, and assume that:

1. $\mathbf{W}$ is A_w-Lipschitz in $[-c,c] \times [-c,c]$ (see [14, Ass. AS2])
2. $\int_{|v|\geq c}\int_{|u|\geq c} \mathbf{W}(u,v)dudv < \epsilon(c)$.

Then, with probability at least $1-\chi$, the difference between the kth eigenvalue of $\mathbf{G}_N$ and $\mathbf{W}$, $1 \leq k \leq K$, is bounded by

$$\begin{aligned} |\lambda_k(\mathbf{W}_N) - \lambda_k(\mathbf{W})| &\leq 4A_w c\gamma + \beta(\chi,N)N^{-1} + \epsilon(c) \\ &\leq 2A_w N\gamma^2 + \beta(\chi,N)N^{-1} + \epsilon(c) \end{aligned}$$

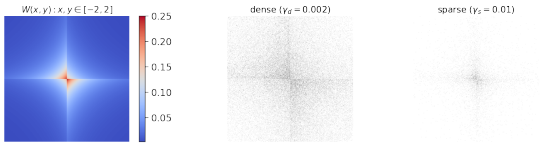
and the difference between their kth eigenvectors by

$$\|\varphi_k(\mathbf{W}_N) - \varphi_k(\mathbf{W})\| \leq \frac{\pi}{2\delta_k}\left(4A_w c\gamma + \beta(\chi,N)N^{-1} + \epsilon(c)\right)$$

where $\mathbf{W}_N$ is the kernel induced by $\mathbf{G}_N$ (13)[2], $\delta_k = \min_i \{|\lambda_k(\mathbf{W}) - \lambda_i(\mathbf{W}_N)|, |\lambda_k(\mathbf{W}_N) - \lambda_i(\mathbf{W})|\}$ and $\beta(\chi,N)$ is sublinear in N and as in [14, Def. 7].

Proof. Refer to the extended version in this repository. □

This theorem shows that the differences between the eigenvalues and eigenvectors of the graph and the underlying random graph model are upper bounded by terms that increase with γ. Consider a dense graph $\mathbf{G}_N^d$ and a sparse graph $\mathbf{G}_N^s$ sampled from DSGMs with same kernel $\mathbf{W}$ but different sparsity parameters $\gamma_d \ll \gamma_s$. If N and c are large enough for the term depending on $4A_w c\gamma$ to dominate the bound in the dense case, the bound on the difference between eigenvalues and eigenvectors in the sparse case is much larger than in the dense case. In the context of community detection, this can be interpreted to mean that, since $\varphi_k(\mathbf{W}_N^d)$ is close to $\varphi_k(\mathbf{W})$ for dense graphs, some linear combination of the eigenvectors $\mathbf{v}_k(\mathbf{G}_N^d)$ provides a good estimate of the true community assignment $\mathbf{Y}$. This is not true for the eigenvectors $\mathbf{v}_k(\mathbf{G}_N^s)$ of the sparse graph, since $\varphi_k(\mathbf{W}_N^s)$ is further away from $\varphi_k(\mathbf{W})$. Another way to think about this is that on dense graphs most of the "community information" is on the first K eigenvectors. On sparse graphs, it is more spread throughout the spectrum. This implies that while spectral embeddings may be effective for community detection on dense graphs, they are less likely to be effective in sparse graphs. We further demonstrate this empirically in Sec. 4.



(a) $\mathbf{W}$ (b) Dense $\mathbf{G}_N^d$ (c) Sparse $\mathbf{G}_N^s$

Fig. 1: Kernel $\mathbf{W} : \mathbb{R}^2 \to [0,1]$ visualized in $[-2,2]^2$ and sampled graphs with different sparsity levels γ.

3.3. Graph Neural Networks for Community Detection

In sparse graphs, GNN embeddings are a better option than spectral embeddings because, provided that the input signal $\mathbf{X}_{\mathcal{T}}$ in (1) is not orthogonal to any of the graph's eigenvectors, GNNs "have access" to the entire spectrum. Moreover, if the true community assignment signal is $\mathbf{Y}$, a GNN can always represent $\mathbf{Y}$ with $K \leq N$ in (8).

[1] $\varphi_1(u) = \theta(u)/C, \varphi_2(u) = \big(-\theta(u)\mathbb{I}(u<0) + \theta(u)\mathbb{I}(u \geq 0)\big)/C$, where $C \coloneqq \int \theta(u)du$.

[2] See [31, Lemma 2] for the relationship between $\lambda_k(\mathbf{G}_N)$, $\mathbf{v}_k(\mathbf{G}_N)$ and $\lambda_k(\mathbf{W}_N)$, $\varphi_k(\mathbf{W}_N)$.

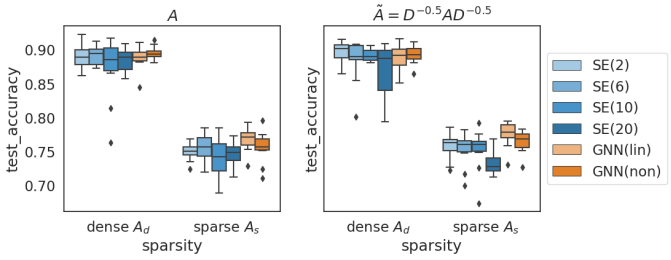
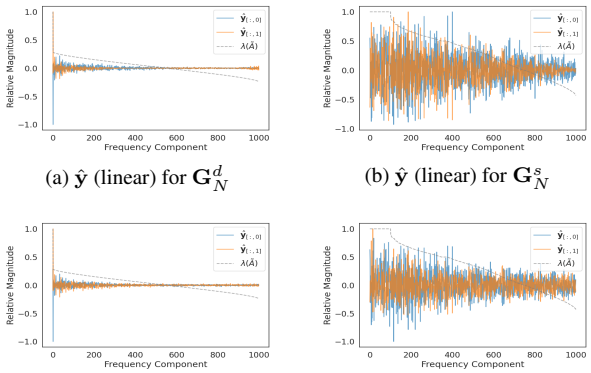


Fig. 2: Test accuracy for different sparsity levels of the sampled graphs. GNNs perform better than SEs in sparse graphs for both operators $\mathbf{A}, \tilde{\mathbf{A}}$.



(a) $\hat{\mathbf{y}}$ (linear) for $\mathbf{G}_N^d$ (b) $\hat{\mathbf{y}}$ (linear) for $\mathbf{G}_N^s$

(c) $\hat{\mathbf{y}}$ (nonlinear) for $\mathbf{G}_N^d$ (d) $\hat{\mathbf{y}}$ (nonlinear) for $\mathbf{G}_N^s$

Fig. 3: Frequency responses $\hat{\mathbf{y}}$ of GNNs using $\tilde{\mathbf{A}}$ on $\mathbf{G}_N^d$ and $\mathbf{G}_N^s$. In the dense case (left), although the optimal frequency response is a step-function on the first two components, GNNs spread energies on the remaining components, adding noise; In the sparse case (right), the community information spreads widely across the spectrum and thus GNNs outperform spectral embedding. Nonlinear GNNs (bottom) leverage the spectrum more uniformly than linear convolutions (top). Eigenvalues of $\tilde{\mathbf{A}}$ (dashed) are sorted in decreasing order.

These claims are formally stated for the simple graph convolution (6) in Thm. 2. They can be readily extended to multi-feature graph convolutions (7) and GNNs (8) where the nonlinearity σ preserves the sign (e.g., the hyperbolic tangent).

Theorem 2 (Expressive power of graph convolution)**.** Let $\mathbf{G}$ be a symmetric graph with full-rank adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ diagonalizable as $\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top$ where all eigenvalues have multiplicity one. Let $\mathbf{x} \in \mathbb{R}^N$ be an input signal satisfying $[\mathbf{V}^\top \mathbf{x}]_i \neq 0$ for $1 \leq i \leq N$. Consider the graph convolution $\hat{\mathbf{y}} = \sum_{k=0}^{K-1} h_k \mathbf{A}^k \mathbf{x}$ (6). Then, the following hold:

1. For all $K \geq 1$, there exist $h_0, \ldots, h_{K-1} \in \mathbb{R}$ such that $\hat{\mathbf{y}}$ satisfies $[\mathbf{V}^\top \hat{\mathbf{y}}]_i \neq 0$ for every i.
2. Let $\mathbf{y} \in \mathbb{R}^N$ be a target signal. There exist $K \leq N$ coefficients $h_0, \ldots, h_{K-1} \in \mathbb{R}$ for which $\hat{\mathbf{y}}$ satisfies $\hat{\mathbf{y}} = \mathbf{y}$.

Proof. Refer to the extended version in this repository. □

Note that the assumptions of Thm. 2 are not too restrictive; most real-world graphs are full rank, and even a random signal $\mathbf{x} \in \mathbb{R}^N$—which may be used as the input in (9) when $\mathbf{x}$ is not given—satisfies $[\mathbf{V}^\top \mathbf{x}]_i \neq 0$ with high probability. It is also worth pointing out that while $K \leq N$ is necessary to *exactly represent* $\mathbf{y}$, in practice small K is often enough to obtain good *approximations* of the true community assignment as illustrated in Sec. 4. This is another reason why in practical, large graph settings, GNN embeddings are advantageous w.r.t. spectral embeddings: a small number of matrix-vector multiplications requires less computations than calculating a number of eigenvectors at least as large as the number of communities.

4. EXPERIMENTS

In what follows, we conduct simulations on synthetic graphs sampled from a DSGM (Section 4.1) and real-world graphs (Section 4.2). For completeness, we consider graph operators $\mathbf{A}, \tilde{\mathbf{A}}$. Our empirical results validate our theoretical analysis and show that GNNs outperform spectral embedding for community detection in sparse graphs.[3]

4.1. Experiments on Synthetic Graphs

Setup. We consider the following kernel

$$\mathbf{W}(u,v) = \begin{cases} \frac{p}{(|u|+1)^2(|v|+1)^2} & uv \geq 0 \\ \frac{q}{(|u|+1)^2(|v|+1)^2} & uv < 0. \end{cases} \quad (12)$$

[3]All the simulations and code are available in this repository.

The graphs $\mathbf{G}$ are sampled from the DSGM with kernel $\mathbf{W}$ above following Def. 2, with $N = 1000$ and different choices of density parameter $\gamma_d = 0.002, \gamma_s = 0.01$ as illustrated in Fig. 1. The node features $\mathbf{X}$ are sampled from a mixture of two Gaussians in $\mathbb{R}^2$ where $\boldsymbol{\mu}_0 = -\boldsymbol{\mu}_1 = [1,1], \mathbf{\Sigma}_0 = \mathbf{\Sigma}_1 = \mathbf{I}/4$. For each tuple $(\mathbf{G}, \mathbf{X})$, we randomly split the nodes in each community by $50/50$ to create the training and test sets. We compare spectral embeddings with various choices of K against GNNs.

Results. Fig. 2 shows that spectral embedding with $K = 2$ outperforms GNNs in dense graphs while GNNs are more competitive in sparse graphs. Fig. 3 depicts the frequency response $\hat{\mathbf{y}}$ from the trained GNN model using $\tilde{\mathbf{A}}$: (a) shows that, in the dense graph, GNNs indeed attend to frequency components other than the first two eigenvectors, which increase the noise/variance of the embedding and thus degrades the downstream classification performance, confirming the discussion in Thm. 2; (b) shows that, in the sparse graph, GNNs increasingly attend to higher-frequency components, which are useful since they may also encode community information; spectral embeddings exhibit higher variance, and can benefit from choosing suitably larger embedding dimension.

4.2. Experiments on Real-World Graphs

Setup. We consider the Wikipedia webpage network Chameleon, a heterophilous benchmark graph with 5 communities introduced in [32]. We treat the original Chameleon network ($|\mathcal{V}| = 2277$, average degree 13.8) as the dense baseline, and randomly drop a fraction of its edges to obtain the sparse(r) graphs. We then evaluate GNNs and spectral embedding in the original and sparsified graphs. For each sparsity level, we randomly generate 10 sparsified graphs.

Results. Table 1 shows that GNNs and spectral embeddings both perform well in the original graph. Yet, in the sparsified graphs ("Drop(20)", "Drop(70)"), performance degradation in GNNs is smaller than spectral embeddings. Moreover, in sparsified graphs, spectral embeddings with large K are numerically unstable and computationally intensive due to the presence of many small eigen-

values. These findings show that GNNs can detect communities more accurately and efficiently than spectral methods in sparse graphs.

Table 1: Test accuracy on Chameleon graphs, reported as `mean(±stderr)` across 10 data splits and 10 sparsified subgraphs.

Graph	Operator	SE(150)	SE(200)	GNN(lin)	GNN(non)
Original	$\mathbf{A}$	$\mathbf{57.29 \pm 0.69}$	56.97 ± 0.59	56.27 ± 0.69	54.38 ± 0.97
	$\tilde{\mathbf{A}}$	52.70 ± 0.36	53.84 ± 0.43	55.60 ± 0.70	$\mathbf{55.90 \pm 0.73}$
Drop(20)	$\mathbf{A}$	53.20 ± 0.21	53.30 ± 0.22	$\mathbf{53.91 \pm 0.25}$	52.69 ± 0.29
	$\tilde{\mathbf{A}}$	49.42 ± 0.21	51.53 ± 0.19	54.45 ± 0.21	$\mathbf{54.66 \pm 0.22}$
Drop(70)	$\mathbf{A}$	45.47 ± 0.20	45.12 ± 0.22	$\mathbf{46.21 \pm 0.23}$	45.95 ± 0.24
	$\tilde{\mathbf{A}}$	41.21 ± 0.19	42.51 ± 0.27	50.10 ± 0.19	$\mathbf{50.25 \pm 0.21}$

5. REFERENCES

[1] H-T. Wai, S. Segarra, A. E. Ozdaglar, A. Scaglione, and A. Jadbabaie, "Blind community detection from low-rank excitations of a graph filter," IEEE Transactions on Signal Processing, vol. 68, pp. 436–451, 2020.

[2] M. Navarro and S. Segarra, "Graphon-aided joint estimation of multiple graphs," in ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2022, pp. 5458–5462.

[3] A. Ortega, P. Frossard, J. Kovačević, J. M. F. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," Proc. IEEE, vol. 106, no. 5, pp. 808–828, 2018.

[4] M. Petrovic, R. Liegeois, T. A.W. Bolton, and D. Van De Ville, "Community-aware graph signal processing: Modularity defines new ways of processing graph signals," IEEE Signal Processing Magazine, vol. 37, no. 6, pp. 150–159, 2020.

[5] Z. Lu, X. Sun, Y. Wen, G. Cao, and T. L. Porta, "Algorithms and applications for community detection in weighted networks," IEEE Transactions on Parallel and Distributed Systems, vol. 26, no. 11, pp. 2916–2926, 2015.

[6] P. Bedi and C. Sharma, "Community detection in social networks," Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 6, no. 3, pp. 115–135, 2016.

[7] Zhengdao Chen, Lisha Li, and Joan Bruna, "Supervised community detection with line graph neural networks," in International Conference on Learning Representations, 2018.

[8] M. T. Schaub, J-C. Delvenne, M. Rosvall, and R. Lambiotte, "The many facets of community detection in complex networks," Applied network science, vol. 2, no. 1, pp. 1–13, 2017.

[9] X. Su, S. Xue, F. Liu, J. Wu, J. Yang, C. Zhou, W. Hu, C. Paris, S. Nepal, D. Jin, et al., "A comprehensive survey on community detection with deep learning," IEEE Transactions on Neural Networks and Learning Systems, 2022.

[10] E. Abbe, "Community detection and stochastic block models: recent developments," The Journal of Machine Learning Research, vol. 18, no. 1, pp. 6446–6531, 2017.

[11] L. Ruiz, F. Gama, and A. Ribeiro, "Graph neural networks: Architectures, stability and transferability," Proc. IEEE, vol. 109, no. 5, pp. 660–682, 2021.

[12] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in 5th Int. Conf. Learning Representations, Toulon, France, 24-26 Apr. 2017, Assoc. Comput. Linguistics.

[13] F. Gama, J. Bruna, and A. Ribeiro, "Stability properties of graph neural networks," IEEE Trans. Signal Process., vol. 68, pp. 5680–5695, 2020.

[14] L. Ruiz, L. F. O. Chamon, and A. Ribeiro, "Transferability properties of graph neural networks," arXiv:2112.04629 [eess.SP]. Submitted to IEEE TSP, 2021.

[15] E. Tolstaya, F. Gama, J. Paulos, G. Pappas, V. Kumar, and A. Ribeiro, "Learning decentralized controllers for robot swarms with graph neural networks," in Conf. Robot Learn., Osaka, Japan, 30 Oct.-1 Nov. 2019, International Foundation of Robotics Research.

[16] M. Eisen and A. Ribeiro, "Optimal wireless resource allocation with random edge graph neural networks," IEEE Trans. Signal Process., vol. 68, pp. 2977–2991, 2020.

[17] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: Going beyond euclidean data," arXiv:1611.08097v2 [cs.CV], 2017.

[18] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs," IEEE Trans. Signal Process., vol. 61, pp. 1644–1656, Apr. 2013.

[19] J. Xie, S. Kelley, and B. K. Szymanski, "Overlapping community detection in networks: The state-of-the-art and comparative study," Acm computing surveys (csur), vol. 45, no. 4, pp. 1–35, 2013.

[20] D. S. Choi, P. J. Wolfe, and E. M. Airoldi, "Stochastic blockmodels with a growing number of classes," Biometrika, vol. 99, no. 2, pp. 273–284, 2012.

[21] T. T. Cai, T. Liang, and A. Rakhlin, "Weighted message passing and minimum energy flow for heterogeneous stochastic block models with side information.," J. Mach. Learn. Res., vol. 21, pp. 11–1, 2020.

[22] J. Cape, M. Tang, and C. E. Priebe, "On spectral embedding performance and elucidating network structure in stochastic blockmodel graphs," Network Science, vol. 7, no. 3, pp. 269–291, 2019.

[23] J. Yang, J. McAuley, and J. Leskovec, "Community detection in networks with node attributes," in 2013 IEEE 13th International Conference on Data Mining, 2013, pp. 1151–1156.

[24] N. Binkiewicz, J. T. Vogelstein, and K. Rohe, "Covariate-assisted spectral clustering," Biometrika, vol. 104, no. 2, pp. 361–377, 2017.

[25] J. Arroyo, A. Athreya, J. Cape, G. Chen, C. E. Priebe, and J. T. Vogelstein, "Inference for multiple heterogeneous networks with a common invariant subspace," Journal of Machine Learning Research, vol. 22, no. 142, 2021.

[26] C. Mu, A. Mele, L. Hao, J. Cape, A. Athreya, and C. E. Priebe, "On spectral algorithms for community detection in stochastic blockmodel graphs with vertex covariates," IEEE Transactions on Network Science and Engineering, 2022.

[27] A. Mele, L. Hao, J. Cape, and C. E. Priebe, "Spectral inference for large stochastic blockmodels with nodal covariates," 2019.

[28] J. Du, J. Shi, S. Kar, and J. M. F. Moura, "On graph convolution for graph CNNs," in 2018 IEEE Data Sci. Workshop, Lausanne, Switzerland, 4-6 June 2018, pp. 239–243, IEEE.

[29] B. Karrer and M. E. J. Newman, "Stochastic blockmodels and community structure in networks," Physical review E, vol. 83, no. 1, pp. 016107, 2011.

[30] T. Qin and K. Rohe, "Regularized spectral clustering under the degree-corrected stochastic blockmodel," Advances in neural information processing systems, vol. 26, 2013.

[31] L. Ruiz, L. F. O. Chamon, and A. Ribeiro, "Graphon signal processing," IEEE Trans. Signal Process., vol. 69, pp. 4961–4976, 2021.

[32] B. Rozemberczki, C. Allen, and Rik Sarkar, "Multi-scale attributed node embedding," Journal of Complex Networks, vol. 9, no. 2, pp. cnab014, 2021.

[33] L. Ruiz, L. F. O. Chamon, and A. Ribeiro, "Graphon neural networks and the transferability of graph neural networks," in 34th Neural Inform. Process. Syst., Vancouver, BC (Virtual), 6-12 Dec. 2020, NeurIPS Foundation.

[34] A. Seelmann, "Notes on the sin 2Θ theorem," Integral Equations and Operator Theory, vol. 79, no. 4, pp. 579–597, 2014.

[35] H. Pei, B. Wei, K. C-C. Chang, Y. Lei, and B. Yang, "Geom-gcn: Geometric graph convolutional networks," in International Conference on Learning Representations, 2020.

[36] M. Fey and J. E. Lenssen, "Fast graph representation learning with PyTorch Geometric," in ICLR Workshop on Representation Learning on Graphs and Manifolds, 2019.

6. APPENDIX

6.1. Proof of Theorem 1

The proof of Theorem 1 relies on slight variations of the Courant-Fisher and Davis-Kahan theorems, stated here as Propositions 1 and 2.

Proposition 1 (Variant of Courant-Fisher)**.** Let $\mathbf{W} : [0,1]^2 \to [0,1]$ and $\mathbf{W}' : [0,1]^2 \to [0,1]$ be two graphons with eigenvalues given by $\{\lambda_i(T_{\mathbf{W}})\}_{i\in\mathbb{Z}\setminus\{0\}}$ and $\{\lambda_i(T_{\mathbf{W}'})\}_{i\in\mathbb{Z}\setminus\{0\}}$, ordered according to their sign and in decreasing order of absolute value, where $T_{\mathbf{W}}$ denotes the integral linear operator with kernel $\mathbf{W}$. Then, for all $i \in \mathbb{Z}\setminus\{0\}$, the following inequalities hold

$$|\lambda_i(T_{\mathbf{W}'}) - \lambda_i(T_{\mathbf{W}})| \leq |||T_{\mathbf{W}'-\mathbf{W}}||| \leq \|\mathbf{W}' - \mathbf{W}\| .$$

Proof. See [33, Proposition 4].

Proposition 2 (Variant of Davis-Kahan)**.** Let T and T' be two self-adjoint operators on a separable Hilbert space $\mathcal{H}$ whose spectra are partitioned as $\gamma \cup \Gamma$ and $\omega \cup \Omega$ respectively, with $\gamma \cap \Gamma = \emptyset$ and $\omega \cap \Omega = \emptyset$. If there exists $d > 0$ such that $\min_{x\in\gamma,\, y\in\Omega} |x-y| \geq d$ and $\min_{x\in\omega,\, y\in\Gamma} |x-y| \geq d$, then the spectral projections $E_T(\gamma)$ and $E_{T'}(\omega)$ satisfy

$$|||E_T(\gamma) - E_{T'}(\omega)||| \leq \frac{\pi}{2}\frac{|||T - T'|||}{d}$$

Proof. See [34].

We thus only need to bound $\|\mathbf{W} - \mathbf{W}_N\|$. To do so, define $\overline{\mathbf{W}}_N$ as

$$\overline{\mathbf{W}}_N(u,v) = \sum_{i=1}^{N-1}\sum_{j=1}^{N-1} \mathbf{W}(u_i, \mathbf{u}_j)\mathbb{I}(u \in I_i)\mathbb{I}(v \in I_j) \tag{13}$$

where $I_i = [u_i, u_{i+1})$ for $1 \leq i \leq N-2$, $I_{N-1} = [u_{N-1}, u_N]$, and u_i is as in (2). Using the triangle inequality, we can write

$$\|\mathbf{W} - \mathbf{W}_N\| \leq \|\mathbf{W} - \overline{\mathbf{W}}_N\| + \|\overline{\mathbf{W}}_N - \mathbf{W}_N\|. \tag{14}$$

The norm difference between $\overline{\mathbf{W}}_N$ and $\mathbf{W}_N$ is bounded as $N^{-1}\beta(\chi, N)$ by [14, Proposition 4] and by the fact that $\|\mathbf{W}_N\|_{L_2} = N^{-1}\|\mathbf{A}_N\|_2$ (see [31, Lemma 2]). Let us now derive a bound for $\|\mathbf{W} - \overline{\mathbf{W}}_N\|$.

By definition of the L^2 norm,

$$\begin{aligned}\|\mathbf{W} - \overline{\mathbf{W}}_N\| &= \sqrt{\int_{-\infty}^{\infty} |\mathbf{W}(u,v) - \overline{\mathbf{W}}_N(u,v)|^2 dudv} \\ &\leq \sqrt{\int_{|v|<c}\int_{|u|<c} |\mathbf{W}(u,v) - \overline{\mathbf{W}}_N(u,v)|^2 dudv} \\ &\quad + \sqrt{\int_{|v|\geq c}\int_{|u|\geq c} |\mathbf{W}(u,v) - \overline{\mathbf{W}}_N(u,v)|^2 dudv}\end{aligned} \tag{15}$$

The rightmost term is bounded by $\epsilon(c)$, as $\overline{\mathbf{W}}_N$ is zero outside of $[-c,c]$. Since $\mathbf{W}$ is A_w-Lispchitz in the $[-c,c]$ interval, we can write

$$\begin{aligned}|\mathbf{W}(u,v) - \overline{\mathbf{W}}_N(u,v)| &\leq A_w \max\left(|u-u_i|, |u_{i+1}-u|\right) \\ &\quad + A_w \max\left(|v-u_j|, |u_{j+1}-v|\right) \\ &\leq A_w\gamma + A_w\gamma = 2A_w\gamma\end{aligned}$$

for $u_i \leq u \leq u_{i+1}$, $u_j \leq v \leq u_{j+1}$, where the u_i, u_j are as in Definition 2 for all $1 \leq i,j \leq N$. Therefore, the leftmost term in (15) can be upper bounded as $\sqrt{2c \times 2c \times (2A_w\gamma)^2} = 4A_w\gamma c$, which completes the proof.

6.2. Proof of Theorem 2

Theorem 2.1 is a direct consequence of the fact that the graph convolution is pointwise in the spectral domain. To see this, substitute $\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top$ in (6) and left-multiply both sides by $\mathbf{V}^\top$. We get

$$[\mathbf{V}^\top\hat{\mathbf{y}}]_i = \sum_{k=0}^{K-1} h_k \lambda_i^k [\mathbf{V}^\top\mathbf{x}]_i. \tag{16}$$

Hence, Theorem 2.1 holds for any $h_k \neq 0$.
To show Theorem 2.2, we write (6) in the matrix form

$$\hat{\mathbf{y}} = [\mathbf{x}\ \mathbf{A}\mathbf{x} \dots\ \mathbf{A}^{K-1}\mathbf{x}]\,[h_0 \dots h_{K-1}]^\top. \tag{17}$$

To show there exists h_k such that $\hat{\mathbf{y}} = \mathbf{y}$, we consider $K = N$, which yields a linear system of N equations (i.e., $\hat{\mathbf{y}}_i = \mathbf{y}_i$ for $i \in [N]$) with N unknowns $h_0, \dots, h_{N-1}$. Thus, it suffices to show that the vectors $\mathbf{x}, \mathbf{A}\mathbf{x}, \dots, \mathbf{A}^{N-1}\mathbf{x}$ are linearly independent. Consider projecting them to the eigen-basis of $\mathbf{A}$, i.e.,

$$\mathbf{V}^\top[\mathbf{x}\ \mathbf{A}\mathbf{x}\ \dots\ \mathbf{A}^{K-1}\mathbf{x}] \equiv [\tilde{\mathbf{x}}\ \mathbf{\Lambda}\tilde{\mathbf{x}}\ \dots\ \mathbf{\Lambda}^{N-1}\tilde{\mathbf{x}}], \tag{18}$$

where $\tilde{\mathbf{x}} \coloneqq \mathbf{V}^\top\mathbf{x}$. Since $\mathbf{V}$ is invertible, it remains to show that $\tilde{\mathbf{x}}, \dots, \mathbf{\Lambda}^{N-1}\tilde{\mathbf{x}}$ are linearly independent. Let $\boldsymbol{c} \in \mathbb{R}^N, \mathbf{M} \in \mathbb{R}^{N\times d}$,

and $\boldsymbol{c} \odot \mathbf{M}$ denote multiplying the i-th row of $\mathbf{M}$ by the i-th component of $\boldsymbol{c}$. We write $\tilde{\mathbf{x}} = \boldsymbol{c} \odot \mathbf{1}$ where $\mathbf{1}$ denotes the all-ones vector. Then the matrix $[\tilde{\mathbf{x}}\ \boldsymbol{\Lambda}\tilde{\mathbf{x}}\ \ldots\ \boldsymbol{\Lambda}^{N-1}\tilde{\mathbf{x}}]$ reduces to

$$\boldsymbol{c} \odot \begin{bmatrix} 1 & \lambda_1 & \ldots & \lambda_1^{N-1} \\ 1 & \lambda_2 & \ldots & \lambda_2^{N-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \lambda_N & \ldots & \lambda_N^{N-1} \end{bmatrix}. \tag{19}$$

Observe that (19) is a row-wise scaled Vandermonde matrix, which has determinant $\prod_i [\boldsymbol{c}]_i \prod_{i<j} (\lambda_i - \lambda_j)$. By assumption that $[\mathbf{V}^\top \mathbf{x}]_i \neq 0$ for $1 \leq i \leq N$, all entries $[\boldsymbol{c}]_i$ are nonzero. By assumption that the eigenvalues all have multiplicity one, $\lambda_i - \lambda_j \neq 0$ for all $i < j$. Therefore, (19) has nonzero determinant and linearly independent columns, which completes the proof.

6.3. Experiment Details for Sec. 4.1

Data. Our chosen $\mathbf{W}$ in (12) follows the degree-corrected SBM model in Def.3, which exhibits block structure via the two parameters p, q and core-periphery pattern via the degree function. It is easy to check that $\mathbf{W}$ in (12) satisfies integrability condition in Def. 2 and the Lipschitz continuity assumption (i) in Thm. 1.

Methods. For a comprehensive investigation, we compare spectral embeddings and GNNs using two graph operators: the graph adjacency matrix $\mathbf{A}$ and the normalized adjacency $\tilde{\mathbf{A}}$. Since $\mathbf{W}$ has 2 communities, we choose ϕ_{SE} as the top-K eigenvectors of the graph operator where $K \in \{2, 6, 10, 20\}$, combined with the top-2 principal components of the nodes features X per (5), and c_{SE} as a multilayer-perception with 1-hidden layer. We choose ϕ_{GNN} as a degree-2 polynomial graph filter with 2 layers, and c_{GNN} as a linear layer. All methods are trained with full-batch gradient descent, using learning rate 0.02 for 200 epochs (with early stopping if the loss has converged) and dropout probability 0.5. For GNNs, We use PReLU nonlinearity (i.e., ReLU with a learnable parameter for negative inputs).

6.4. Experiment Details for Sec. 4.2

Data. The Chameleon webpage network has 2277 nodes with average node degree 13.8, where nodes represent webpages and edges are hyperlinks between them. The node features are 2325-dimensional bag-of-words vectors of the webpages, and node labels are 5 webpage categories. We use the same data splits (48/32/20 for train/validation/test) from [35] released in Pytorch Geometric [36].

Methods. We use the similar setup as described in Section 6.3, except using SE dimension $K = \kappa \in \{150, 200\}$, and learning rate 0.01.