A Spectral Framework for Tracking Communities in Evolving Networks

Jacob Hume
University of Cambridge*
jakehume@umich.edu

Laura Balzano
University of Michigan, Ann Arbor
girasole@umich.edu

Abstract

Discovering and tracking communities in time-varying networks is an important task in network science, motivated by applications in fields ranging from neuroscience to sociology. In this work, we characterize the celebrated family of spectral methods for static clustering in terms of the low-rank approximation of high-dimensional node embeddings. From this perspective, it becomes natural to view the evolving community detection problem as one of subspace tracking on the Grassmann manifold. While the resulting optimization problem is nonconvex, we adopt a recently proposed block majorize-minimize Riemannian optimization scheme to learn the Grassmann geodesic which best fits the data. Our framework generalizes any static spectral community detection approach and leads to algorithms achieving favorable performance on synthetic and real temporal networks, including those that are weighted, signed, directed, mixed-membership, multiview, hierarchical, cocommunity-structured, bipartite, or some combination thereof. We demonstrate how to specifically cast a wide variety of methods into our framework, and demonstrate greatly improved dynamic community detection results in all cases.

1 Introduction

Consider a multiplex graph whose layers represent snapshots $G_1, \ldots, G_T$ of a time-varying network with d nodes. The goal is to partition each G_i into communities — node groupings of relatively higher intraconnectivity — in a manner that is temporally coherent and robust to noise [1]. This problem of *evolving community detection* has recently found myriad applications. For example, in social and computer networks, the continual estimation of community structure allows for robust containment of infection [2]. In brain networks, communities are believed to represent node collections dedicated to specialized functions such as memory or vision [3]; tracking changes (induced e.g. via hormonal fluctuations) in these collections illuminates how physiological factors influence cognition [4].

Many classical approaches to evolving community detection behave locally, matching network topology and/or partitions across adjacent snapshots [5]. More recently, a number of *global* [6, 7], or *cross-time* [8], approaches have arisen in the offline setting. These methods utilize information across all snapshots to obtain more stable communities of long-term temporal coherence [5, 8]. Numerous techniques for static community detection have been extended to this global setting [6, 9, 10]. However, the well-studied family of *spectral methods*, approaches based on clustering Euclidean node embeddings derived from the extreme end of a real matrix's spectrum [11–17], is a collection of notable exceptions. In addition to being among the most competitive [18], scalable [19], and theoretically rich [20–23] approaches to static community detection, spectral methods are also among the most general, with the above definition including techniques for clustering signed [24–26], mixed-membership [19, 27], directed [28, 29], multiview [24, 30], cocommunity-structured [31], hierarchical [32], hyper [33], motif-based [34], and higher-order [35, 36] networks. It is thus of great interest to develop temporal methods for spectral community detection inheriting these properties.

*Work completed while the author was at Michigan.

J. Hume and L. Balzano, A Spectral Framework for Tracking Communities in Evolving Networks. *Proceedings of the Third Learning on Graphs Conference (LoG 2024)*, PMLR 269, Virtual Event, November 26–29, 2024.

Notation	Description
G	A (simple, directed, multiview...) graph with d nodes, suitably connected
$\boldsymbol{A}$	The (possibly weighted) adjacency matrix of G
$\deg(\ell)$	The degree of a node ℓ of G defined as $\deg(\ell) = \sum_{j \neq \ell} A_{\ell j}$
$\mathrm{vol}(S)$	The volume of a node subset S, defined as $\mathrm{vol}(S) = \sum_{v \in S} \deg(v)$
$\boldsymbol{D}$	The degree matrix $\mathrm{diag}(\deg(1), \ldots, \deg(d))$ of G.
$\lambda(\boldsymbol{H}); \lambda_i(\boldsymbol{H})$	The spectrum of a matrix $\boldsymbol{H}$; the ith largest eigenvalue of $\boldsymbol{H}$.
$\sigma(\boldsymbol{H}); \sigma_i(\boldsymbol{H})$	The singular values of a matrix $\boldsymbol{H}$; the ith largest singular value of $\boldsymbol{H}$.
$\langle \boldsymbol{H} \rangle$	The span of a matrix or vector $\boldsymbol{H}$
$\mathrm{St}(d, k)$	The Stiefel manifold of matrices in $\mathbb{R}^{d \times k}$ with orthonormal columns
O_k	The orthogonal group of matrices in $\mathbb{R}^{k \times k}$ with orthonormal columns
$\mathrm{Gr}(d, k)$	The Grassmann manifold of k-dimensional linear subspaces of $\mathbb{R}^d$
$\mathscr{A}$	An algorithm which clusters d nodes with Euclidean embeddings into $\mathbb{R}^k$ derived from the extreme of a matrix spectrum (a *static spectral method*)
$\boldsymbol{M}_k = \boldsymbol{U}_k \boldsymbol{\Sigma}_k \boldsymbol{V}_k^\top$	The rank-k singular value decomposition $\boldsymbol{U}_k(\boldsymbol{M}) \boldsymbol{\Sigma}_k(\boldsymbol{M}) \boldsymbol{V}_k(\boldsymbol{M})^\top$ of $\boldsymbol{M}$
MCM of $\mathscr{A}$	A matrix $\boldsymbol{M}$ satisfying $\langle \boldsymbol{U}_k(\boldsymbol{M}) \rangle = \langle \boldsymbol{C} \rangle$, where $\boldsymbol{C}$'s rows are the spectral embeddings from $\mathscr{A}$ into $\mathbb{R}^k$. Called a *modeled clustering matrix (MCM)*

Contributions. We study a natural and versatile framework for temporally generalizing spectral methods for static community detection in a global fashion. Our approach is based on the observation (Section 2) that any spectral method may be characterized in terms of a least squares-optimal *low-rank approximation of some matrix* $\boldsymbol{M}$. In this light, it becomes natural to formulate the evolving spectral community detection problem using dynamics on the Grassmann manifold $\mathrm{Gr}(d, k)$ of k-dimensional linear subspaces of $\mathbb{R}^d$ [37]. Identifying $\{G_i\}_{i=1}^T$ with a discrete trajectory on $\mathrm{Gr}(d, k)$, the aforementioned notions of 'stability' and 'temporal coherence' may be interpreted as constraining this subspace trajectory to lie on some highly regular Grassmann curve. Recent subspace estimation literature [38, 39] suggests geodesic curves to be an elegant and effective candidate. Our approach is to thus learn the Grassmann geodesic that best models the data, capturing a robust continuum of node embeddings from which a sequence of clusterings may be computed. Advantages include:

- Any community detection method based on the leading or trailing eigenvectors of a matrix admits temporal generalization, as shown in Section 2.2. This is achieved using subspace estimation methods that leverage the regularity of Grassmann geodesics. The proposed framework provides algorithms for community detection in a vast array of evolving networks, including evolving networks with weighted, signed, and/or directed edges, evolving networks with overlapping, hierarchical, or cocommunity structure, and evolving networks with multiple views (Table 1).
- If the desired number of communities k_c is known in advance, the framework produces algorithms capable of utilizing this information (Algorithm 2). If not, a simple yet effective extension can ascertain k_c at each time step automatically (Figure 4, Algorithm 9).
- An intuitive heuristic (proven in Section 3) allows practitioners to visualize the extent to which their data satisfies the framework's core assumption regarding Grassmann geodesic structure when $k_c = 2$. Empirically, it indicates that smooth temporal evolution guarantees the presence of approximately geodesic structure in a dynamic stochastic block model (dynamic SBM), affirming that the geodesic assumption is a natural one.
- The proposed method is remarkable for the performance gains it achieves across the diverse array of networks mentioned above. For all network types considered, the adjusted mutual information and/or element-centric similarity between the estimated and true dynamic communities is above 0.8 — and often very near 1.0 (perfect recovery) — on appropriate variants of the dynamic SBM and on real data. This represents significant gains over the static counterparts and other dynamic methods. The approach is robust to noise, reliably recovering planted community structure in noise regimes for which the corresponding static methods perform around random chance.

Related Work. Much early work on evolving community detection focuses on the online setting via the following 'two-stage approach [5]':

1. Detect communities per-snapshot using a static method, such as [40–43] ;
2. Match communities across adjacent snapshots; we will call this *local temporal smoothing* [8].

For example, [44] applies the Louvain method [41] at each snapshot, imposing local temporal smoothing by initializing the method at snapshot t with results from snapshot $t-1$. The technique in [45] computes similarity scores between the static communities detected at time t and those at $t \pm 1, 2$ and uses a 'network sliding window' to smoothen away noise. In terms of spectral methods, a local temporal smoothing approach to spectral clustering is offered in [46]. The authors apply spectral clustering per-snapshot, but with an added 'temporal cost' parameter to the normalized cut objective function that quantifies how well a partition at time t clusters the data at time $t-1$. They then relax their objective in a manner similar to that deriving static spectral clustering [47], and obtain an optimal solution to this relaxation at each time step in terms of eigenvectors of a new, 'temporally smoothed' matrix. In what may be viewed as a substantial offline extension of this, [48] offers an approach to temporal spectral clustering via the simultaneous estimation of T smoothened Laplacian eigenbases $\bar{\boldsymbol{U}}_i$, $i \in [T]$, which are encouraged to be close both to their static counterpart $\boldsymbol{U}_i$ and smoothened predecessor $\bar{\boldsymbol{U}}_{i-1}$. This approach is 'global' in the sense that its objective function aggregates community information across all time points, and hence estimates all communities simultaneously. At the same time, it is 'local' in the sense that the objective only considers relationships between data in directly adjacent snapshots. In offline contexts where the full graph sequence is known *a priori*, it is natural to pursue methods capable of capturing long-term temporal correlations [7]. Recent such global approaches include [6], which concatenates adjacency matrix snapshots into a rank-3 tensor and applies nonnegative tensor factorization techniques, generalizing [49]; [50], which offers a dynamic model based on preferential attachment phenomena, generalizing [9]; and [10], which applies methods of statistical inference to the dynamic stochastic block model [51], generalizing techniques surveyed in [52]. Analogously, our work simultaneously generalizes to the global temporal setting methods such as spectral modularity maximization [12–14], (un)normalized spectral clustering/graph partitioning [15, 16, 53], Bethe Hessian clustering [17], and the multitudinous extensions of these to different network modalities. Different spectral approaches find utility in different contexts, and we envision accordingly diverse applications for our framework. All proofs are deferred to the appendix.

2 Proposed Framework

Our proposed framework leverages the general assumption of spectral methods that the extremal eigenvectors of a given matrix provide a good embedding for clustering graph nodes [11–17]. We will show that these eigenvectors also span a subspace of best low-rank approximation to a modification of said matrix, and low-dimensional subspaces have a natural dynamic extension with curves on the Grassmann manifold of k-dimensional subspaces of $\mathbb{R}^d$, which forms a compact Riemannian manifold with metric inherited from Euclidean space [37] . Our framework produces an extension of any static spectral method to the time-varying setting by applying a recent novel algorithm for fitting Grassmann geodesics [38]. In this section, we first discuss the general template for spectral community detection in a single network instance. A key feature is the connection of a subspace for node embeddings to a low-rank matrix approximation. We then give our proposed method, which uses ideas from subspace tracking to guarantee those subspace node embeddings are smooth in time.

Let $\{G_i\}_{i=1}^T$ be a multiplex graph whose layers represent snapshots of a time-varying network with d nodes. The goal is to partition each $\boldsymbol{G}_i$ into k communities — divisions into groups of increased connectivity [54] — in a temporally coherent manner. For the case $T = 1$, a number of successful *static* community detection methods exist. Especially well-studied are the *spectral methods*, which derive Euclidean node embeddings from the eigenvectors of a *clustering matrix* $\boldsymbol{R}$, then employ a Euclidean clustering approach to obtain community assignments. Our proposed method relies on the fact that a majority of spectral methods may be (re)formulated per the following template, a claim we substantiate in Section 2.2 following the outline of our method in terms of said template in Section 2.1.

Algorithm 1 Template for Spectral Community Detection in Static Networks

Input: Graph G, number k_c of communities to detect, embedding dimension k_e
Input: Spectral algorithm $\mathscr{A}$ for community detection, with clustering matrix $\boldsymbol{R}$
1: **Embedding:** Embed the nodes of G into $\mathbb{R}^d$ as columns of some matrix $\boldsymbol{M} = \boldsymbol{M}(\boldsymbol{R}) \in \mathbb{R}^{d \times d}$
2: **Spatial denoising:** Compress the node embeddings into a linear subspace $\mathcal{U} = \langle \boldsymbol{U} \rangle \in \mathrm{Gr}(d, k_e)$, $\boldsymbol{U} \in \mathrm{St}(d, k_e)$, such that the reconstruction error $\|\boldsymbol{M} - \boldsymbol{U}\boldsymbol{U}^\top \boldsymbol{M}\|_{\mathrm{F}}^2$ is minimized
3: **Euclidean clustering:** Follow $\mathscr{A}$'s specifications to derive community assignments from $\boldsymbol{U}$
Output: An assignment $(Z_1, \ldots, Z_k)$ of G's nodes into k_c communities.

The spectral nature of $\mathscr{A}$ is encoded in the subspace $\mathcal{U} \in \mathrm{Gr}(d,k)$, as this subspace is represented by the eigenbasis $\boldsymbol{U}_k \in \mathrm{St}(d,k)$ for $\boldsymbol{M}_k = \boldsymbol{U}_k \boldsymbol{\Sigma}_k \boldsymbol{V}_k^\top$ a rank-k approximation to $\boldsymbol{M}$. We note, though, that any choice of orthonormal basis $\boldsymbol{U}$ of $\mathcal{U}$ suffices.[2] We call the matrix $\boldsymbol{M}$ a *modeled clustering matrix (MCM) of* $\mathscr{A}$, for its defining property is that the rank-k subspace $\langle \boldsymbol{U}_k \rangle$ optimally modeling its columns is provided by spectral embeddings from $\mathscr{A}$.

Constructing an MCM for a given spectral method $\mathscr{A}$ is not difficult; indeed, Section 2.2 will show that choosing $\boldsymbol{M} = \boldsymbol{I} \pm \boldsymbol{R}/\|\boldsymbol{R}\|_{\mathrm{F}}$ suffices for all spectral methods of which the authors are aware, although other choices may yield more elegant interpretations. Finally, we remark that $k_c = k_e$ for the vast majority of algorithms. In principle they can differ, though. For example, when $k_c = 2$ it is common to choose $k_e = 1$, usually because $\mathscr{A}$ admits some natural derivation for this special case which exploits the algebraic structure of $\mathbb{R}$ in step 3 [12, 47]. It has also been argued that choosing $k_e \geq k_c$ is superior for some $\mathscr{A}$ [11, 13, 55].

2.1 Proposed Method

Now assume we are given a time-varying graph $\{G_i\}_{i=1}^T$, with the goal of recovering the true community structure at each time step in a manner that is robust to both spatial (i.e., in the high-dimensional embedding space $\mathbb{R}^d$) and temporal noise. The linear subspace modeling of step 2 in Algorithm 1 aims to handle the former, but a regularity constraint must be applied to the sequence $\{\langle \boldsymbol{U}_i \rangle\}_{i=1}^T$ of subspaces in order to attain the latter. A natural choice is to seek a geodesic of best fit for the data on the Grassmann manifold, either by fitting a geodesic directly through the points $\{\langle \boldsymbol{U}_i \rangle\}_{i=1}^T$ or by learning a Grassmann curve whose ith sample approximates $\boldsymbol{M}_i$ while obeying a geodesic-constrained trajectory. The former may be viewed as spatial denoising followed by temporal denoising; the latter as simultaneous spatiotemporal denoising. Our method can accommodate either interpretation, but we will limit focus to the latter.

Geodesics on the Grassmann manifold, like lines in Euclidean space, behave as parsimonious interpolating curves. A geodesic $\boldsymbol{U} : [0,1] \to \mathrm{Gr}(d,k)$ with starting point $\langle \boldsymbol{H} \rangle$, $\boldsymbol{H} \in \mathrm{St}(d,k)$, in the direction of $\boldsymbol{Y} \in T_{\langle \boldsymbol{H} \rangle}\mathrm{Gr}(d,k)$, may be parameterized as [56]

$$\boldsymbol{U}(t; \boldsymbol{H}, \boldsymbol{Y}, \boldsymbol{\Theta}) = \overbrace{\begin{bmatrix} \boldsymbol{H} & \boldsymbol{Y} \end{bmatrix}}^{=:\boldsymbol{P}} \overbrace{\begin{bmatrix} \cos(\boldsymbol{\Theta} t) \\ \sin(\boldsymbol{\Theta} t) \end{bmatrix}}^{=:\boldsymbol{C}(t)} = \boldsymbol{P}\boldsymbol{C}(t), \tag{1}$$

where $\boldsymbol{\Theta} \in \mathrm{diag}(\mathbb{R}^{k \times k})$ consists of principal angles between geodesic endpoints.[3] Our goal is to learn geodesic parameters such that the aggregated reconstruction error

$$\mathcal{L}\big(\boldsymbol{U}(t; \boldsymbol{H}, \boldsymbol{Y}, \boldsymbol{\Theta})\big) = \sum_{i=1}^{T} \|\boldsymbol{M}_i - \boldsymbol{U}(t_i)\boldsymbol{U}(t_i)^\top \boldsymbol{M}_i\|_{\mathrm{F}}^2 \tag{2}$$

is minimized. Here, t_i is the continuous time point assigned by the user to snapshot index i. For simplicity, we always assume $t_1, \ldots, t_T$ are equally spaced along $[0,1]$. Unlike Euclidean lines, no closed-form solution for geodesic regression on the Grassmann manifold is known, and moreover the objective (2) is nonconvex. Following [38], we attempt to minimize (2) via block coordinate descent alternating between optimizing $\boldsymbol{P}$ and optimizing $\boldsymbol{\Theta}$. We include the algorithm steps here for completeness, and their derivations may be found in [38]. We adopt this approach because it is hyperparameter-free and each update monotonically descends (2) while converging to a global optimum in a majority of [38]'s experiments. Additionally, the work in [57] analyzed this algorithm and proved convergence to a stationary point.

$\boldsymbol{P}$ Update. Assume $\boldsymbol{\Theta}$ is fixed along with prior iterate $\boldsymbol{P}^{(n)}$. Then (2) is minimized by setting

$$\boldsymbol{P}^{(n+1)} = \boldsymbol{W}\boldsymbol{V}^\top, \tag{3}$$

where $\boldsymbol{W}$ and $\boldsymbol{V}$ are obtained from the singular value decomposition $\boldsymbol{W}\boldsymbol{\Sigma}\boldsymbol{V}^\top$ of the $d \times 2k$ matrix

$$\sum_{i=1}^{T} \begin{bmatrix} \boldsymbol{M}_i \boldsymbol{M}_i^\top \boldsymbol{P}^{(n)} \boldsymbol{D}_i \cos(\boldsymbol{\Theta} t_i) & \boldsymbol{M}_i \boldsymbol{M}_i^\top \boldsymbol{P}^{(n)} \boldsymbol{D}_i \sin(\boldsymbol{\Theta} t_i) \end{bmatrix} \tag{4}$$

[2] Technically, it is assumed here that the Euclidean clustering method of $\mathscr{A}$ (step 3, e.g. k-means) is invariant under linear isometry of the input space. This is true for all spectral methods of which the authors are aware.

[3] That is, $\boldsymbol{\Theta} = \cos^{-1}(\boldsymbol{S})$, where $\boldsymbol{S}$ is obtained via the singular value decomposition $\boldsymbol{Z}\boldsymbol{S}\boldsymbol{Q}^\top = \boldsymbol{U}(0)^\top \boldsymbol{U}(1)$.

$\boldsymbol{\Theta}$ Update. Suppose $\boldsymbol{P}$ is fixed along with prior iterate $\boldsymbol{\Theta}^{(n)}$. In contrast to the above, when $\boldsymbol{\Theta}$ is fixed and $\boldsymbol{P}$ is to be updated, no analytic expression for a minimizer of (2) is known in this case. However, the objective is separable in the entries $\theta_1, \ldots, \theta_k$ of $\boldsymbol{\Theta}$, and they can each be iteratively updated over M iterations as

$$\theta_j^{(0)} := [\boldsymbol{\Theta}^{(n)}]_{jj};\ \theta_j^{(m+1)} = \theta_j^{(m)} - s^{(m)} \sum_{i=1}^{T} \dot{f}_{i,j}\big(\theta_j^{(m)}\big),\ \text{where } s^{(m)} = 1 \Big/ \sum_{i=1}^{T} w_{f_{i,j}}\big(\theta_j^{(m)}\big) \tag{5}$$

may be interpreted as a (variable) gradient descent step size and

$$\dot{f}_{i,j}(\theta_j) = \frac{t_i \sqrt{(\alpha_{i,j} - \gamma_{i,j})^2 + 4\beta_{i,j}^2}}{2} \sin\left(2\theta_j t_i - \phi_{i,j}\right) \quad ; \tag{6}$$

$$w_{f_{i,j}}(\theta_j) = \frac{\dot{f}_{i,j}(\theta_j)}{-\frac{\pi}{2t_i} + \left(\theta_j - \frac{\phi_{i,j}+\pi}{2t_i}\right) \bmod \frac{2\pi}{2t_i}} \tag{7}$$

for

$$\phi_{i,j} = \operatorname{arctan2}\left(\beta_{i,j}, \frac{\alpha_{i,j} - \gamma_{i,j}}{2}\right) \qquad \beta_{i,j} = [\boldsymbol{Y}^\top \boldsymbol{M}_i \boldsymbol{M}_i \boldsymbol{H}]_{jj} \tag{8}$$

$$\alpha_{i,j} = [\boldsymbol{H}^\top \boldsymbol{M}_i \boldsymbol{M}_i^\top \boldsymbol{H}]_{jj} \qquad \gamma_{i,j} = [\boldsymbol{Y}^\top \boldsymbol{M}_i \boldsymbol{M}_i \boldsymbol{Y}]_{jj}. \tag{9}$$

Here, $\operatorname{arctan2}(y, x)$ denotes the angle swept by the x-axis in $\mathbb{R}^2$ as it rotates counter-clockwise into the point (x, y). We then set

$$\boldsymbol{\Theta}^{(n+1)} = \operatorname{diag}(\theta_1^{(M)}, \ldots, \theta_k^{(M)}). \tag{10}$$

Initialization. Choosing $\boldsymbol{P}^{(0)}$ and $\boldsymbol{\Theta}^{(0)}$ amounts to choosing an initial geodesic to iteratively optimize. This in turn amounts to choosing geodesic endpoints based on the data $\{\boldsymbol{M}_i\}_{i=1}^T$. For our problem setting, we choose the rank-k least squares-optimal subspaces approximating $\boldsymbol{M}_1$ and $\boldsymbol{M}_T$ — obtained via k-truncated singular value decompositions $\boldsymbol{M}_1 = \boldsymbol{H}_1 \boldsymbol{\Sigma}_1 \boldsymbol{K}_1^\top$ and $\boldsymbol{M}_T = \boldsymbol{H}_T \boldsymbol{\Sigma}_T \boldsymbol{K}_T^\top$ — as said geodesic's initial point and end point respectively. This determines the direction $\boldsymbol{Y}$ to be $\boldsymbol{F}\boldsymbol{G}^\top$, where, writing $\boldsymbol{Z}\boldsymbol{S}\boldsymbol{Q}^\top = \boldsymbol{H}_1^\top \boldsymbol{H}_T$, $\boldsymbol{F}$ and $\boldsymbol{G}^\top$ come from a singular value decomposition $(\boldsymbol{I} - \boldsymbol{H}_1 \boldsymbol{H}_1^\top)\boldsymbol{H}_T \boldsymbol{Q} = \boldsymbol{F}\boldsymbol{D}\boldsymbol{G}^\top$. Thus, we initialize with

$$\boldsymbol{P}^{(0)} := \begin{bmatrix} \boldsymbol{H}_1 & \boldsymbol{F}\boldsymbol{G}^\top \end{bmatrix} \qquad \boldsymbol{\Theta}^{(0)} := \cos^{-1}(\boldsymbol{H}_1^\top \boldsymbol{H}_T). \tag{11}$$

Algorithm 2 Evolving Community Detection with Grassmann Geodesics

Input: Graphs and associated snapshot times $\{G_i, t_i\}_{i=1}^T$, number k_c of communities to track, dynamic embedding dimension k_e

Input: Static spectral method $\mathscr{A}$ admitting an MCM (i.e., $\mathscr{A}$ formulated as in algorithm 1)

1: **Embedding:** Follow $\mathscr{A}$ to form high-dimensional node embeddings as MCMs $\{\boldsymbol{M}_i\}_{i=1}^T$

2: **Spatiotemporal denoising** Estimate a $\mathrm{Gr}(d, k_e)$-geodesic $\boldsymbol{U}(t; \boldsymbol{H}, \boldsymbol{Y}, \boldsymbol{\Theta})$ via the minimization of (2) through alternating iterates (3) and (10), initialized per (11)

3: **Euclidean clustering:** For each snapshot index i, follow $\mathscr{A}$ to obtain community assignments from low-dimensional Euclidean node embeddings derived from $\boldsymbol{U}(t_i)$

Output: Assignments $(Z_1 \ldots, Z_{k_c})_i$ of the nodes in each of $\{G_i\}_{i=1}^T$ to communities

2.2 Instantiations

This section illustrates the generality of algorithm 1 by way of several examples, thereby providing a collection of static community detection methods extended by algorithm template 2 to the temporal setting. The (possibly weighted) simple network cases of spectral clustering and spectral modularity maximization are a faithful prototype for the exploration of other techniques using our framework. We therefore restrict present attention to these methods, with analogous discussion of other methods deferred to Appendix A. Table 1 summarizes all static spectral methods that we analyze and evaluate. To begin, we observe that any spectral method based on the extremal eigenvectors of a clustering matrix $\boldsymbol{R}$ easily admits an MCM $\boldsymbol{M}$, simply via the normalization, then shifting, of its spectrum.

Table 1: Example applications of static community detection across different network modalities, and associated spectral methods for pursuing it. SC abbreviates *spectral clustering*.

Modality	Example Application(s)	Methods
Simple	Image segmentation [15]	(Un)Normalized SC (USC)(NSC) [15, 16]
	Brain connectivity [58]	Spectral Modularity Maximization (SMM) [12–14]
		Bethe Hessian Clustering (BHC) [17]
Signed	Time series [59]	Signed Ratio SC (SRSC) [26]
	Voting networks [60]	Geometric Mean SC (GMSC) [25]
		Signed Power Mean SC (SPMSC) [24]
Overlapping	Social networks [61]	Overlapping SC (OSC) [19]
	Neuronal networks [62]	c-means SC (CSC) [27]
Directed	Genomics [63]	Degree-Discounted SC (DDSC) [28]
	Information networks [64]	Bibliographic SC (BSC) [28]
		Random Walk Directed SC (RWSC) [29]
Multiview	Measuring networks [65]	Grassmannian Multiview SC (GMVSC) [30]
	Multimedia analysis [66]	Power Mean Laplacian SC (PMLSC) [24]
Cocommunity	NLP [67]	Spectral Coclustering (SCC) [31]
Hierarchical	Biochemical graphs [68]	Hierarchical SC (HSC) [32]
	Brain networks [69]	

Proposition 1. *Let $\mathscr{A}$ be a static spectral algorithm with clustering matrix $\boldsymbol{R}$. If $\mathscr{A}$ clusters using the leading (resp. trailing) eigenvectors of $\boldsymbol{R}$, then $\boldsymbol{M} = \boldsymbol{I} + \boldsymbol{R}/\|\boldsymbol{R}\|_{\mathrm{F}}$ (resp. $\boldsymbol{I} - \boldsymbol{R}/\|\boldsymbol{R}\|_{\mathrm{F}}$) is an MCM for $\mathscr{A}$ (cf. Algorithm 1).*

Some alternative choices of $\boldsymbol{M}$ come with extra interpretations. See e.g. propositions 2 and 3 below. Proposition 1 suffices to generate MCMs for all spectral methods considered in this text, though, and will be employed by default when no alternative is specified.

Unnormalized Spectral Clustering (USC). Let $\boldsymbol{A}$ be the (possibly weighted) adjacency matrix and $\boldsymbol{D}$ be the degree matrix of the graph G. Unnormalized spectral clustering arises from the observation that the matrix $\boldsymbol{V}$ composed of the trailing k eigenvectors of the graph Laplacian $\boldsymbol{R} = \boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$ minimizes a relaxation of the NP-Hard $\mathrm{RatioCut}$ objective for partitioning a graph into communities of roughly balanced size [70], such that the k-dimensional embedding of G offered by $\boldsymbol{V}^\top$ unveils clustering structure in Euclidean space discoverable by heuristics like cardinality-constrained vector partitioning [11] or k-means [47].

Proposition 2. *The matrix $\boldsymbol{M} = n\boldsymbol{I} - \boldsymbol{L}$ is an MCM for unnormalized spectral clustering for any $n \geq \max|\lambda(\boldsymbol{L})|$. In particular, choosing any $n \geq 2\max_i \deg(i)$ — for example, $n = 2d$ when G is unweighted — yields an MCM.*

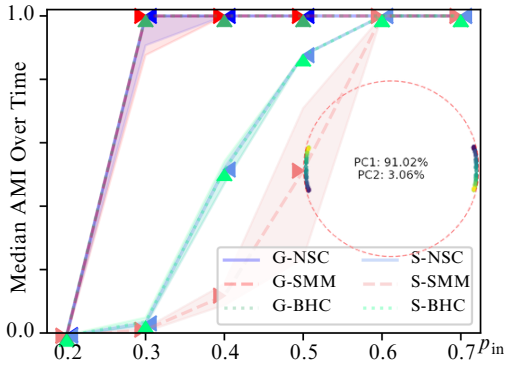
A notable special case of Proposition 2 occurs when G is a graph on k disjoint cliques containing s nodes each. In this case, the community structure of G is entirely unambiguous and $\boldsymbol{M} = s\boldsymbol{I} - \boldsymbol{L}$ turns out to be precisely rank-k (Appendix A.1).

Normalized Spectral Clustering (NSC). Normalized spectral clustering arises from a modification of the $\mathrm{RatioCut}$ objective wherein balance is encouraged among in-community edge densities rather than community sizes. The new objective's relaxation is minimized with the k smallest eigenvectors of the *normalized symmetric graph Laplacian* $\boldsymbol{L}^{\mathrm{sym}} := \boldsymbol{D}^{-1/2}\boldsymbol{L}\boldsymbol{D}^{-1/2}$.

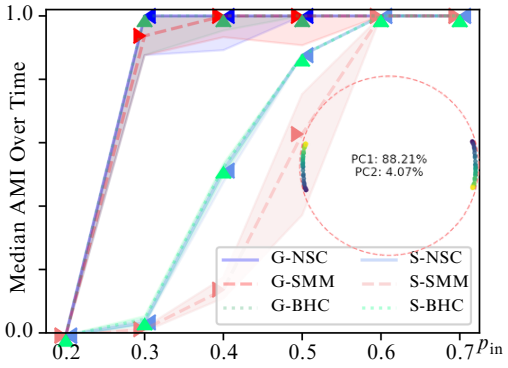
Proposition 3. *The normalized signless Laplacian $\boldsymbol{Q}^{sym} = \boldsymbol{D}^{-1/2}\boldsymbol{Q}\boldsymbol{D}^{-1/2}$, $\boldsymbol{Q} = \boldsymbol{D} + \boldsymbol{A}$, is an MCM for normalized spectral clustering.*

The tail of $\boldsymbol{Q}^{\mathrm{sym}}$'s spectrum is intimately related to *anti*community structure [71]. Proposition 3 complementarily relates the head of its spectrum to community structure.

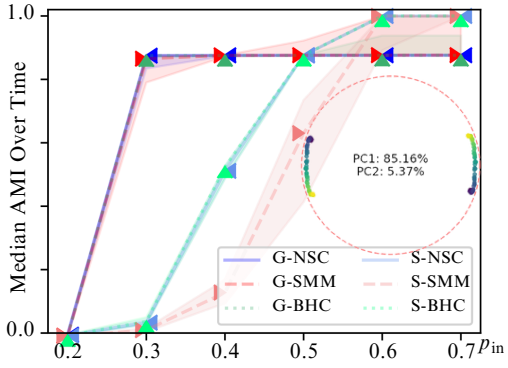
Spectral Modularity Maximization (SMM). In many real-world networks, what matters is not that inter-community edge density is low, but that it is *lower than expected*. The notion of network



(a) $p_{\text{switch}} = T/1000$



(b) $p_{\text{switch}} = T/750$



(c) $p_{\text{switch}} = T/500$

Figure 2. Comparison of median AMI $\in [0, 1]$ over time versus p_{in}, medianed over 50 simulations of the dynamic SBM ($d = T = 50$, $k = 2$, $p_{\text{out}} = 0.2$) as p_{switch} ranges from low to very high. Each unit circle displays the trajectory of modularity matrix first-eigenvectors as T progresses when $p_{\text{in}} = 0.4$. For low (1a) and medium (1b) values of p_{switch}, said trajectory 'walks along' the unit circle, suggesting (Proposition 4) that the dynamics satisfy the geodesic assumption. When p_{switch} is very high (1c), the trajectory 'falls off': the assumption has been violated. G- resp. S- refers to geodesic resp. static algorithm versions.

modularity proposed in [40] quantifies this. Numerous spectral approaches to approximately maximize modularity exist by clustering embeddings derived from the leading positive eigenvectors of the *modularity matrix* $\boldsymbol{B} = \boldsymbol{A} - \mathbb{E}\boldsymbol{A}$ [12–14].[4]

3 Experiments

We demonstrate the effectiveness of our method on real and synthetic evolving network data. On the synthetic data, we relate the behavior of our method to the presence of latent geodesic structure as graph dynamics range from smooth to jagged. We also show that the geodesic generalizations of 15 spectral methods (Table 1) via Algorithm 2 empirically outperform their static counterpart, where the static method is applied separately at each time point. On real data, we show that the geodesic method achieves favorable performance over a collection of popular temporal community detection benchmarks, including when the algorithm no longer assumes a fixed number of temporally stable latent communities. Experiments were all performed in `Python` on a 2019 MacBook Pro.[5]

Model selection and checking the geodesic assumption. Throughout this section, model selection (i.e., choosing k_e and/or k_c) is performed by letting k range over successive algorithm runs and choosing that which yields partitions of highest modularity [14]. We quantify 'partitions of highest modularity' by taking the mode over k, but other summarizations such as multilayer modularity [73] could be used as well. When $k_c = 2$, an additional step is often available for assessing the presence of geodesic structure. For clarity of exposition, we focus on spectral modularity maximization, $\boldsymbol{M} = \overline{\boldsymbol{B}} = \boldsymbol{I} + \boldsymbol{B}/\|\boldsymbol{B}\|_{\text{F}}$. When the goal is two-way clustering, it is customary to classify node ℓ

[4]Here, the expectation is taken with respect to the *Newman-Girvan Null Model* [12]: $[\mathbb{E}\boldsymbol{A}]_{j\ell} = \deg(j)\deg(\ell)/\text{vol}(A)$. This is standard in contemporary treatments of network science such as [72].

[5]Code is available at https://github.com/jacobh140/spectral-dcd.

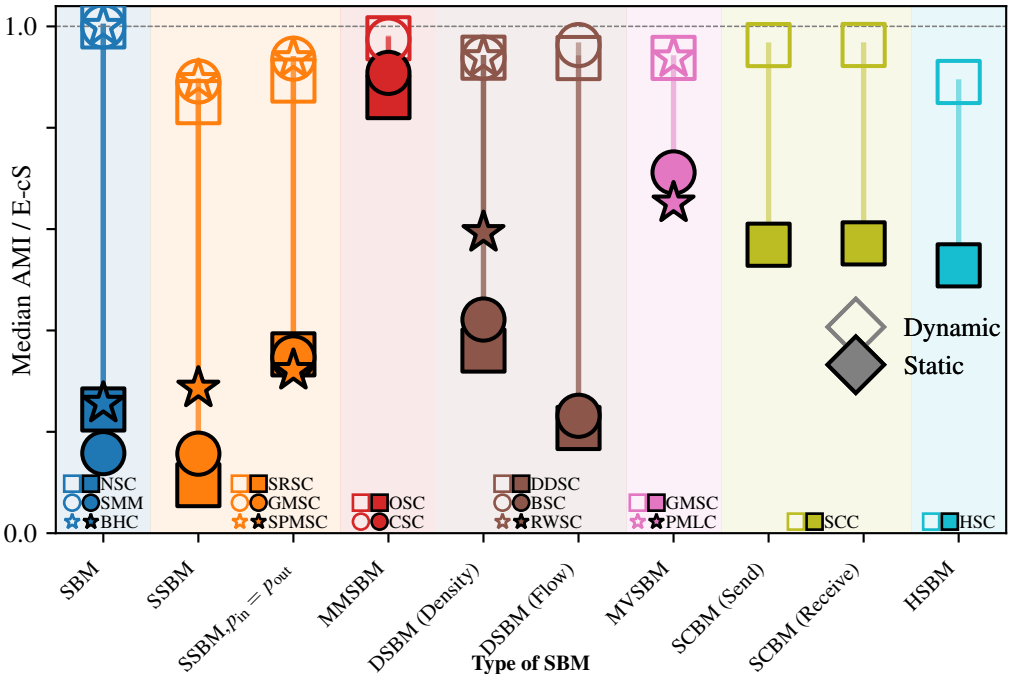


Figure 3. Comparison of median (over 50 simulations and 20 time steps of the appropriate SBM) AMI/E-cS for various static spectral methods and their dynamic generalization. Each color corresponds to a network modality, each line an SBM setting for that modality, and each symbol a spectral method for detecting communities in that modality (hollow for its dynamic extension, filled for static). By default where applicable, $d = 120$, $k = 2$, $p_{\text{in}} = 0.3$, $p_{\text{out}} = 0.2$, and $p_{\text{switch}} = 10^{-2}$. Exceptions are in the second SSBM and second DSBM parameter settings, where $p_{\text{in}} = p_{\text{out}}$ to enforce clustering based solely on edge affinity/orientation. Appendix A elaborates upon each individual column.

based on $\text{sign}(u_\ell)$, where $\boldsymbol{u}$ is the first eigenvector of $\overline{\boldsymbol{B}}$ [12]. This corresponds to considering the space $\text{Gr}(1, d)$ in Algorithm 2 (i.e., $k_e = 1$, $k_c = 2$). Of course, the lines comprising $\text{Gr}(1, d)$ are naturally identified with pairs of antipodal points on the $(d-1)$-sphere $\mathbb{S}^{d-1} \subset \mathbb{R}^d$, and geodesics on $\mathbb{S}^{d-1}$ coincide with great circle arcs. The main idea, then, is that if we concatenate the respective first eigenvectors $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_T$ of $\overline{\boldsymbol{B}}_1, \ldots, \overline{\boldsymbol{B}}_T$ and their negations $-\boldsymbol{u}_1, \ldots, -\boldsymbol{u}_T$ into a data matrix $\boldsymbol{X} \in \mathbb{R}^{d \times 2T}$ and project these columns into their 2-dimensional PCA subspace, we should see the projections of the $\boldsymbol{u}_i$ into $\mathbb{R}^2$ 'walk along' the unit circle in two mirrored trajectories if geodesic structure is present. The following proposition makes this precise. Results can be seen in Figure 2.

Proposition 4. *$\sigma_1(\boldsymbol{X}) \geq \sigma_2(\boldsymbol{X}) > \sigma_3(\boldsymbol{X}) = \cdots = \sigma_{\min(d,2T)}(\boldsymbol{X}) = 0$ if and only if the first singular subspaces of the $\overline{\boldsymbol{B}}_i$ lie in the image of a curve in $Gr(d, 1)$ that is a Riemannian geodesic.*

Synthetic. We evaluate with a dynamic model based on a setting of the dynamic stochastic block model (dynamic SBM) studied in [51]. Initially, d nodes are divided equally among k planted communities. At a given snapshot index $i \in [T]$, an edge is placed between each pair of intra-community nodes resp. inter-community nodes with probability p_{in} resp. p_{out}. Dynamics are introduced by stipulating that at each snapshot, any node (that has yet to switch) switches community with probability p_{switch}. The performance of algorithms for simple networks, quantified using adjusted mutual information (AMI) [74], is compared in Figure 2, which also assesses geodesic structure (using Proposition 4) as SBM dynamics range from smooth to very jagged. Algorithms for non-simple network modalities are evaluated using a dynamic extension of the appropriate SBM. That includes the signed SBM (SSBM) for signed networks, mixed-membership SBM (MMSBM), directed SBM (DSBM), multiview SBM (MVSBM), stochastic coblock model (SCBM), and hierarchical SBM (HSBM). The definition of each modified SBM differs minutely from that of the standard SBM outlined here and is provided in appendix A. Figure 3 shows these results: in all cases, the communities recovered by the Grassmannian-smoothed dynamic network embeddings outperform

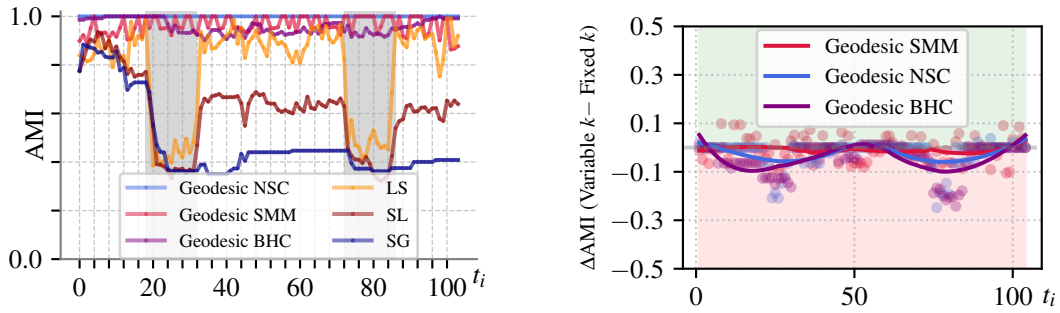


Figure 4. Evaluation on the two-day elementary school face-to-face interaction network of [76]. LEFT: Included as benchmarks are the Label Smoothing (LS) approach of [78], the Smoothed Louvain (SL) algorithm of [44], and the Graph Smoothing (SG) approach of [79]. The three geodesic approaches uniformly outperform the benchmarks. RIGHT: The AMI difference at each time step between Algorithm 2 (fixed k_c) and its extension (Appendix C) to the variable-k_c case. The data points are scattered, with each trajectory smoothened using a third-order Savitzky-Golay filter [80] to visually enhance any trends. The results imply that the two algorithms perform very similarly.

those from the static embeddings, sometimes dramatically. Overlapping and hierarchical community detection algorithms are evaluated using element-centric similarity (E-cS) [75].

Real. We assess performance on the face-to-face interaction network of [76]. Over the course of two elementary school days, $77,602$ contact events were recorded among 242 individuals (232 children in 10 classes, and the 10 teachers of those classes). The data was segmented into 10-minute intervals, yielding a sequence of $T = 102$ contact networks encoded by unweighted symmetric 232×232 adjacency matrices. As preprocessing, bridge edges were placed between graph components to maintain connectedness and teachers were removed from the network. The class memberships lend natural community structure to this evolving network, which are perfectly recovered at each time step by geodesic spectral clustering and near-perfectly recovered by the other geodesic methods (Figure 4, LEFT). We benchmark against the suite of popular dynamic community detection algorithms implemented within the `tnetwork` library [77], specifically the Label Smoothing (LS) [78], Smoothed Louvain (SL) [44], and Graph Smoothing (SG) [79] algorithms. The three geodesic approaches each outperform the benchmarks across all time, apparently dramatically so around the 3-4h point — presumably a lunch break — on each day. We refrain from making a claim as to whether the mingling of classes around lunchtime is best considered as a noisy sample or as the dissolution of the latent community structure, and we indicate our ambivalence by masking these intervals in gray. We hypothesized that the geodesic method's favorable performance could be attributed to it seeking precisely $k_c = 10$ communities at each time step while the benchmarks seek a variable number of communities. This turned out to be largely false: it is indeed the case that when stable latent communities are anticipated in the network dynamics, a dynamic community detection algorithm capable of utilizing this information such as Algorithm 2 is preferred. However, (Figure 4, RIGHT) shows that a simple extension of Algorithm 2 to the case of varying k_c, based on sweeping k_c to locally maximize modularity (Appendix C), performs nearly identically to Algorithm 2.

4 Conclusion

This work presented and analyzed a Grassmann geometry-based framework for generalizing the popular family of spectral algorithms for community detection in static networks to the time-varying setting. Our method is broadly applicable to spectral community detection methods and has excellent performance, as demonstrated on both synthetic and real data experiments. We note that our method only focuses on a single approach to fitting a geodesic to data. Extensions and improvements of said approach would induce analogous extensions and improvements here.

5 Acknowledgments

The authors would like to thank Mark Newman, Samuel Sottile, Jeffrey Fessler, Soo Min Kwon, Haroon Raja, Cameron Blocker, and Alexander Saad-Falcon for their insightful discussions.

References

[1] Deepayan Chakrabarti, Ravi Kumar, and Andrew Tomkins. Evolutionary clustering. In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 554–560, 2006. 1

[2] Nam P Nguyen, Thang N Dinh, Yilin Shen, and My T Thai. Dynamic social community detection and its applications. PLOS ONE, 9(4):e91431, 2014. 1

[3] Ed Bullmore and Olaf Sporns. The economy of brain network organization. Nature reviews neuroscience, 13(5):336–349, 2012. 1

[4] Joshua M Mueller, Laura Pritschet, Tyler Santander, Caitlin M Taylor, Scott T Grafton, Emily Goard Jacobs, and Jean M Carlson. Dynamic community detection reveals transient reorganization of functional brain networks across a female menstrual cycle. Network Neuroscience, 5(1):125–144, 2021. 1

[5] Rémy Cazabet, Giulio Rossetti, and Frederic Amblard. Dynamic community detection. Encyclopedia of social network analysis and mining, 2017. 1, 2, 29, 30

[6] Laetitia Gauvin, André Panisson, and Ciro Cattuto. Detecting the community structure and activity patterns of temporal networks: a non-negative tensor factorization approach. PLOS ONE, 9(1):e86028, 2014. 1, 3

[7] Kostas Christopoulos and Kostas Tsichlas. State-of-the-art in community detection in temporal networks. In IFIP International Conference on Artificial Intelligence Applications and Innovations, pages 370–381. Springer, 2022. 1, 3

[8] Giulio Rossetti and Rémy Cazabet. Community discovery in dynamic networks: a survey. ACM computing surveys (CSUR), 51(2):1–37, 2018. 1, 2, 30

[9] Zejun Sun, Yanan Sun, Xinfeng Chang, Qiming Wang, Xuetong Yan, Zhongqiang Pan, and Zong-ping Li. Community detection based on the matthew effect. Knowledge-Based Systems, 205:106256, 2020. 1, 3

[10] Amir Ghasemian, Pan Zhang, Aaron Clauset, Cristopher Moore, and Leto Peel. Detectability thresholds and optimal algorithms for community structure in dynamic networks. Physical Review X, 6(3):031005, 2016. 1, 3

[11] Charles J Alpert and So-Zen Yao. Spectral partitioning: The more eigenvectors, the better. In Proceedings of the 32nd annual ACM/IEEE design automation conference, pages 195–200, 1995. 1, 3, 4, 6, 29, 30

[12] Mark EJ Newman. Finding community structure in networks using the eigenvectors of matrices. Physical Review E—Statistical, Nonlinear, and Soft Matter Physics, 74(3):036104, 2006. 3, 4, 6, 7, 8, 29

[13] Xiao Zhang and Mark EJ Newman. Multiway spectral community detection in networks. Physical Review E, 92(5):052808, 2015. 4, 29, 30

[14] Scott White and Padhraic Smyth. A spectral clustering approach to finding communities in graphs. In Proceedings of the 2005 SIAM international conference on data mining, pages 274–285. SIAM, 2005. 3, 6, 7, 19

[15] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. IEEE Transactions on pattern analysis and machine intelligence, 22(8):888–905, 2000. 3, 6, 19

[16] Andrew Ng, Michael Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. Advances in neural information processing systems, 14, 2001. 3, 6

[17] Alaa Saade, Florent Krzakala, and Lenka Zdeborová. Spectral clustering of graphs with the bethe hessian. Advances in neural information processing systems, 27, 2014. 1, 3, 6, 19

[18] Zhao Yang, René Algesheimer, and Claudio J Tessone. A comparative analysis of community detection algorithms on artificial networks. Scientific reports, 6(1):30750, 2016. 1

[19] Yuan Zhang, Elizaveta Levina, and Ji Zhu. Detecting overlapping communities in networks using spectral methods. SIAM Journal on Mathematics of Data Science, 2(2):265–283, 2020. 1, 6, 22, 23, 30

[20] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. Neural computation, 15(6):1373–1396, 2003. 1

[21] Marianna Bolla, Brian Bullins, Sorathan Chaturapruek, Shiwen Chen, and Katalin Friedl. Spectral properties of modularity matrices. Linear Algebra and Its Applications, 473:359–376, 2015.

[22] Dario Fasino and Francesco Tudisco. An algebraic analysis of the graph modularity. SIAM Journal on Matrix Analysis and Applications, 35(3):997–1018, 2014.

[23] Fan RK Chung. Spectral graph theory, volume 92. American Mathematical Soc., 1997. 1

[24] Pedro Mercado, Francesco Tudisco, and Matthias Hein. Spectral clustering of signed graphs via matrix power means. In International Conference on Machine Learning, pages 4526–4536. PMLR, 2019. 1, 6, 20, 21, 28

[25] Pedro Mercado, Francesco Tudisco, and Matthias Hein. Clustering Signed Networks with the Geometric Mean of Laplacians. volume to appear, December 2016. doi: 10.48550/arXiv.1701.00757. 6, 20, 21

[26] Jérôme Kunegis, Stephan Schmidt, Andreas Lommatzsch, Jürgen Lerner, Ernesto W De Luca, and Sahin Albayrak. Spectral analysis of signed graphs for clustering, prediction and visualization. In Proceedings of the 2010 SIAM international conference on data mining, pages 559–570. SIAM, 2010. 1, 6, 20, 21

[27] Scott Wahl and John Sheppard. Hierarchical fuzzy spectral clustering in social networks using spectral characterization. In The twenty-eighth international flairs conference, 2015. 1, 6, 22, 23

[28] Venu Satuluri and Srinivasan Parthasarathy. Symmetrizations for clustering directed graphs. In Proceedings of the 14th international conference on extending database technology, pages 343–354, 2011. 1, 6, 23

[29] Dengyong Zhou, Jiayuan Huang, and Bernhard Schölkopf. Learning from labeled and unlabeled data on a directed graph. In Proceedings of the international conference on Machine learning (IOCML), pages 1036–1043, 2005. 1, 6, 24

[30] Xiaowen Dong, Pascal Frossard, Pierre Vandergheynst, and Nikolai Nefedov. Clustering on multi-layer graphs via subspace analysis on grassmann manifolds. IEEE Transactions on signal processing, 62(4):905–918, 2013. 1, 6, 27, 28

[31] Karl Rohe, Tai Qin, and Bin Yu. Co-clustering directed graphs to discover asymmetries and directional communities. Proceedings of the National Academy of Sciences, 113(45): 12679–12684, 2016. 1, 6, 24, 26

[32] Steinar Laenen, Bogdan Adrian Manghiuc, and He Sun. Nearly-optimal hierarchical clustering for well-clustered graphs. In International Conference on Machine Learning, pages 18207–18249. PMLR, 2023. 1, 6, 26, 27

[33] Dengyong Zhou, Jiayuan Huang, and Bernhard Schölkopf. Learning with hypergraphs: Clustering, classification, and embedding. Advances in neural information processing systems, 19, 2006. 1, 28

[34] William G Underwood, Andrew Elliott, and Mihai Cucuringu. Motif-based spectral clustering of weighted directed networks. Applied Network Science, 5:1–41, 2020. 1, 28

[35] Vincent P Grande and Michael T Schaub. Topological point cloud clustering. arXiv preprint arXiv:2303.16716, 2023. 1

[36] Sanjukta Krishnagopal and Ginestra Bianconi. Spectral detection of simplicial communities via hodge laplacians. Physical Review E, 104(6):064303, 2021. 1, 28

[37] Nicolas Boumal. An introduction to optimization on smooth manifolds. Cambridge University Press, 2023. 2, 3, 28

[38] Cameron J Blocker, Haroon Raja, Jeffrey A Fessler, and Laura Balzano. Dynamic subspace estimation with Grassmannian geodesics. arXiv preprint arXiv:2303.14851, 2023. URL https://arxiv.org/abs/2303.14851. 2, 3, 4

[39] Yi Hong, Roland Kwitt, Nikhil Singh, Nuno Vasconcelos, and Marc Niethammer. Parametric regression on the Grassmannian. IEEE transactions on pattern analysis and machine intelligence, 38(11):2284–2297, 2016. 2

[40] Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. Physical review E, 69(2):026113, 2004. 2, 7

[41] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. Journal of statistical mechanics: theory and experiment, 2008(10):P10008, 2008. 3

[42] Martin Rosvall and Carl T Bergstrom. Maps of random walks on complex networks reveal community structure. Proceedings of the national academy of sciences, 105(4):1118–1123, 2008.

[43] Brian Karrer and Mark EJ Newman. Stochastic blockmodels and community structure in networks. Physical Review E—Statistical, Nonlinear, and Soft Matter Physics, 83(1):016107, 2011. 2

[44] Thomas Aynaud and Jean-Loup Guillaume. Static community detection algorithms for evolving networks. In 8th international symposium on modeling and optimization in mobile, ad hoc, and wireless networks, pages 513–519. IEEE, 2010. 3, 9, 32, 33, 34

[45] Matteo Morini, Patrick Flandrin, Eric Fleury, Tommaso Venturini, and Pablo Jensen. Revealing evolutions in dynamical networks. arXiv preprint arXiv:1707.02114, 2017. 3

[46] Yun Chi, Xiaodan Song, Dengyong Zhou, Koji Hino, and Belle L Tseng. Evolutionary spectral clustering by incorporating temporal smoothness. In Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 153–162, 2007. 3

[47] Ulrike Von Luxburg. A tutorial on spectral clustering. Statistics and computing, 17:395–416, 2007. 3, 4, 6, 29, 30

[48] Fuchen Liu, David Choi, Lu Xie, and Kathryn Roeder. Global spectral clustering in dynamic networks. Proceedings of the National Academy of Sciences, 115(5):927–932, 2018. 3

[49] 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, pages 587–596, 2013. 3

[50] Zejun Sun, Yanan Sun, Xinfeng Chang, Feifei Wang, Zhongqiang Pan, Guan Wang, and Jianfen Liu. Dynamic community detection based on the matthew effect. Physica A: Statistical Mechanics and its Applications, 597:127315, 2022. 3

[51] Catherine Matias and Vincent Miele. Statistical clustering of temporal networks through a dynamic stochastic block model. Journal of the Royal Statistical Society Series B: Statistical Methodology, 79(4):1119–1141, 2017. 3, 8

[52] Emmanuel Abbe. Community detection and stochastic block models. arXiv preprint arXiv:1703.10146, 2017. 3

[53] Alex Pothen, Horst D Simon, and Kang-Pu Liou. Partitioning sparse matrices with eigenvectors of graphs. SIAM journal on matrix analysis and applications, 11(3):430–452, 1990. 3

[54] Mark EJ Newman. Spectral methods for community detection and graph partitioning. Physical Review E—Statistical, Nonlinear, and Soft Matter Physics, 88(4):042822, 2013. 3

[55] Nicola Rebagliati and Alessandro Verri. Spectral clustering with more than k eigenvectors. Neurocomputing, 74(9):1391–1401, 2011. 4, 29

[56] Thomas Bendokat, Ralf Zimmermann, and P-A Absil. A grassmann manifold handbook: Basic geometry and computational aspects. Advances in Computational Mathematics, 50(1): 6, 2024. 4, 28

[57] Yuchen Li, Laura Balzano, Deanna Needell, and Hanbaek Lyu. Convergence and complexity of block majorization-minimization for constrained block-riemannian optimization. arXiv preprint arXiv:2312.10330, 2023. 4

[58] Ed Bullmore and Olaf Sporns. Complex brain networks: graph theoretical analysis of structural and functional systems. Nature reviews neuroscience, 10(3):186–198, 2009. 6

[59] Saeed Aghabozorgi, Ali Seyed Shirkhorshidi, and Teh Ying Wah. Time-series clustering–a decade review. Information systems, 53:16–38, 2015. 6, 20

[60] Robert West, Hristo S Paskov, Jure Leskovec, and Christopher Potts. Exploiting social network structure for person-to-person sentiment analysis. Transactions of the Association for Computational Linguistics, 2:297–310, 2014. 6, 19

[61] J Chitra Devi and E Poovammal. An analysis of overlapping community detection algorithms in social networks. Procedia Computer Science, 89:349–358, 2016. 6

[62] Paul Kim and Sangwook Kim. Detecting overlapping and hierarchical communities in complex network using interaction-based edge clustering. physica A: Statistical Mechanics and its Applications, 417:46–56, 2015. 6, 22

[63] Ovidiu Popa, Einat Hazkani-Covo, Giddy Landan, William Martin, and Tal Dagan. Directed networks reveal genomic barriers and dna repair bypasses to lateral gene transfer among prokaryotes. Genome research, 21(4):599–609, 2011. 6, 23

[64] Jiayuan Huang, Tingshao Zhu, and Dale Schuurmans. Web communities identification from random walks. In European Conference on Principles of Data Mining and Knowledge Discovery, pages 187–198. Springer, 2006. 6, 23

[65] Mark EJ Newman. Network structure from rich but noisy data. Nature Physics, 14(6):542–545, 2018. 6, 26

[66] Guoqing Chao, Shiliang Sun, and Jinbo Bi. A Survey on Multi-View Clustering. IEEE transactions on artificial intelligence, 2(2):146–168, April 2021. ISSN 2691-4581. doi: 10.1109/tai.2021.3065894. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8925043/. 6

[67] Inderjit S Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, pages 269–274, 2001. 6, 24

[68] Erzsébet Ravasz, Anna Lisa Somera, Dale A Mongru, Zoltán N Oltvai, and A-L Barabási. Hierarchical organization of modularity in metabolic networks. Science, 297(5586):1551–1555, 2002. 6, 24

[69] Arian Ashourvan, Qawi K Telesford, Timothy Verstynen, Jean M Vettel, and Danielle S Bassett. Multi-scale detection of hierarchical community architecture in structural and functional brain networks. PLOS ONE, 14(5):e0215520, 2019. 6, 24

[70] Horst D Simon. Partitioning of unstructured problems for parallel processing. Computing systems in engineering, 2(2-3):135–148, 1991. 6

[71] Steve Kirkland and Debdas Paul. Bipartite subgraphs and the signless laplacian matrix. Applicable Analysis and Discrete Mathematics, pages 1–13, 2011. 6

[72] Mark Newman. Networks. Oxford university press, 2018. 7

[73] Peter J Mucha, Thomas Richardson, Kevin Macon, Mason A Porter, and Jukka-Pekka Onnela. Community structure in time-dependent, multiscale, and multiplex networks. Science, 328 (5980):876–878, 2010. 7, 31

[74] Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: is a correction for chance necessary? In Proceedings of the 26th annual international conference on machine learning, pages 1073–1080, 2009. 8

[75] Alexander J Gates, Ian B Wood, William P Hetrick, and Yong-Yeol Ahn. Element-centric clustering comparison unifies overlaps and hierarchy. Scientific reports, 9(1):8574, 2019. 9, 22

[76] Juliette Stehlé, Nicolas Voirin, Alain Barrat, Ciro Cattuto, Lorenzo Isella, Jean-François Pinton, Marco Quaggiotto, Wouter Van den Broeck, Corinne Régis, Bruno Lina, et al. High-resolution measurements of face-to-face contact patterns in a primary school. PLOS ONE, 6(8):e23176, 2011. 9

[77] Remy Cazabet. Documentation — tnetwork documentation, 2022. URL https://tnetwork.readthedocs.io/en/latest/documentation.html. 9, 32

[78] Tanja Falkowski, Jorg Bartelheimer, and Myra Spiliopoulou. Mining and visualizing the evolution of subgroups in social networks. In IEEE/WIC/ACM International Conference on Web Intelligence, pages 52–58. IEEE, 2006. 9, 32, 33, 34

[79] Chonghui Guo, Jiajia Wang, and Zhen Zhang. Evolutionary community structure discovery in dynamic weighted networks. Physica A: Statistical Mechanics and its Applications, 413: 565–576, 2014. 9, 32, 33, 34

[80] Abraham Savitzky and Marcel JE Golay. Smoothing and differentiation of data by simplified least squares procedures. Analytical chemistry, 36(8):1627–1639, 1964. 9

[81] Jeffrey A Fessler and Raj Rao Nadakuditi. Linear Algebra for Data Science, Machine Learning, and Signal Processing. Cambridge University Press, 2024. 18

[82] Dragoš Cvetković, Peter Rowlinson, and Slobodan K Simić. Signless laplacians of finite graphs. Linear Algebra and its applications, 423(1):155–171, 2007. 19

[83] Fritz Heider. Attitudes and cognitive organization. The Journal of psychology, 21(1):107–112, 1946. 20

[84] Mihai Cucuringu, Peter Davies, Aldo Glielmo, and Hemant Tyagi. Sponge: A generalized eigenproblem for clustering signed networks. In The 22nd International Conference on Artificial Intelligence and Statistics, pages 1088–1098. PMLR, 2019. 20

[85] Mark Goldberg, Stephen Kelley, Malik Magdon-Ismail, Konstantin Mertsalov, and Al Wallace. Finding overlapping communities in social networks. In IEEE International Conference on Social Computing, pages 104–113. IEEE, 2010. 22

[86] Fergal Reid, Aaron McDaid, and Neil Hurley. Partitioning breaks communities. Mining social networks and security informatics, pages 79–105, 2013. 22

[87] Janmenjoy Nayak, Bighnaraj Naik, and HSr Behera. Fuzzy c-means (fcm) clustering algorithm: a decade review from 2000 to 2014. In Proceedings of the International Conference on Computational Intelligence in Data Mining (CIDM), pages 133–149. Springer, 2015. 22

[88] Sanjeev Arora, Prabhakar Raghavan, and Satish Rao. Approximation schemes for euclidean k-medians and related problems. In Proceedings of the thirtieth annual ACM symposium on Theory of computing, pages 106–113, 1998. 22

[89] Edo M Airoldi, David Blei, Stephen Fienberg, and Eric Xing. Mixed membership stochastic blockmodels. Advances in neural information processing systems, 21, 2008. 22

[90] Liaoruo Wang, Tiancheng Lou, Jie Tang, and John E Hopcroft. Detecting community kernels in large social networks. In 2011 IEEE 11th International Conference on Data Mining, pages 784–793. IEEE, 2011. 23

[91] Roger Guimerà, DB Stouffer, Marta Sales-Pardo, EA Leicht, MEJ Newman, and Luis AN Amaral. Origin of compartmentalization in food webs. Ecology, 91(10):2941–2951, 2010. 23

[92] Gustavo Deco and Maurizio Corbetta. The dynamical balance of the brain at rest. The Neuroscientist, 17(1):107–123, 2011. 23

[93] Fragkiskos D Malliaros and Michalis Vazirgiannis. Clustering and community detection in directed networks: A survey. Physics reports, 533(4):95–142, 2013. 23, 24

[94] Maxwell Mirton Kessler. Bibliographic coupling between scientific papers. American documentation, 14(1):10–25, 1963. 23

[95] Henry Small. Co-citation in the scientific literature: A new measure of the relationship between two documents. Journal of the American Society for information Science, 24(4):265–269, 1973. 23

[96] Fan Chung. Laplacians and the cheeger inequality for directed graphs. Annals of Combinatorics, 9:1–19, 2005. 24

[97] David Gleich. Hierarchical directed spectral graph partitioning. Information Networks, 443, 2006. 24

[98] Mihai Cucuringu, Huan Li, He Sun, and Luca Zanetti. Hermitian matrices for clustering directed graphs: insights and applications. In Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics, pages 983–992. PMLR, June 2020. URL https://proceedings.mlr.press/v108/cucuringu20a.html. ISSN: 2640-3498. 24, 25

[99] Mojtaba Rezvani and Fazeleh Sadat Kazemian. A survey on hierarchical community detection in large-scale complex networks. AUT Journal of Mathematics and Computing, 3(2):173–184, 2022. 24

[100] Michael T Schaub, Jiaze Li, and Leto Peel. Hierarchical community structure in networks. Physical Review E, 107(5):054305, 2023. 26

[101] Vincent Cohen-Addad, Varun Kanade, Frederik Mallmann-Trenn, and Claire Mathieu. Hierarchical clustering: Objective functions and algorithms. Journal of the ACM (JACM), 66(4): 1–42, 2019. 26

[102] Juan Ignacio Perotti, Claudio Juan Tessone, and Guido Caldarelli. Hierarchical mutual information for the comparison of hierarchical community structures in complex networks. Physical Review E, 92(6):062825, 2015. 26

[103] Matteo Magnani, Obaida Hanteer, Roberto Interdonato, Luca Rossi, and Andrea Tagarelli. Community detection in multiplex networks. ACM Computing Surveys (CSUR), 54(3):1–35, 2021. 26

[104] Pedro Mercado, Antoine Gautier, Francesco Tudisco, and Matthias Hein. The power mean laplacian for multilayer graph clustering. In International Conference on Artificial Intelligence and Statistics, pages 1828–1838. PMLR, 2018. 27

[105] Yexin Zhang, Zhongtian Ma, Qiaosheng Zhang, Zhen Wang, and Xuelong Li. Community detection in the multi-view stochastic block model. arXiv preprint arXiv:2401.09510, 2024. 27

[106] A. Concas, S. Noschese, L. Reichel, and G. Rodriguez. A spectral method for bipartizing a network and detecting a large anti-community. Journal of Computational and Applied Mathematics, 373:112306, August 2020. doi: 10.1016/j.cam.2019.06.022. 28

[107] Vincent P Grande and Michael T Schaub. Disentangling the spectral properties of the hodge laplacian: not all small eigenvalues are equal. In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 9896–9900. IEEE, 2024. 28

[108] Tai Qin and Karl Rohe. Regularized spectral clustering under the degree-corrected stochastic blockmodel. Advances in neural information processing systems, 26, 2013. 28

[109] Mihai Cucuringu, Apoorv Vikram Singh, Déborah Sulem, and Hemant Tyagi. Regularized spectral methods for clustering signed networks. Journal of Machine Learning Research, 22 (264):1–79, 2021.

[110] Huan Qing and Jingli Wang. Regularized spectral clustering under the mixed membership stochastic block model. Neurocomputing, 550:126490, 2023.

[111] Abhishek Kumar, Piyush Rai, and Hal Daume. Co-regularized multi-view spectral clustering. Advances in neural information processing systems, 24, 2011. 28

[112] John Lee. Introduction to Smooth Manifolds. URL https://link.springer.com/book/10.1007/978-1-4419-9982-5. 28

[113] JEAN QUAINTANCE Gallier and Jocelyn Quaintance. Differential geometry and lie groups, volume 12. Springer, 2020. 28

[114] Lars Hagen and Andrew B Kahng. New spectral methods for ratio cut partitioning and clustering. IEEE transactions on computer-aided design of integrated circuits and systems, 11 (9):1074–1085, 1992. 30

[115] Gaoxia Wang, Yi Shen, and Ming Ouyang. A vector partitioning approach to detecting community structure in complex networks. Computers & Mathematics with Applications, 55 (12):2746–2752, 2008. 30

[116] Xiaochen He, Ruochen Zhang, and Bin Zhu. A generalized modularity for computing community structure in fully signed networks. Complexity, 2023(1):8767131, 2023. 31

[117] Vincenzo Nicosia, Giuseppe Mangioni, Vincenza Carchiolo, and Michele Malgeri. Extending the definition of modularity to directed graphs with overlapping communities. Journal of Statistical Mechanics: Theory and Experiment, 2009(03):P03024, 2009. 31

[118] Elizabeth A Leicht and Mark EJ Newman. Community structure in directed networks. Physical review letters, 100(11):118703, 2008. 31

[119] Michael J Barber. Modularity and community detection in bipartite networks. Physical Review E—Statistical, Nonlinear, and Soft Matter Physics, 76(6):066102, 2007. 31

[120] Jörg Reichardt and Stefan Bornholdt. Statistical mechanics of community detection. Physical Review E—Statistical, Nonlinear, and Soft Matter Physics, 74(1):016110, 2006. 31

[121] Remy Cazabet, Souâad Boudebza, and Giulio Rossetti. Evaluating community detection algorithms for progressively evolving graphs. Journal of Complex Networks, 8(6):cnaa027, 2020. 31, 32

[122] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. Proceedings of the national academy of sciences, 99(12):7821–7826, 2002. 31, 33

[123] College Football Data Team. College Football Data API Documentation. College Football Data, 2024. URL https://api.collegefootballdata.com/documentation. Accessed: 2024-10-26. 31

[124] Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a data set via the gap statistic. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 63(2):411–423, 2001. 33

[125] Ketan Rajshekhar Shahapure and Charles Nicholas. Cluster quality analysis using silhouette score. In 2020 IEEE 7th international conference on data science and advanced analytics (DSAA), pages 747–748. IEEE, 2020. 33

[126] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of computational and applied mathematics, 20:53–65, 1987. 33

[127] Dan Pelleg, Andrew Moore, et al. X-means: Extending k-means with efficient estimation of the number of clusters. In ICML'00, pages 727–734. Citeseer, 2000. 33

[128] Bahman Bahmani, Benjamin Moseley, Andrea Vattani, Ravi Kumar, and Sergei Vassilvitskii. Scalable k-means++. arXiv preprint arXiv:1203.6402, 2012. 33

[129] Michael Shindler, Alex Wong, and Adam Meyerson. Fast and accurate k-means for large datasets. Advances in neural information processing systems, 24, 2011.

[130] Greg Hamerly. Making k-means even faster. In Proceedings of the 2010 SIAM international conference on data mining, pages 130–140. SIAM, 2010. 33

Table of Contents

A Instantiations and Evaluations of the Proposed Framework

This section provides proofs, interpretations, and experiments expounding the discussion in Section 2.2. We begin with the straightforward proof of Proposition 1.

Proof of Proposition 1. First suppose $\mathscr{A}$ clusters based on the leading eigenvectors of $\boldsymbol{R}$. Since $|\lambda_d(\boldsymbol{R})| \leq \|\boldsymbol{R}\|_{\mathrm{F}}$, scaling by $1/\|\boldsymbol{R}\|_{\mathrm{F}}$ normalizes the spectrum of $\boldsymbol{R}$ to lie in $[-1, 1]$. Then the addition of $\boldsymbol{I}$ turns the matrix positive semidefinite, whence its singular vectors and leading eigenvectors agree. Now suppose $\mathscr{A}$ clusters based on the trailing eigenvectors of $\boldsymbol{R}$. Analogous to the above, division by $\|\boldsymbol{R}\|_{\mathrm{F}}$ normalizes the spectrum of $\boldsymbol{R}$ to lie in $[-1, 1]$. Then subtracting it from $\boldsymbol{I}$ turns the matrix positive semidefinite, such that the top singular vectors of the matrix $\boldsymbol{I} - \boldsymbol{R}/\|\boldsymbol{R}\|_{\mathrm{F}}$ agree with its leading eigenvectors, which in turn agree with the trailing eigenvectors of $\boldsymbol{R}$. □

We now proceed to discuss specific network modalities. The figures shown in this section are the numerical results used to create Figure 3 in the main text.

A.1 Simple Networks

Proof of Proposition 2. Since $\boldsymbol{L}$ is symmetric, its spectrum is real — say, $\lambda(\boldsymbol{L}) \in [\lambda_{\min}, \lambda_{\max}] \subset \mathbb{R}$. $\max|\lambda(\boldsymbol{L})|$ either equals $|\lambda_{\min}|$ or $|\lambda_{\max}|$; the spectrum of the matrix $\boldsymbol{L}/\max|\lambda(\boldsymbol{L})|$ therefore lives in $[-1, 1]$, and so we have $\lambda(n\boldsymbol{I} - \boldsymbol{L}) \subset [0, 2]$ for any $n \geq \max|\lambda(\boldsymbol{L})|$. Since this matrix is positive semidefinite, its top singular vectors and leading eigenvectors agree. Of course, its leading eigenvectors are precisely the trailing eigenvectors of $\boldsymbol{L}$.

For the second portion of the statement, let $n \geq 2\max_{i \in V} \deg(i)$. Given a node $i \in V$, notice that

$$\sum_{j \neq i} |L_{ij}| = \sum_{j \neq i} |-A_{ij}| = \deg(i).$$

By the Gershgorin circle theorem ([81], Section 8.5.2), the eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d \geq 0$ of $\boldsymbol{L}$ are located within the union of closed discs

$$\lambda_i \in \bigcup_{i=1}^{d} \{\lambda \in \mathbb{R} : |\lambda - \deg(i)| \leq \deg(i)\}$$

This implies $\lambda(\boldsymbol{L}) \subset [0, 2\max_{i \in V} \deg(i)]$ and thus $\lambda(\frac{\boldsymbol{L}}{n}) \subset [0, 1]$. By the same reasoning that concludes the paragraph above, the result follows. □

Ideal communities and subspace structure. We now elaborate on the special case of Proposition 2 alluded to in Section 2.2, wherein G consists of k cliques of s nodes each.

Permuting the node indices so that the graph Laplacian $\boldsymbol{L}$ of G is block diagonal, with each block corresponding to a clique and identical to each other block, computing the spectrum of $\boldsymbol{L}$ amounts to computing the spectrum corresponding to a single complete graph K_s of size s and copying that spectrum k times. The graph Laplacian $\boldsymbol{L}(K_s)$ of K_s is

$$\boldsymbol{L}_{K_s} = (s-1)\boldsymbol{I} - (\mathbf{1}\mathbf{1}^\top - \boldsymbol{I}) = s\boldsymbol{I} - \mathbf{1}\mathbf{1}^\top,$$

where $\mathbf{1}$ denotes the vector of all ones. As with any graph Laplacian, the vector $\mathbf{1}$ is an eigenvector with eigenvalue 0. Any other eigenvector $\boldsymbol{v}$ (of which there are $s-1$) must be orthogonal to this one, implying that

$$\mathbf{1}\mathbf{1}^\top \boldsymbol{v} = \mathbf{1}\left(\sum_j v_j\right) = \mathbf{1} \cdot 0 = 0.$$

It follows that

$$\boldsymbol{L}_{K_s}\boldsymbol{v} = s\boldsymbol{I}\boldsymbol{v} - \mathbf{1}\mathbf{1}^\top \boldsymbol{v} = s\boldsymbol{v},$$

i.e., $\boldsymbol{v}$ has eigenvalue s. Putting together each spectrum, we obtain

$$\lambda(\boldsymbol{L}) = (\overbrace{0, \ldots, 0}^{k \text{ times}}, \overbrace{s, \ldots, s}^{d-k \text{ times}}).$$

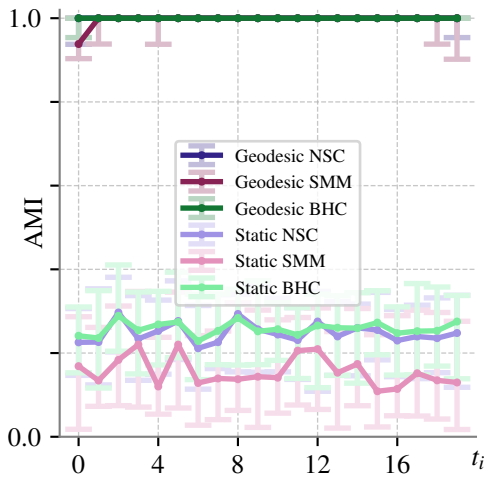


Figure 5. Representative comparison of geodesic and static methods for dynamic community detection in simple networks, medianed over 50 simulations of the dynamic stochastic block model from Section 3 ($d = 120$, $T = 20$, $p_{\text{switch}} = 10^{-2}$, $p_{\text{in}} = 0.3$, $p_{\text{out}} = 0.2$). Error bars correspond to 25th and 75th percentiles. With the exception of spectral modularity maximization at time $i = 1$, the geodesic methods each median to recover the true community structure of each snapshot. The methods compared are normalized spectral clustering (NSC) [15], spectral modularity maximization (SMM) [14], and Bethe Hessian clustering (BHC) [17].

Then the matrix $\boldsymbol{M} = s\boldsymbol{I} - \boldsymbol{L}$ has spectrum

$$\lambda(\boldsymbol{M}) = s - \lambda(\boldsymbol{L}) = (\overbrace{s, \ldots, s}^{k \text{ times}}, \overbrace{0, \ldots, 0}^{d-k \text{ times}}).$$

The result is that the MCM $\boldsymbol{M}$ has rank-k, meaning that in this special case the seemingly high-dimensional node embeddings it contains as columns already live in a low-dimensional latent subspace. This lends extra intuition to the formulation of spectral clustering as a subspace estimation problem (Algorithm 1): in the case of a graph with entirely unambiguous community structure, the embeddings from $\boldsymbol{M}$ live in a k-dimensional subspace. We can then view small perturbations of this ideal graph structure — the random removal of edges between intracluster nodes and placement of edges between interclique nodes — as adding spatial noise to the original node embeddings living in the subspace, which it is then the goal of spectral clustering to remove (Algorithm 1) via low-rank approximation.

Proof of Proposition 3. For a discussion on signless Laplacians, see [82]. The only property needed here is that, like $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$, $\boldsymbol{Q} = \boldsymbol{D} + \boldsymbol{A}$ is positive semidefinite. It follows that the matrices

$$\boldsymbol{L}^{\text{sym}} = \boldsymbol{D}^{-1/2}(\boldsymbol{D} - \boldsymbol{A})\boldsymbol{D}^{-1/2} \text{ and } \boldsymbol{Q}^{\text{sym}} = \boldsymbol{D}^{-1/2}(\boldsymbol{D} + \boldsymbol{A})\boldsymbol{D}^{-1/2} = 2\boldsymbol{I} - \boldsymbol{L}^{\text{sym}} \tag{12}$$

are each positive semidefinite. So,

$$\lambda(\boldsymbol{L}^{\text{sym}}) \subset \mathbb{R}_{\geq 0} \text{ and } 2 - \lambda(\boldsymbol{L}^{\text{sym}}) \subset \mathbb{R}_{\geq 0}, \tag{13}$$

which constrains $\lambda(\boldsymbol{L}^{\text{sym}})$ to the interval $[0, 2]$. It follows that $\lambda(\frac{1}{2}\boldsymbol{L}_{\text{sym}}) \subset [0, 1]$, and in turn that $\lambda(\boldsymbol{I} - \frac{1}{2}\boldsymbol{L}^{\text{sym}}) \subset [0, 1]$. In particular, $\boldsymbol{I} - \frac{1}{2}\boldsymbol{L}^{\text{sym}}$ is positive semidefinite, hence its top singular vectors $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k$ agree with its leading eigenvectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k$. These are precisely the k trailing eigenvectors of $\frac{1}{2}\boldsymbol{L}^{\text{sym}}$, and precisely the top singular vectors of $\frac{1}{2}\boldsymbol{Q}^{\text{sym}}$; the result follows. □

Bethe Hessian clustering. There is one additional static spectral method for clustering simple networks whose discussion is largely omitted from the main text. Bethe Hessian clustering [17] generalizes spectral clustering by replacing the graph Laplacian with a regularized generalization $\boldsymbol{H}(r) = (r^2 - 1)\boldsymbol{I} - r\boldsymbol{A} + \boldsymbol{D}$. We set $r = \sqrt{c}$, where c is the mean degree of G, per the suggestion in [17]. Unlike the graph Laplacian, the Bethe Hessian is not in general positive semidefinite. Like the modularity matrix $\boldsymbol{B}$, the most positive and most negative eigenvectors in fact each contain important information about (anti)community structure. Here, we consider Bethe Hessian clustering (BHC) via the trailing eigenvectors of $\boldsymbol{H}(r)$, with MCM obtained (as always when no alternative is specified) via Proposition 1.

A.2 Signed Networks

Let $G = (V, E^+, E^-)$ be a graph comprised of both positive (attractive) and negative (repulsive) edges, such as voting networks [60] or the Pearson correlation networks ubiquitous in time series

analysis [59]. The study of community detection in signed networks has its roots in structural balance theory [83]; the goal is to obtain a partition under which increased positive edges exist within communities and increased negative edges exist between communities [84].

Signed networks — algorithms and modeled clustering matrices. Two generalizations of spectral clustering to signed networks are based on trailing eigenvectors of the *signed ratio Laplacian*

$$|\boldsymbol{D}| - \boldsymbol{A}, \tag{14}$$

where $|\boldsymbol{D}|$ is the entrywise absolute value of $\boldsymbol{D}$ [26], and the *geometric mean Laplacian* [25][6]

$$\boldsymbol{L}^{+^{1/2}}(\boldsymbol{L}^{+^{-1/2}}\boldsymbol{Q}^{-}\boldsymbol{L}^{+^{-1/2}})^{1/2}\boldsymbol{L}^{+^{1/2}} \tag{15}$$

(or their normalized counterparts). More generally, choose $p \in \mathbb{R}$. Set $k' := k-1$ if $p \geq 1$ and $k' = k$ if $p < 1$. The approach in [24] subsumes both of the aforementioned approaches as the special cases $p = 1$ and $p \to 0$ of applying k-means to the k' smallest eigenvectors of the *signed power mean p-Laplacian*

$$\boldsymbol{L}(p) := \boldsymbol{M}_p(\boldsymbol{L}^{\text{sym}+}, \boldsymbol{Q}^{\text{sym}-}), \tag{16}$$

where $\boldsymbol{M}_p(\boldsymbol{A}, \boldsymbol{B})$ denotes the matrix power mean $\boldsymbol{M}_p(\boldsymbol{A}, \boldsymbol{B}) := (\frac{\boldsymbol{A}^p + \boldsymbol{B}^p}{2})^{1/p}$.[7]

Proposition 5. *Let $\boldsymbol{U}_{k'}\boldsymbol{\Sigma}_{k'}\boldsymbol{V}_{k'}^{\top}$ be a truncated singular value decomposition of $\overline{\boldsymbol{L}}(p) := \boldsymbol{I} - \frac{1}{2}\boldsymbol{M}_p(\boldsymbol{L}^{sym+}, \boldsymbol{Q}^{sym-})$. Take the leading singular vectors $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{k'}$ as node embeddings. Then the matrix serves as an MCM for the three algorithms described here.*

Proof of Proposition 5. $M_p(\boldsymbol{L}^{\text{sym}+}, \boldsymbol{Q}^{\text{sym}-})$ is positive semidefinite as a combination of sums, powers, and positive scalings thereof. Hence $\lambda(M_p(\boldsymbol{L}^{\text{sym}+}, \boldsymbol{Q}^{\text{sym}-})) \subset \mathbb{R}_{\geq 0}$. Also, we can bound its leading eigenvalue as:

$$\begin{aligned}
\lambda_1(M_p(\boldsymbol{L}^{\text{sym}+}, \boldsymbol{Q}^{\text{sym}-})) =& \lambda_1\left(\left(\frac{\boldsymbol{L}^{\text{sym}+^p} + \boldsymbol{Q}^{\text{sym}-^p}}{2}\right)^{1/p}\right) && (17)\\
=& \frac{1}{2}\lambda_1^{1/p}\left(\boldsymbol{L}^{\text{sym}+^p} + \boldsymbol{Q}^{\text{sym}-^p}\right) && (18)\\
\leq& \frac{1}{2}\lambda_1^{1/p}(\boldsymbol{L}^{\text{sym}+^p}) + \frac{1}{2}\lambda_1^{1/p}(\boldsymbol{Q}^{\text{sym}-^p}) && (19)\\
=& \frac{1}{2}\lambda_1^{p/p}(\boldsymbol{L}^{\text{sym}+}) + \frac{1}{2}\lambda_1^{p/p}(\boldsymbol{Q}^{\text{sym}-}) && (20)\\
\leq& \frac{1}{2}(2) + \frac{1}{2}(2) && (21)\\
=& 2, && (22)
\end{aligned}$$

where the inequality in (19) follows from Weyl's inequality and the inequality in (21) follows from the fact that both $\lambda(\boldsymbol{L}^{\text{sym}+}) \subset [0,2]$ and $\lambda(\boldsymbol{Q}^{\text{sym}-}) \subset [0,2]$ (cf. Proposition 3). Hence $\lambda(M_p(\boldsymbol{L}^{\text{sym}+}, \boldsymbol{Q}^{\text{sym}-})) \subset \mathbb{R}_{\leq 2}$. Thus $\lambda(M_p(\boldsymbol{L}^{\text{sym}+}, \boldsymbol{Q}^{\text{sym}-})) \subset [0,2]$. Or, if $p < 0$ forces us to spectrum-shift by $\varepsilon > 0$, $[\varepsilon, 2+\varepsilon]$. In turn, $\lambda\left(\frac{1}{2}M_p(\boldsymbol{L}^{\text{sym}+}, \boldsymbol{Q}^{\text{sym}-})\right) \subset [0,1]$; it follows that the matrix $\boldsymbol{I} - \frac{1}{2}M_p(\boldsymbol{L}^{\text{sym}+}, \boldsymbol{Q}^{\text{sym}-})$ is positive semidefinite and hence its top singular vectors and top eigenvectors agree. But its top eigenvectors are precisely the bottom eigenvectors of $\frac{1}{2}M_p(\boldsymbol{L}^{\text{sym}+}, \boldsymbol{Q}^{\text{sym}-})$; the result follows. □

Signed networks — experiments. We evaluate on a dynamic model, given in Algorithm 3, based on the signed stochastic block model found in [24, 84] and the standard dynamic stochastic block model from Section 3. The results are shown in Figure 6.

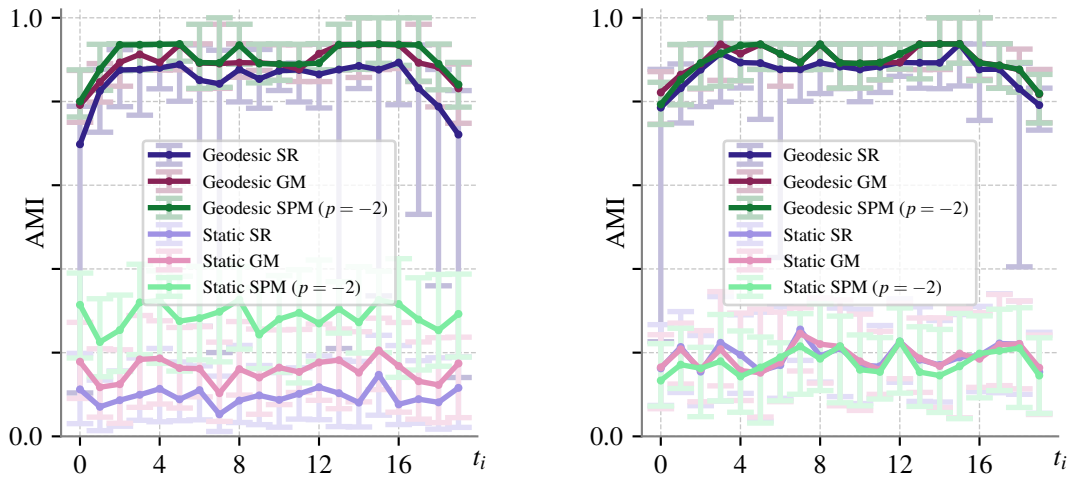
[6] Here, the $+$ and $-$ exponents refer to quantities defined over the unsigned networks $G^+ = (V, E^+)$ and $G^- = (V, E^-)$ respectively.

[7] When $p < 0$ the matrix power mean requires positive definite matrices; the authors address this by considering $\boldsymbol{L}^{\text{sym}+} + \varepsilon\boldsymbol{I}$ and $\boldsymbol{Q}^{\text{sym}-} + \varepsilon\boldsymbol{I}$ for some $\varepsilon > 0$ as necessary.

Algorithm 3 Dynamic Signed Stochastic Block Model (Dynamic SSBM)

Input: Probabilities p_{in}, p_{out}, η_{in}, η_{out}, p_{switch}, number k of planted communities

1: **At time** $i = 1$**:**
2: Partition node set (up to remainder) into k equally-sized planted communities. Let $Z(\ell)$ denote the community to which node ℓ belongs.
3: For every node pair (i, j), place a positive edge $(+1)$ between i and j with probability p_{in} if $Z(i) = Z(j)$, and a negative edge (-1) with probability p_{out} if $Z(i) \neq Z(j)$. Note that, unlike the unsigned case, it is sensible to choose $p_{\text{in}} = p_{\text{out}}$.
4: *Flip* the sign of each placed edge according to respective probabilities η_{in} and η_{out} for $\eta_{\text{in}}, \eta_{\text{out}} < \frac{1}{2}$. Commonly $\eta_{\text{in}} = \eta_{\text{out}}$.
5: **Repeat the above for snapshot indices** $2 \leq i \leq T$. Dynamics are induced by asserting that, at a given time step i, each node (that has not before switched) switches community with probability p_{switch}, its destination chosen uniformly at random among the $k - 1$ options.



(a) $p_{\text{in}} > p_{\text{out}}$ **(b)** $p_{\text{in}} = p_{\text{out}}$

Figure 6. Comparison of geodesic and static signed community detection methods, medianed over 50 simulations of two settings of a dynamic signed stochastic block model ($d = 120$, $T = 20$, $k = 2$, $p_{\text{switch}} = 10^{-2}$, $\eta_{\text{in}} = \eta_{\text{out}} = 0.4$). (6a): $p_{\text{in}} = 0.3$, $p_{\text{out}} = 0.2$. (6b): $p_{\text{in}} = p_{\text{out}} = 0.3$: even when the intra- and inter-community connection probabilities are identical, the algorithms are still able to discover community structure based solely on positive and negative edge affinities. We compare algorithms based on the signed ratio Laplacian (SR) [26], the geometric mean Laplacian (GM) [25], and the more general matrix power mean Laplacian (SPM) [24] with $p = -2$.

A.3 Mixed-Membership Networks

Oftentimes a node belongs to more than one community, e.g., in many social [85, 86] and neuronal [62] networks. The output of an overlapping community detection algorithm generally takes the form of a $d \times k$ *membership matrix* whose $j\ell$th element equals the estimated probability that node j belongs to community ℓ. For the purposes of evaluation, this matrix is often thresholded into a binary matrix whose jth row indicates the communities to which the jth node belongs.

Mixed-membership networks — algorithms and modeled clustering matrices. A simple extension of spectral clustering to the context of overlapping communities is to replace the k-means step with a fuzzy c-means step. See [87] for a discussion on fuzzy c-means and [27] for analysis of the 'fuzzy spectral clustering' method it induces. Other mixed-membership spectral methods include [19], which compute a thin spectral decomposition $\boldsymbol{A} = \boldsymbol{V}_k \boldsymbol{\Lambda}_k \boldsymbol{V}_k^\top$ (eigenvalues in descending order), and defines node embeddings as $\boldsymbol{X} = \boldsymbol{V}_k \boldsymbol{\Lambda}_k^{1/2}$. After normalization and regularization, the authors apply k-medians clustering (as analyzed in [88]); projecting the rows of $\boldsymbol{X}$ onto the subspace spanned by cluster centers yields a final $d \times k$ matrix representing soft cluster memberships.

Mixed-Membership Networks — Experiments. We evaluate on a dynamic model based on the mixed-membership stochastic block model described in [89]. To understand the static mixed-membership stochastic block model, we will provide a new interpretation of the (single-membership) stochastic block model, and then generalize it to the mixed-membership case. Following this, we will extend to the dynamic case.

One can view the connectivity under the version of the stochastic block model from Section 3 in terms of the parameter matrix

$$\begin{bmatrix} p_{\text{in}} & p_{\text{out}} & \cdots & p_{\text{out}} \\ p_{\text{out}} & p_{\text{in}} & \cdots & p_{\text{out}} \\ \vdots & \vdots & \ddots & \vdots \\ p_{\text{out}} & p_{\text{out}} & \cdots & p_{\text{in}} \end{bmatrix} =: \boldsymbol{B} \in \mathbb{R}^{k \times k} \tag{23}$$

where each element of is a Bernoulli random variable with parameter p_{in} or p_{out}. Then, with $Z(\ell)$ denoting the planted community of a node ℓ, the probability $\mathbb{P}(A_{ij} = 1)$ of an edge between two nodes $i, j \in [d]$ is $B_{Z(i),Z(j)}$, which we can write as

$$\mathbb{P}(A_{ij} = 1) = B_{Z(i),Z(j)} = \phi_{i,Z(i)} \phi_{j,Z(j)} B_{Z(i),Z(j)} = \sum_{g=1}^{k} \sum_{h=1}^{k} \phi_{ig} \phi_{jh} B_{gh},$$

where $\boldsymbol{\phi}_\ell = [0 \cdots 0 \overbrace{1}^{g\text{th entry}} 0 \cdots 0]^\top \in \mathbb{R}^{k \times k}$ indicates that node ℓ belongs to the gth planted community with probability 1. A mixed-membership prescription then follows by allowing $\boldsymbol{\phi}_\ell$ to not be an indicator vector, but rather a normalized vector of probabilities wherein $\phi_{\ell g}$ describes the probability that node ℓ belongs to the gth community. We introduce simple dynamics into the model by asserting that, at each snapshot index, a node switches with probability p_{switch}, with its destination chosen uniformly at random from among all planted communities and their intersections. See Algorithm 4.

We evaluate using element-centric similarity as outlined in [75]. Although the overlapping community detection algorithms described above output community membership probabilities for each node, the element-centric similarity metric utilizes binary assignments. To be compatible, we threshold such that a node is declared part of a community if it belongs to that community with probably exceeding $p_{\text{thresh}} := 0.2$. Results are in Figure 7.

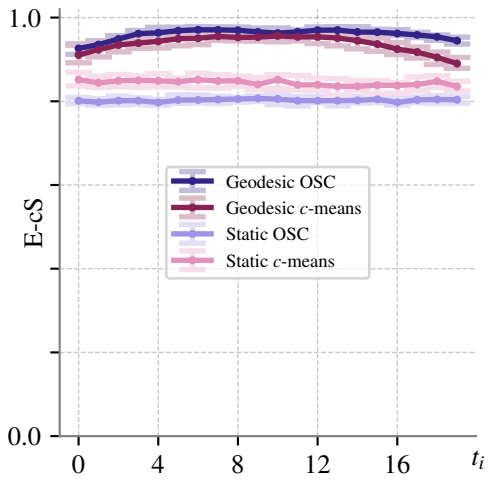


Figure 7. Comparison of geodesic and static overlapping community detection methods in time, medianed over 50 simulations of a dynamic mixed-membership stochastic block model ($d = 120$, $k = 2$, $T = 20$, $p_{\text{switch}} = 10^{-2}$, $p_{\text{in}} = 0.3$, $p_{\text{out}} = 0.2$). At time $i = 1$, two communities are planted, each possessing 50 'single-affiliated' nodes. The remaining 20 nodes are equally affiliated with each community. That is, $\boldsymbol{\phi}_1 = [1 \ \cdots \ 1 \ \ 0.5 \ \cdots \ 0.5 \ \ 0 \ \cdots \ 0]^\top$ and $\boldsymbol{\phi}_2 = [0 \ \cdots \ 0 \ \ 0.5 \ \cdots \ 0.5 \ \ 1 \ \cdots \ 1]^\top$. Error bars correspond to 25th and 75th percentiles. The algorithms compared are the overlapping spectral clustering (OSC) method of [19] and c-means spectral clustering based on [27].

Algorithm 4 Dynamic Mixed-Membership Stochastic Block Model

Input: Number of communities k, number of nodes d, number of time points T, parameter matrix $\boldsymbol{B}$, switching probability p_{switch}, initial mixed-membership vectors $\boldsymbol{\Phi} \in \mathbb{R}^{d \times k}$ with columns $\{\boldsymbol{\phi}_\ell\}_{\ell=1}^d \subset \mathbb{R}^k$

1: **At time** $i = 1$**:**
2: Use provided $\boldsymbol{\phi}_\ell \in \mathbb{R}^k$ for each node $\ell \in [d]$
3: Sample an adjacency matrix A, $\mathbb{P}(A_{ij} = 1) = \sum_{g=1}^k \sum_{h=1}^k \phi_{ig}\phi_{jh}B_{gh}$
4: **Repeat the above for** $2 \leq i \leq T$. Dynamics are induced as follows: at a given time step i, each node switches mixed-membership vector with probability p_{switch}, its new vector chosen uniformly at random from among the unique columns of $\boldsymbol{\Phi}$.

A.4 Directed Networks

A great deal of network modalities lend themselves to asymmetric representations, including many social [90], informational [64], biological [63, 91] and neuroscientific [92] networks. We consider two notions of 'community' which naturally arise in this directed network context. The first aligns with the undirected case: a good partition divides the network into communities such that intracommunity edge densities far exceed intercommunity edge densities. The second is especially pertinent to the directed case, and concerns the grouping of nodes based on 'patterns' rather than edge density. These include co-citation patterns (outgoing resp. incoming edges on in-community nodes are more likely to share common targets resp. sources) and flow-based patterns (a random walker is more likely to get 'stuck' in a community) [93].

Directed Networks — algorithms and modeled clustering matrices. One popular extension of spectral methods to the directed context is based on replacement of the directed adjacency matrix $\boldsymbol{A}$ with an appropriate symmetrization. Two such symmetrizations are evaluated in [28] in the context of spectral clustering. The first is *bibliographic symmetrization*:

$$\boldsymbol{A}^{\text{bibliographic}} = \boldsymbol{A}\boldsymbol{A}^\top + \boldsymbol{A}^\top\boldsymbol{A}, \tag{24}$$

where the *bibliographic coupling matrix* $\boldsymbol{A}\boldsymbol{A}^\top$ counts the number of nodes in G to which i and j both point [94] and the *co-citation matrix* $\boldsymbol{A}^\top\boldsymbol{A}$ counts the number of nodes in G that point to both i and j [95]. The second is *degree-discounted symmetrization*:

$$\boldsymbol{A}^{\text{degree-discounted}} := \boldsymbol{D}_{\text{out}}^{-1/2}\boldsymbol{A}\boldsymbol{D}_{\text{in}}^{-1/2}\boldsymbol{A}^\top\boldsymbol{D}_{\text{out}}^{-1/2} + \boldsymbol{D}_{\text{in}}^{-1/2}\boldsymbol{A}^\top\boldsymbol{D}_{\text{out}}^{-1/2}\boldsymbol{A}\boldsymbol{D}_{\text{in}}^{-1/2}, \tag{25}$$

which normalizes (24) to account for the heterogeneity of degree distributions found in real-world networks [28]. Also studied are Laplacian extensions for directed graphs, the most popular among these

[29, 93, 96, 97] being the random walk-based directed Laplacian (RW) whose *largest* eigenvectors encode graph partition structure

$$\boldsymbol{\Theta} = \frac{1}{2}\left(\boldsymbol{\Pi}^{1/2}\boldsymbol{P}\boldsymbol{\Pi}^{-1/2} + \boldsymbol{\Pi}^{-1/2}\boldsymbol{P}^{\top}\boldsymbol{\Pi}^{1/2}\right), \tag{26}$$

where $\boldsymbol{P}$ is the transition matrix $P_{ij} = \frac{A_{ij}}{\sum_{j=1}^{d} A_{ij}}$ and $\boldsymbol{\Pi} = \boldsymbol{D}_{\text{out}}^{-1}$.

We note that the aforementioned symmetrization approaches are intended to detect both density-based and pattern-based communities, whereas the directed Laplacian-based approaches are intended only to detect density-based communities ([93], Table 2).

Directed networks — experiments. We evaluate on following dynamic model (Algorithm 5), based on the directed stochastic block model proposed in [98]. Results are shown in Figure 8.

Algorithm 5 Dynamic Directed Stochastic Block Model

Input: probabilities p_{in}, p_{out}, p_{switch}, matrix $\boldsymbol{F} \in [0,1]^{k\times k}$ satisfying $F_{\ell j} + F_{j\ell} = 1$ for all $j, \ell \in [k]$, number k of planted communities

1: **At time** $i = 1$**:**
2: Partition node set (up to remainder) into k equally-sized planted communities. Let $Z(\ell)$ denote the community to which the node ℓ belongs
3: For every node pair (i, j), place an edge between i and j with probability p_{in} if $Z(i) = Z(j)$ and probability p_{out} if $Z(i) \neq Z(j)$. The edge is directed from i to j with probability $F_{Z(i),Z(j)}$; otherwise, it is directed from j to i.
4: **Repeat the above for snapshot indices** $2 \leq i \leq T$. Dynamics are induced by asserting that, at a given time step i, each node (that has not before switched) switches community with probability p_{switch}, its destination chosen uniformly at random among the $k-1$ options.

A.5 Networks with Cocommunity or Bipartite Structure

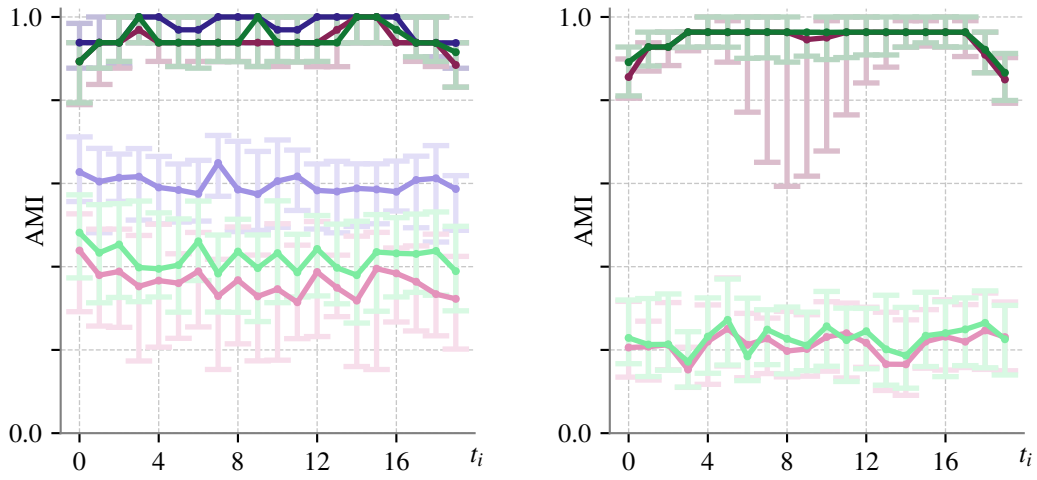
An adjacent problem to that of directed network community detection is that of co-community detection. The general coclustering problem concerns the grouping of data according to multiple attributes (e.g., samples and features) simultaneously. We focus on two notions of cocommunity detection found in network analysis. The first notion applies to directed networks, where the two attributes correspond to rows and columns of the directed adjacency matrix. The output is two network partitions: one which clusters nodes with similar sending patterns, and another which clusters nodes with similar receiving patterns. The second notion applies to bipartite networks, where the goal is to cluster nodes of both types simultaneously. The output is two coindexed partitions of first-type nodes and second-type nodes respectively. The cocommunity detection method we generalize via Algorithm 2 applies to both notions.

Algorithms and modeled clustering matrices. We will consider the spectral coclustering method found in [31]. A related algorithm may be found in [67]. Given a numbers k_y and k_z of sending clusters and receiving clusters to detect, the method computes a k-truncated singular value decomposition $\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\top}$ of a regularized (asymmetric) graph Laplacian $\boldsymbol{L}$, where $k = \min(k_y, k_z)$. $\boldsymbol{U}$ is then used for 'sending embeddings', while $\boldsymbol{V}$ is used for 'receiving embeddings'. From the perspective of Algorithm 1, $\boldsymbol{L}$ is already in MCM form when the goal is to detect receiving clusters. When the goal is to detect sending clusters, $\boldsymbol{L}^{\top}$ has the requisite form.

Experiments. We evaluate with a dynamic model, given in Algorithm 6, based on the dynamic stochastic coblock model from [31]. Results are given in Figure 9.

A.6 Hierarchical Networks

A great deal of real-world complex systems exhibit interesting behavior at multiple resolutions. Examples of detecting hierarchical communities within such systems abound in neuroscience [69], biochemistry [68], and the social sciences [99]. The output of a hierarchical community detection algorithm is a rooted tree whose vertices represent communities, edges represent parent-child relationships, and levels represent scales of resolution.



(a) Density-based clustering

(b) Pattern-based clustering

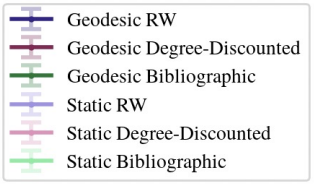


Figure 8. Comparison of geodesic and static methods medianed over 50 simulations of two settings of a dynamic directed stochastic block model, both with $d = 120$, $T = 20$, $p_{\text{switch}} = 10^{-2}$. (8a) $k = 2$ communities are planted with $\boldsymbol{F} = [0.5 \quad 0.4; 0.6 \quad 0.5]$, $p_{\text{in}} = 0.3$, $p_{\text{out}} = 0.2$ in a 'density-based' parameter setting. (8b) $k = 3$ communities are planted with $\boldsymbol{F} = \begin{bmatrix} 1/2 & 2/3 & 1/3 \\ 1/3 & 1/2 & 2/3 \\ 2/3 & 1/3 & 1/2 \end{bmatrix}$, $p_{\text{in}} = p_{\text{out}} = 0.2$ in a 'flow/pattern-based' parameter setting [98].

Algorithm 6 Dynamic Stochastic Coblock Model

Input: Number k_y of sending communities, number k_z of receiving communities; $\boldsymbol{B} \in \mathbb{R}^{k_y \times k_z}$ where $B_{\ell\ell'} = p_{\ell\ell'} \in [0, 1]$ represents the probability of an edge existing from a node in sending cluster $\ell \in [k_y]$ to a node in receiving cluster $\ell' \in [k_z]$. Switching probabilities $p_{\text{switch, send}}$ and $p_{\text{switch, receive}}$.

1: If the graph is bipartite with node sets V_1 and V_2, declare V_1 the sending node set and V_2 the receiving node set. If the graph is not bipartite, both sending and receiving node sets are the full node set $V = V_1 = V_2$.
2: Assign each node in the sending node set V_1 to one of the k_y sending communities, divided equally. Similarly, assign each node in the receiving node set to one of the k_z receiving communities, divided equally.
3: For each node pair (i, j), place a (directed) edge from node i to node j with probability B_{y_i, z_j}, where y_i is the sending community of i and z_j is the receiving community of j.
4: **Repeat for time steps** $2 \leq i \leq T$**:**
5: At each time step i, for each node in V_1, with probability $p_{\text{switch, send}}$, swap its sending community assignment with a randomly chosen node from a uniformly random different sending community.
6: Similarly, for each node in V_2, with probability $p_{\text{switch, receive}}$, swap its receiving community assignment with a randomly chosen node from a uniformly random different receiving community.
7: After community assignments are updated, regenerate all edges according to step 3.

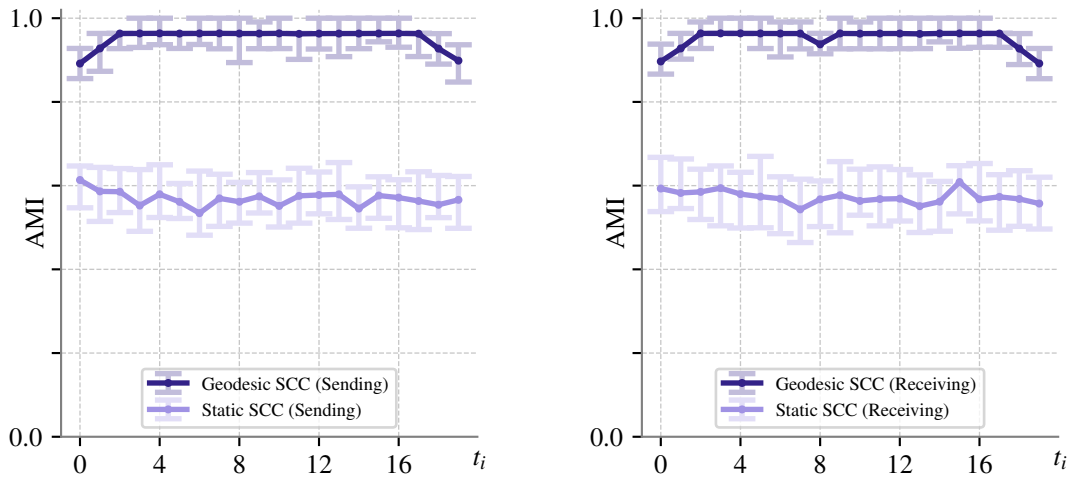


Figure 9. Comparison of geodesic and static versions of the spectral coclustering (SCC) method in [31] over 50 simulations of the dynamic stochastic coblock model ($d = 120$, $T = 20$, $p_{\text{switch, send}} = p_{\text{receive, send}} = 10^{-2}$, $k_y = k_z = 3$, $\boldsymbol{B} = \begin{bmatrix} 0.5 & 0.3 & 0.3 \\ 0.3 & 0.5 & 0.3 \\ 0.3 & 0.3 & 0.5 \end{bmatrix}$). Ribbons correspond to 25th and 75th percentiles.

Hierarchical networks — Algorithms and modeled clustering matrices. We explore the approach to hierachical community detection described in [32], which uses Laplacian eigenvectors to obtain the coarsest partition, then employs a degree-bucketing approach to unfold the community hierarchy. Another spectral method for hierarchical community detection (not included in our experiments) may be found in [100].

Hierarchical networks — experiments. Evaluation is performed on a dynamic model, given in Algorithm 7, based on the hierarchical stochastic block model proposed in [101]. We score according to the hierarchial normalized mutual information metric proposed in [102]. Results are found in Figure 10.

Algorithm 7 Dynamic Hierarchical Stochastic Block Model

Input: A rooted tree on L leaves consisting of M nodes with weights $p_m \in [0, 1]$, $m \in [M]$, wherein each node corresponds to a planted community, and each level of the tree corresponds to a level of hierarchy in the network. Note that the weight assigned to the root node coincides with p_{out} in the stochastic block model described in Section 3. Number T of time steps. Probability p_{switch}.

1: **At time** $i = 1$**:**
2: Assign d nodes equally (up to remainder) among the L leaves
3: For every pair of nodes i, j belonging to leaves $L(i)$, $L(j)$ respectively, place an edge with probability p_m, where m denotes the least common ancestor of $L(i)$ and $L(j)$
4: **For time steps** $2 \leq i \leq T$**, repeat the above.** Dynamics are induced as follows: at a given time step i, each node (that has not before switched) switches leaf community to one of its siblings with probability p_{switch}, its destination chosen uniformly at random.

A.7 Multiview Networks

In practice, one often has access to multiple graphs corresponding to the same network. Experiments used to construct network representations of data often have many trials [65]. Or there may exist many modes of relationship between the same set of nodes, each yielding its own 'view' of the network — [103]. Like snapshot-represented dynamic networks, the underlying representation of

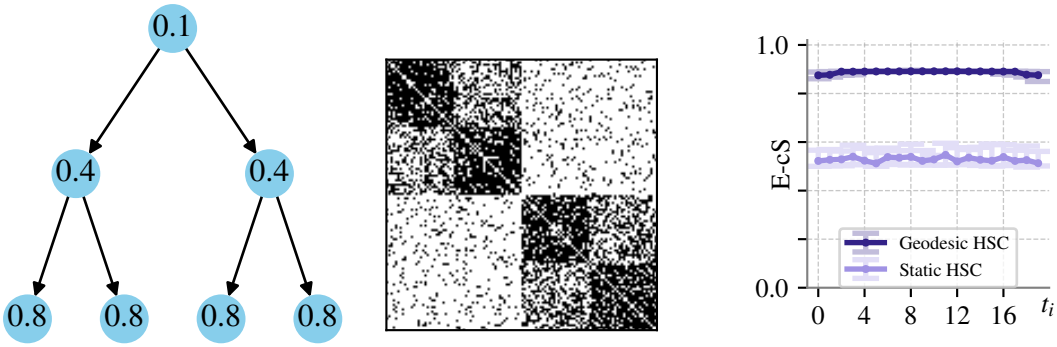


Figure 10. LEFT: An example probability tree input into the dynamic hierarchical stochastic block model. MIDDLE: An adjacency matrix, sampled from the dynamic HSBM according to this tree. RIGHT: Comparison of geodesic and static versions of the hierarchical spectral clustering (HSC) method in [32], medianed over 50 simulations of the dynamic HSBM ($d = 120$, $T = 20$, $p_{\text{switch}} = 10^{-2}$, probability tree mimicking that displayed but with values in 0.4 ± 0.05 to introduce noise.) Error bars correspond to 25th and 75th percentiles.

a multiview network is generally a multiplex graph. Unlike dynamic networks, the goal is not to detect one community partition per layer, but rather to construct a single partition of the nodes using information from multiple layers. That said, the two problem settings can interact. For instance, suppose the observed dynamic simple graph $\{G_i\}_{i=1}^T$ of Section A.1 is of high temporal resolution, in the sense that T is very large but the network evolution between adjacent snapshots is minuscule, perhaps with the exception of some outliers of high temporal discontinuity. By segmenting the snapshots into 'windows', a dynamic multiview network can be created, and applying a dynamic multiview community detection to this network might capture a coarsened perspective on the unfolding community structure, perhaps while 'smoothening away' the outliers.

Multiview networks — algorithms and modeled clustering matrices. We discuss the dynamic generalization of two spectral methods for detecting communities in multiview networks, each based on analyzing the spectrum of a single 'summary Laplacian' computed using the graph Laplacians from the individual layer. The first approach is directly adjacent to the signed power mean Laplacian method discussed in Section A.2, where the spectrum of the power mean of S Laplacians — one per view of the network — is considered [104]. The second approach is based on Grassmann manifold geometry [30]: the authors look at Laplacian spectrums layer-by-layer to obtain a collection of subspaces $\{\langle \boldsymbol{U}_i \rangle\}_{i=1}^S$, then solve a Riemannian optimization problem on the Grassmann manifold for combining the $\langle \boldsymbol{U}_i \rangle$ into a single representative subspace $\langle \boldsymbol{U} \rangle$ that is 'close' to each $\langle \boldsymbol{U}_i \rangle$. They obtain from this analysis a single matrix whose spectrum aims to summarize the clustering structure across all layers.

Multiview networks — experiments. We evaluate with a dynamic model, given in Algorithm 8, based on a simple setting of the multiview stochastic block model analyzed in [105]. Results are found in Figure 11.

Algorithm 8 Dynamic Multiview Stochastic Block Model

Input: Number S of views, T time points, p_{in}, p_{out}, p_{switch}

1: **At time** $i = 1$**:**
2: Partition node set (up to remainder) into k equally-sized planted communities. Let $Z(\ell)$ denote the community to which node ℓ belongs.
3: Sample S adjacency matrices from the (static) stochastic block model described in Section 3 according to p_{in} and p_{out} and concatenate into an $S \times d \times d$ adjacency tensor
4: **Repeat the above for snapshot indices** $2 \leq i \leq T$. Dynamics are induced as follows: at a given time step i, each node (that has not before switched) switches community with probability p_{switch}, its destination chosen uniformly at random among the $k - 1$ options.

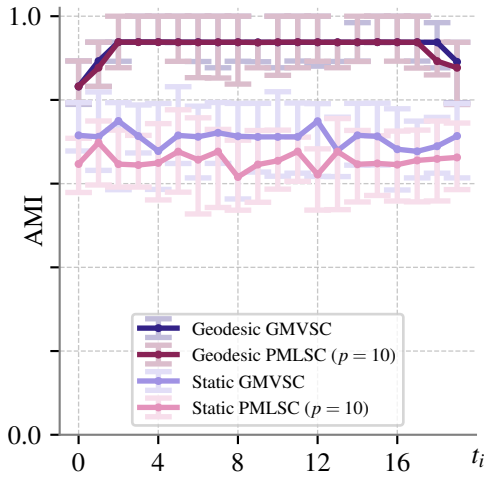


Figure 11. Comparison of geodesic and static multiview community detection methods in time, medianed over 50 simulations of a dynamic mixed-membership stochastic block model ($d = 120$, $k = 2$, $T = 20$, $p_{\text{switch}} = 10^{-2}$, $p_{\text{in}} = 0.3$, $p_{\text{out}} = 0.2$, $S = 3$). Error bars correspond to 25th and 75th percentiles. The algorithms compared are the Grassmannian multiview spectral clustering (GMVSC) algorithm of [30] and the power mean Laplacian spectral clustering (PMLSC) method of [24] with $p = 10$.

A.8 Other Network Modalities

Many successful methods for community detection in principle admit generalization via Algorithm 2, even though they are not included among our experiments. Examples include spectral methods for hypergraph clustering [33, 106], motif-based methods [34] and higher-order, topological methods [36, 107]. We also note that many of the non-regularized methods explored in this text have regularized counterparts which generalize via our method to the evolving setting as well [108–111].

B Proof of Proposition 4

Suppose $\sigma_1 \geq \sigma_2 > \sigma_3 = \cdots = \sigma_m = 0$, so that $\boldsymbol{X} = \boldsymbol{a}_1\sigma_1\boldsymbol{b}_1^\top + \boldsymbol{a}_2\sigma_2\boldsymbol{b}_2^\top$ and $\langle \boldsymbol{X} \rangle = \operatorname{span}(\boldsymbol{a}_1, \boldsymbol{a}_2) = \mathcal{U}$, where $\mathcal{U}$ is a 2-plane in $\mathbb{R}^d$ containing the origin. It follows that for each column $\boldsymbol{c}$ of $\boldsymbol{X}$ we have

$$\boldsymbol{c} \in \mathbb{S}^{d-1} \cap \mathcal{U},$$

so that $\boldsymbol{c}$ belongs to a great circle arc in $\mathbb{S}^{d-1}$. Hence the columns of $\boldsymbol{X}$ each live on the trace of a geodesic $\gamma : [0, 1] \to \mathbb{S}^{d-1}$. It suffices, then, to show that $\varphi \circ \gamma$ is a geodesic on the real projective space $\operatorname{Gr}(1, d)$, where $\varphi : \mathbb{S}^{d-1} \to \operatorname{Gr}(1, d)$ is given by $\varphi(\boldsymbol{w}) := \langle \boldsymbol{w} \rangle$.

To this end, let the Lie group $\mathbb{Z}/2\mathbb{Z} = \{1, -1\}$ act on $\mathbb{S}^{d-1}$ by (left) multiplication. The orbits of this action induce an equivalence relation ([37], Definition 9.13) whose associated projection $\pi : \mathbb{S}^{d-1} \to \mathbb{S}^{d-1}/(\mathbb{Z}/2\mathbb{Z})$ sends $\boldsymbol{w} \in \mathbb{S}^{d-1}$ to $[\boldsymbol{w}] = \{\boldsymbol{w}, -\boldsymbol{w}\}$. This action is clearly smooth, free, and — since $\mathbb{Z}/2\mathbb{Z}$ is compact as a finite topological group — it is proper ([112], Corollary 21.6). It is a local isometry as well; checking this amounts to verifying ([37], Theorem 9.38) that the antipodal map $f : \mathbb{S}^{d-1} \to \mathbb{S}^{d-1}$, $\boldsymbol{w} \mapsto -\boldsymbol{w}$ satisfies, for all $\boldsymbol{w} \in \mathbb{S}^{d-1}$ and $\boldsymbol{x}_1, \boldsymbol{x}_2 \in T_{\boldsymbol{w}}(\mathbb{S}^{d-1})$,

$$\langle df(\boldsymbol{w})[\boldsymbol{x}_1], df(\boldsymbol{w})[\boldsymbol{x}_2] \rangle_{f(\boldsymbol{w})} = \langle \boldsymbol{x}_1, \boldsymbol{x}_2 \rangle_{\boldsymbol{w}}.$$

Because $df(\boldsymbol{w}) = -\mathrm{id}_{\mathbb{S}^{d-1}}$, we can rewrite the desired assertion as

$$\langle -\boldsymbol{x}_1, -\boldsymbol{x}_2 \rangle_{-\boldsymbol{w}} = \langle \boldsymbol{x}_1, \boldsymbol{x}_2 \rangle_{-\boldsymbol{w}} = \langle \boldsymbol{x}_1, \boldsymbol{x}_2 \rangle_{\boldsymbol{w}};$$

since the Riemannian metric on $\mathbb{S}^{d-1}$ is inherited from the Euclidean inner product on $\mathbb{R}^d$, and the latter is invariant under orthogonal transformations, the equation holds. π is a therefore a Riemannian covering map ([113], Theorem 23.18), and hence ([113], Proposition 18.6) it projects and lifts geodesics to geodesics.

Moreover, recall that the canonical geodesic distance between two points on the Grassmannian is $\|\Theta\|_2$, where $\Theta \in \mathbb{R}^k$ consists of principal angles between the subspaces [56]. It follows immediately that the (clearly well-defined) diffeomorphism $\psi : \mathbb{S}^{d-1}/(\mathbb{Z}/2\mathbb{Z}) \to \operatorname{Gr}(1, d)$ mapping $[\boldsymbol{w}]$ to $\langle \boldsymbol{w} \rangle$ preserves geodesic distances.

Thus, φ factors as $\psi \circ \pi$, where ψ and π each preserve geodesics. It follows that

$$\{\varphi(\pm\boldsymbol{v}_1^{(1)}), \ldots, \varphi(\pm\boldsymbol{v}_1^{(d)})\} = \{\langle \pm\boldsymbol{v}_1^{(1)} \rangle, \ldots, \langle \pm\boldsymbol{v}_1^{(d)} \rangle\}$$

belongs to the trace of a Grassmann geodesic, as claimed.

Conversely, suppose the first singular subspaces $\langle \boldsymbol{v}_1^{(1)} \rangle, \ldots, \langle \boldsymbol{v}_1^{(T)} \rangle$ lie on the trace of a geodesic $\delta : [0, 1] \to \mathrm{Gr}(1, d)$. Since ψ is a diffeomorphism that preserves geodesic distances, ψ^{-1} is as well. Additionally, π lifts geodesics to geodesics as a Riemannian covering map. By lifting δ through ψ and π to geodesics on $\mathbb{S}^{d-1}$, we can realize each of the $\pm \boldsymbol{v}_1^{(i)}$ as living on the trace of a geodesic on $\mathbb{S}^{d-1}$, i.e., as living in the intersection of $\mathbb{S}^{d-1}$ with a 2-dimensional linear subspace $\mathcal{U}$. Hence $\sigma_3 = \cdots = \sigma_m = 0$.

Remark. In spectral clustering when $k_c = 2$, one also clusters based on the signs of an eigenvector's entries [47]. However, the most extremal eigenvector is the vector of all ones not interesting (indeed, it corresponds to the partition trivially minimizing the cut size objective by placing all nodes into one cluster[12]); instead, the second extremal eigenvector $\hat{\boldsymbol{u}}$ is used. All of the discussions phrased here in terms of spectral modularity maximization carry through for spectral clustering provided that one works with $\hat{\boldsymbol{u}}$ instead of the most extremal eigenvector $\boldsymbol{u}$.

C Extension to Time-Varying k_c

This section elaborates upon the straightforward extension, mentioned in Section 3, of Algorithm 2 to the case where the number of latent communities k_c varies in time. The approach described is applicable to any network modality for which an unsupervised benefit function (e.g. modularity or an extension of it) has been studied.

Section C.1 will motivate our extension. Section C.2 will provide an explicit algorithm for it, modeled off of Algorithm 2, together with empirical results. While the approach taken has some theoretical motivation, it is nevertheless a heuristic, and Section C.3 gives potential alternative heuristics and future directions toward a more principled approach.

C.1 Motivation

Static spectral methods for community detection generally require the number k_c of desired communities, the embedding dimension k_e, or both of these values to be specified in advance. The temporal setting amplifies this drawback, since in certain contexts the number of latent communities may earnestly change in time, as a result e.g. of merging or splitting [5]. This section justifies an extension of Algorithm 2 that is capable of automatically detecting a variable number of communities at each time step.

Our approach is motivated by the observation that, upon writing the $\mathrm{RatioCut}$ objective function for unnormalized spectral clustering in terms of spectrum $\lambda_1(\boldsymbol{M}), \ldots, \lambda_d(\boldsymbol{M})$ of the MCM $\boldsymbol{M} = n\boldsymbol{I} - \boldsymbol{L}$ (Proposition 2), the problem of minimizing $\mathrm{RatioCut}$ becomes equivalent to a flavor of *max-sum vector partitioning* applied to d-dimensional spectral embeddings. If we use k-dimensional spectral embeddings instead, the vector partitioning problem is approximately equivalent to $\mathrm{RatioCut}$, with error proportional to the energy lost by discarding $\lambda_{k+1}(\boldsymbol{M}), \ldots, \lambda_d(\boldsymbol{M})$. We use this observation to argue that, although letting the spectral embedding dimension k_e equal the desired number of communities k_c is conventional and effective for spectral partitioning, if k_e exceeds k_c by a small-to-moderate amount[8] then we should still expect good performance (since Euclidean clustering with a larger k_e is theoretically optimizing a function closer to the true $\mathrm{RatioCut}$ objective). The consequence is that, by choosing k_e in Algorithm 2 to be an upper bound for the estimated number $k_{c,i}$ of latent communities at any time step i, we are permitted to vary $k_c = k_c(t)$ freely as a function of time without penalty. The task then becomes deciding how to automatically choose each $k_{c,i} = k_c(t_i)$; we provide one approach but note that unexplored alternatives may do better (appendix C.3).

Minimal graph cuts and maximal vector partitions. The present discussion uses (unnormalized) spectral clustering as a prototypical example, but the argument we offer has analogues in (at least) the contexts of spectral modularity maximization [13] and size-contrained graph partitioning [11]. Said analogues have motivated broader efforts to better understand how large k_e should be compared to k_c in general spectral settings [55].

[8] Not by too much, however, in light of noise considerations and dimensionality curses.

Recall the definition of the *cut size objective* for evaluating a partition $(Z_1, \ldots, Z_{k_c})$ of a simple graph G with adjacency matrix $\boldsymbol{A}$ into k_c communities:

$$\mathrm{Cut}(Z_1, \ldots, Z_{k_c}) := \frac{1}{2}\sum_{i=1}^{k} W(Z_i, V \setminus Z_i), \text{ where } W(Z, Z') := \sum_{z \in Z, z' \in Z'} A_{zz'}, \tag{27}$$

and the $\mathrm{RatioCut}$ objective [114] whose relaxation yields (unnormalized) spectral clustering:

$$\mathrm{RatioCut}(Z_1, \ldots, Z_{k_c}) := \sum_{i=1}^{k_c} \frac{\mathrm{Cut}(Z_i, V \setminus Z_i)}{|Z_i|}. \tag{28}$$

Defining $\boldsymbol{S}$ to be the $d \times k$ community matrix $S_{ij} = \frac{\mathbb{1}_{Z(i)=Z(j)}}{\sqrt{|Z_j|}}$, it can be shown [47] that $\mathrm{RatioCut}$ may be rewritten

$$\mathrm{RatioCut}(Z_1, \ldots, Z_k) = \mathrm{Tr}(\boldsymbol{S}^\top \boldsymbol{L}\boldsymbol{S}), \tag{29}$$

where $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$ is the unnormalized graph Laplacian of G and $Z(\ell)$ denotes the community to which node ℓ belongs. With the MCM $\boldsymbol{M} = n\boldsymbol{I} - \boldsymbol{L}$ defined as in Proposition 2, we can rewrite this as

$$\mathrm{RatioCut}(Z_1, ..., Z_k) = n\,\mathrm{Tr}(\boldsymbol{S}^T\boldsymbol{S}) - \mathrm{Tr}(\boldsymbol{S}^T\boldsymbol{M}\boldsymbol{S}) = nk - \mathrm{Tr}(\boldsymbol{S}^T\boldsymbol{M}\boldsymbol{S}). \tag{30}$$

In particular, choosing $(Z_1, \ldots, Z_{k_c})$ minimizing $\mathrm{RatioCut}$ is equivalent to choosing $(Z_1, \ldots, Z_{k_c})$ maximizing $\mathrm{Tr}(\boldsymbol{S}^\top \boldsymbol{M}\boldsymbol{S})$.

Take a spectral decomposition $\boldsymbol{M} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^\top$, with the entries of $\boldsymbol{\Lambda}$ positioned in descending order, and define $\boldsymbol{N} := \boldsymbol{V}\boldsymbol{\Lambda}^{1/2}$. The ℓth row $\boldsymbol{r}_\ell$ of $\boldsymbol{N}$ equals the ℓth Euclidean node embedding derived by unnormalized spectral clustering when appropriately truncated, up to scalings from $\boldsymbol{\Lambda}^{1/2}$.[9] We can rewrite our objective in terms of these node embeddings:

$$\mathrm{Tr}(\boldsymbol{S}^\top \boldsymbol{M}\boldsymbol{S}) = \mathrm{Tr}(\boldsymbol{S}^T\boldsymbol{N}\,\boldsymbol{N}^T\boldsymbol{S}) = \sum_{j=1}^{k} \frac{1}{|Z_j|} \left\| \sum_{\ell \in Z_j} \boldsymbol{r}_\ell \right\|_2^2. \tag{31}$$

In this light, we see that minimizing $\mathrm{RatioCut}$ is equivalent to a flavor of *max-sum vector partitioning* applied to d-dimensional node embeddings derived from the leading eigenvectors of $\boldsymbol{M}$. Exactly solving max-sum vector partitioning problems is extremely expensive [13]; as such, various heuristics have been proposed [11, 13, 115], including ones that are nearly equivalent to k-means [13]. Truncating to $k_c \leq p \leq d$ rows of $\boldsymbol{N}$ yields a max-sum vector partitioning problem in $\mathbb{R}^p$ whose objective approximates the $\mathrm{RatioCut}$ objective. When $p = k_c$ we recover an algorithm very similar to spectral clustering. When $p = d$ we are optimizing the $\mathrm{RatioCut}$ objective exactly rather than a low-rank approximation to it — if doing so were computationally feasible, this would theoretically give the best results. So choosing an embedding dimension $p = k_e$ exceeding the number of desired clusters k_c is, in principle, an advantage rather than a detriment. In practice, due e.g. to noise considerations, dimensionality curses, and other challenges to which Euclidean clustering heuristics are generally susceptible in high-dimensional space, it is not recommended to let k_e be arbitrarily high relative to k_c.

We can utilize these ideas towards extending Algorithm 2 in the following manner. By definition, k_e must be fixed over time in Algorithm 2. However, nothing prevents $k_c = k_{c,i} = k_c(t_i)$ from varying based on $i \in [T]$, and — based on the discussion above — this should not incur any penalty compared to if we had chosen $k_e = k_{c,i}$ from the start, so long as $k_{c,i} \leq k_e$ and there exists earnest k_c-way community structure at time i.

Automatically choosing the number of communities. How should $k_{c,i}$ be chosen? The easiest approach would be to evaluate some unsupervised partition score (e.g., modularity or the appropriate extension to other network modalities) with respect to different choices of $k_{c,i}$ and choose the maximizer. Of course, evolving community detection algorithms are in part motivated by the premise that the 'correct' community partition at a given time step may not always be the one maximizing the static modularity (e.g., due to measurement noise or stability considerations [5, 8]). We therefore

[9] In fact, some versions of spectral clustering use $\boldsymbol{N}$ directly [19].

recommend incorporating some temporal awareness into the heuristic by filtering the scores for dynamic regularity, and reiterate our suggestion from Section 3 to initialize the Euclidean clustering heuristic of $\mathscr{A}$ with clusters found at the previous time step. This is the approach taken in the following algorithm; potential alternatives are mentioned in Section C.3.

C.2 Algorithm and Evaluation

Informed by the previous discussion (appendix C.1), we provide an extension of Algorithm 2 to the case where the true number of communities earnestly varies over time, e.g., due to communities merging or splitting.

Algorithm 9 Evolving Community Detection with Grassmann Geodesics — Variable k_c

Input: Graphs and associated snapshot times $\{G_i, t_i\}_{i=1}^{T}$
Input: Estimates $k_{\min}$ and $k_{\max}$ of the minimum and maximum number of communities present at any time step
Input: Static spectral method $\mathscr{A}$ admitting an MCM (i.e., $\mathscr{A}$ formulated as in Algorithm 1)
Input: Unsupervised network partition benefit function (e.g., modularity for simple networks)
Apply steps 1 and 2 of algorithm 2 to fit a geodesic, using $k_e := k_{\max}$ as the embedding dimension
Instantiate $\boldsymbol{H} \in \mathbb{R}^{(k_{\max}-k_{\min}+1)\times T}$
for k in $[k_{\min}, k_{\max}] \cap \mathbb{Z}$ **do**
    for $i \in [T]$ **do**
        Euclidean clustering: apply the Euclidean clustering heuristic of $\mathscr{A}$ (e.g., k-means) at snapshot i, storing the modularity Q achieved by the resulting partition as $H_{ki} \leftarrow Q$. Recommended: initialize with results from time $i-1$ if applicable.
    end for
    Filtering: Optionally, smoothen over the row $\boldsymbol{H}_{k,:}$ (our experiments convolve with a 1D Gaussian filter; median filtering is another straightforward option.)
end for
for $i \in [T]$ **do**
    Define $k_{c,i}$ to be the choice of k corresponding to the maximum element of the vector $\boldsymbol{H}_{:,i}$
end for
Output: Partitions $(Z_1, \ldots, Z_{k_{c,i}})_i$, $i \in [T]$, of each G_i into communities

Implementing Algorithm 9 requires a benefit function suitable for the network modality. We remark that, at minimum, an extension of modularity exists in the literature for all network modalities considered in this text (signed [116]; overlapping [117]; directed [118]; multiview [73]; cocommunity [119]; hierarchical [120]). We also remark that, unless $k_{\max}$ is quite large, it generally suffices to fix $k_{\min} = 2$ if no preferred choice is known. In practice, therefore, $k_{\max}$ is usually the only required parameter. All experiments in this paper use the default $k_{\min} = 2$.

Figure 12 shows representative results of Algorithm 9 on a synthetic benchmark generated according to the methodology in [121], wherein eight planted communities gradually merge into six over time. The extension of geodesic normalized spectral clustering to case of time-varying k_c via Algorithm 9 uniformly outperforms the benchmarks; indeed, it is the only method to recover the planted community structure at every snapshot.

Evaluation on the temporal college football network. Figure 13 shows the results of Algorithm 9 applied to a temporal network inspired by the the popular American college football network of [122], where an edge is placed between two football teams whenever a game is played between them over a ten-year span (2009-2019).[10] The ground truth community structure is provided by conference alignment, as in [122]. The community structure evolves as teams change their conference membership over time. The raw data is obtained via the API offered in [123], then segmented into one snapshot per month of each football season (September 2009, October 2009, November 2009, September 2010...) to obtain a time series of $T = 33$ networks containing $d = 118$ nodes each.

[10]The full dataset is available at https://github.com/jacobh140/century-of-college-football.

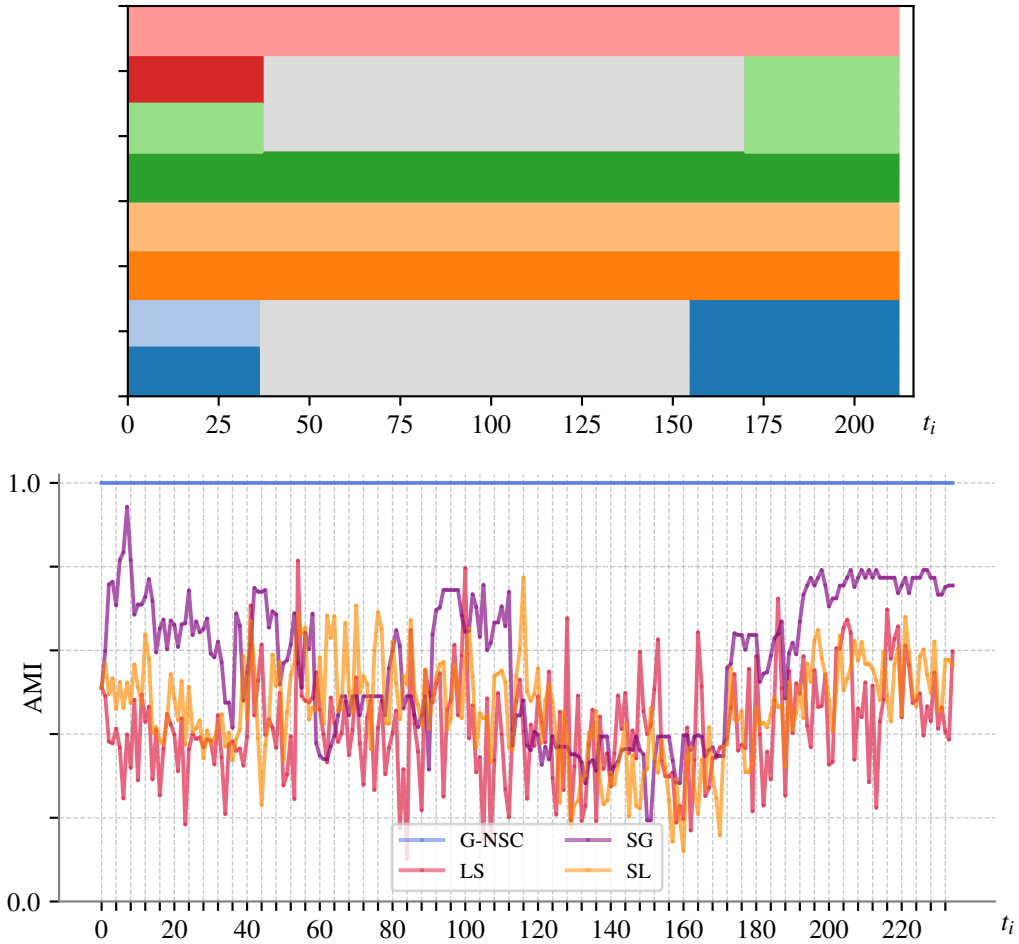


Figure 12. Evaluation of Algorithm 9 on an evolving community detection benchmark generated using the `tnetwork` library [77] via the methodology from [121]. Initially, 8 communities are present in the data. Over the course of $T = 211$ snapshots, two pairs of communities merge, such that ultimately six communities remain. The other four communities remain stable throughout. LEFT: A longitudinal plot of the planted community structure over time. The red community gradually merges into the light green community, and the light blue community gradually merges into the dark blue community. Note that the benchmark, by convention, does not treat nodes as belonging to any community at all when they are transitioning; this is indicated in gray. RIGHT: AMI comparison over time for geodesic normalized spectral clustering (G-NSC, extended to detect varying k_c via Algorithm 9 with $k_{\min} = 2$, $k_{\max} = 10$) versus the Label Smoothing (LS) approach of [78], the Smoothed Louvain (SL) algorithm of [44], and the Graph Smoothing (SG) approach of [79]. We remove from consideration the nodes currently lacking a true community membership when computing AMI at a given time step.

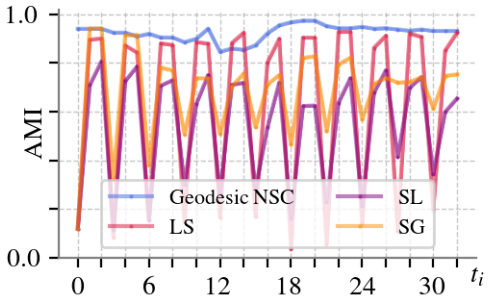


Figure 13. Evaluation of Algorithm 9 on a monthly extension of the college football network of [122]. Geodesic normalized spectral clustering (G-NSC, extended to detect varying k_c via Algorithm 9 with $k_{\min} = 2$, $k_{\max} = 10$) is compared against the Label Smoothing (LS) approach of [78], the Smoothed Louvain (SL) algorithm of [44], and the Graph Smoothing (SG) approach of [79], consistently outperforming the benchmarks over time.

A comparison of results can be seen in Figure 13. Geodesic normalized spectral clustering largely outperforms the benchmarks across time. We remark that the periodicity in some of the benchmarks' performance may be attributed to the fact that the density of in-conference games played is lower in September than in October and November.

C.3 Alternative Methods and Future Interest

The heuristic offered by Algorithm 9, though empirically effective and simple to implement, is nevertheless one of many options for extending Algorithm 2 to the setting where the number of communities changes over time. One alternative to the 'benefit function + sweep k_c' approach detailed previously would be to employ one of many methods for automated model selection in k-means if applicable, using e.g. the GAP statistic [124] or silhouette scores [125, 126] either as a benefit function for Algorithm 9 or standalone, or using X-means [127]. We also note that the geodesic generalization of hierarchical community detection described in Section A.6, put in conjunction with Algorithm 9 by using the latter to estimate the size of the initial partition, yields a dynamic algorithm for detecting hierarchical communities that is hyperparameter-free. It is of future interest to seek a variant of Algorithm 2 which circumvents the requirement of a temporally fixed embedding dimension k_e altogether.

D Remarks on Scaling

The worst case per-iteration time complexity of the geodesic-fitting step in Algorithm 2 and Algorithm 9 is $O(Td^2k_e)$, where k_e is the embedding dimension. The other potentially expensive step is the multiple applications of a Euclidean clustering heuristic. In the case of k-means, various techniques have been devised to increase efficiency [128–130], though none of our experiments make use of these.

In this section, we assess the scaling of Algorithm 9 with respect to d and T on the dynamic stochastic block model introduced in Section 3. We implement Algorithm 9 for normalized spectral clustering in Python on a 2019 MacBook Pro (2.3 GHz 8-Core Intel Core i9 Processor, 16GB RAM), using sparse matrices and the k-means++ implementation found in `scikit-learn`. The other algorithms are compared using their implementations in the `tnetwork` `Python` library. We find that, for the parameter settings, hardware, and implementations considered, the geodesic recovered or nearly recovered the true community structure, outperforming the benchmarks while running about an order of magnitude (10.47x on average) faster.

d	T	GNSC	LS	SL	SG
10^3	10^2	$\mathbf{1.0 \pm 0.0}$ **(36s)**	0.65 ± 0.32 (481s)	0.86 ± 0.02 (144s)	0.80 ± 0.03 (349s)
10^3	10^3	$\mathbf{1.0 \pm 0.0}$ **(387s)**	0.65 ± 0.33 (5893s)	0.87 ± 0.02 (2172s)	0.79 ± 0.01 (5445s)
10^4	10^2	$\mathbf{0.99 \pm 5 \times 10^{-5}}$ **(641s)**	N/A	0.71 ± 0.11 (7266s)	N/A

Table 2: Comparison of mean AMI and empirical runtime on the dynamic stochastic block model ($p_{\text{out}} = 200d^{-1}$, $p_{\text{in}} = 300d^{-1}$, $p_{\text{switch}} = (10T)^{-1}$, $k = 5$) as size and longitude vary. The methods compared are geodesic normalized spectral clustering (G-NSC, ours), label smoothing (LS, [78]), smoothed Louvain (SL, [44]), and smoothed graph (SG, [79]), as in Section 3. A 'N/A' designation is given wherever a method timed out during a simulation.