SOFT PARTITION MARKOV STABILITY FOR GRAPH CLUSTERING

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose soft partition Markov stability (SPMS), a novel, dynamics-based quality metric for graph clustering that offers greater interpretability than existing metrics. SPMS extends the ideas of traditional Markov stability to soft clusterings by describing the rate at which communities trap random walkers on the graph for a fixed time, while explicitly integrating information about cluster inhomogeneity. We argue that SPMS is a suitable objective function for graph clustering and analyze how the spectrum of the underlying graph influences its behavior. In our algorithmic framework, we describe a clustering with a set of localized probability distributions, which we call a soft partition, and maximize SPMS over the space of soft partitions to obtain a nuanced description of the community structure. The optimal soft partition yields information about intra-cluster degree inhomogeneity and naturally identifies ambiguous nodes, which may intuitively belong to multiple clusters or none at all. We describe an efficient method to optimize SPMS and demonstrate its empirical success on model graphs and large, real-world datasets. Finally, we propose and test a downstream algorithm for identifying core-periphery substructure within clusters.

For an undirected graph $G = (V, E)$, we consider the multi-way clustering problem of identifying k sufficiently disjoint communities, whose nodes are well-connected among themselves but sparsely connected with the rest of the network. For real-world data, however, the structure of a graph with communities usually cannot be fully described by a simple partition of the nodes. Many graphs exhibit degree inhomogeneity within their clusters or contain nodes whose cluster membership is ambiguous. A flexible clustering algorithm should capture these features, but the most commonly used techniques, including spectral methods and modularity optimization, return a hard partition of the nodes and therefore neglect them.

Spectral clustering refers to a class of network clustering algorithms that leverage invariant subspaces of graph-related matrices to find well-connected subgraphs. We direct the reader to Ding et al. (2024) for a survey of algorithms and their applications. Traditionally, spectral clustering involves computing a predetermined set of eigenvectors of the normalized Laplacian and using k-means (or similar) to cluster the spectral embeddings (von Luxburg, 2007). Other approaches for community detection rely on metrics that directly describe the quality of a clustering. For instance, modularity optimization seeks a partition of nodes that maximizes the density of edges within clusters, compared to what would be expected in a random graph with the same degree distribution (Newman and Girvan, 2004; Lambiotte and Schaub, 2022). A related, but more flexible, metric known as Markov stability describes how random walkers move between clusters and has recently seen success as a clustering objective function (Delvenne et al., 2010; Lambiotte and Schaub, 2022).

As a consequence of seeking partitions of the nodes, these clustering algorithms lack the flexibility to highlight the degree inhomogeneity and cluster membership ambiguity that is common in real-world data. One can imagine a graph that exhibits assortative community structure but also has ambiguous nodes, e.g., low-degree vertices that do not naturally belong in any cluster or vertices that sit in the boundary of multiple communities. One may also consider a graph with inhomogeneous clusters, i.e., clusters with a highly non-uniform degree distribution. An algorithm that outputs a classic hard partition of V lacks the flexibility to detect these ambiguous nodes since each vertex is in a single community. Furthermore, if cluster membership is binary, cluster inhomogeneity will not be captured.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Instead, we model a k-way clustering by a *soft partition*: a set of k sufficiently disjoint probability distributions on V, where cluster membership is assigned to nodes with significant mass in a distribution. Soft partitions are flexible enough to detect ambiguous nodes. For example, a node in the intersection of multiple communities receives mass in several distributions, and a node with no natural cluster assignment receives negligible mass everywhere. Moreover, non-uniform distributions capture degree inhomogeneity within each community, placing high mass on central vertices.

We then introduce *soft partition Markov stability* (SPMS), a novel quality metric for spectral clustering that utilizes this framework to robustly capture node ambiguity and cluster inhomogeneity. Our optimization procedure transforms the dominant eigenspace V_k of the normalized Laplacian matrix L_N to output soft partitions that capture the underlying cluster structure. We initialize our algorithm with an efficient pivoted QR-based clustering algorithm (CPQR) (Damle et al., 2019) and optimize SPMS with gradient-based methods. On stochastic block models (Holland et al., 1983), random geometric graphs, and real datasets, our method naturally finds cluster structure, while simultaneously detecting ambiguous nodes and capturing cluster inhomogeneity. Finally, we introduce SPMS-CP, a downstream algorithm that exploits the flexibility of SPMS optimization to find well-connected cores in graphs that exhibit core-periphery substructure (Tang et al., 2019).

1 Background

Consider a connected, undirected graph $G = (V, E)$ on n nodes with adjacency matrix $A \in \{0,1\}^{n\times n}$. Let $d = A\mathbb{1}_n$ be the degree vector, where $\mathbb{1}_n$ is the all ones vector, and let $D = \text{diag}(d)$. Throughout our discussion, we let $L_N = I - D^{-1/2}AD^{-1/2}$ be the normalized Laplacian and $V_k \in \mathbb{R}^{n\times k}$ be the matrix whose columns are the k algebraically smallest eigenvectors of L_N.

1.1 Markov Stability

Beyond edge-based metrics like normalized cut (von Luxburg, 2007), quality functions based on the dynamics of random processes have proven effective for clustering. One such metric is Markov stability, which measures the probability that an unbiased random walk remains in its starting cluster after t steps, minus the stationary rate. This idea generalizes modularity by introducing a resolution parameter t that is directly related to random walk dynamics. Formally, consider an undirected graph G that admits an ergodic unbiased random walk with transition matrix $T = D^{-1}A$. For example, taking G to be connected and non-bipartite ensures ergodicity of the underlying Markov chain and the existence of a unique stationary measure π. The Markov stability of a partition (definition 1.2) is then defined as the trace of the $k \times k$ stability matrix (definition 1.1) whose entries capture the flow of random walks between clusters as the difference between finite-time and stationary rates. We direct the reader to Lambiotte and Schaub (2022) for a thorough discussion of Markov stability.

Definition 1.1. Given a partition of the nodes $\mathcal{C} = \{C_1, \ldots, C_k\}$, let $H \in \{0,1\}^{n\times k}$ be the indicator matrix of $\mathcal{C}$ with $H_{ij} = 1$ if $i \in C_j$ and $H_{ij} = 0$ otherwise. Let $\Pi = \text{diag}(\pi)$. For any positive integer t, define the *Markov stability matrix* of $\mathcal{C}$ at time t to be

$$S(t, H) = H^T(\Pi T^t - \pi\pi^T)H. \tag{1}$$

The a, b entry of the $k \times k$ matrix S describes the probability that a random walk starting in C_a will be in C_b after t steps, minus the stationary rate of this occurring. Hence, the diagonal entries of $S(t, H)$ describe the rate at which random walkers end in their starting cluster after t steps, a quantity that will be large for a good clustering. This motivates the definition of Markov stability.

Definition 1.2. For a partition H and time step t, define the *Markov stability* of H for time t as

$$s(t, H) = \min_{0\le\tau\le t} \text{Tr}\,[S(\tau, H)]. \tag{2}$$

Remark 1. For fixed H, we often see that $\text{Tr}\,[S(t, H)]$ is monotonically decreasing in t on clusterable graphs. The minimum is taken in eq. (2) to make the metric sensible for bipartite and almost-bipartite graphs, for which a partition may appear stable for even t but unstable for odd t.

Markov stability has seen success as a quality metric for graph clustering algorithms and comes with interesting properties. Its dependence on the time step allows for tunable cluster resolution and it is easy to see that $s(1, H)$ is equal to the classical notion of modularity. We direct the reader to Delvenne et al. (2013) for a discussion of Markov stability and its connections to other methods.

1.2 Related Work

More interpretable approaches to graph clustering have been explored previously. Fuzzy graph clustering (FGC) (Peng et al., 2021) employs a similar idea to SPMS optimization and outputs a matrix with non-negative entries and unit row sums that captures the underlying cluster structure. This approach allows one to explicitly assign a clustering by taking the index of the largest element in each row of the resulting fuzzy indicator matrix. Nodes receiving significant weight in multiple indices may be interpreted as ambiguous and can be assigned to multiple clusters. While this method has seen success empirically, its use of row normalization, as opposed to column normalization, prevents it from distinguishing high-degree ambiguous nodes, which straddle several clusters, from low-degree ambiguous nodes, which belong to no community at all.

The idea of interpreting a clustering as a set of probability distributions was inspired by a recent master's thesis (Singh, 2023). This work involves defining a sensible notion of variance for distributions supported on graphs that incorporates the network topology. An optimal set of k probability distributions is found by minimizing the sum of their variances. This spectral approach defines the optimal distributions as a function of the eigenspace V_k, identically to SPMS optimization. This work inspired the optimization framework we discuss in section 3.1. The empirical success of clustering graphs in terms of low-variance distributions is demonstrated in the thesis. This compelling framework incited our exploration of SPMS as a canonical and flexible clustering objective function.

The focus of our discussion on core-periphery detection is to illustrate a downstream application of SPMS optimization that leverages our algorithm's flexibility. However, core-periphery detection algorithms have been well-studied outside of the context of clustering. For a thorough discussion of core-periphery structure, detection algorithms, and applications, we direct the reader to a survey paper (Tang et al., 2019). Two main perspectives on this computational problem have prevailed: binary classification (Brusco, 2011; Zhang et al., 2015) and quantization assignment (Yan and Luo, 2019). The former views core-periphery structure as a dichotomy, with every node being an element of either core or periphery. The latter perspective seeks to assign a coreness score to each node. Similarly to SPMS-CP, Della Rossa et al. (2013) identify core-periphery structure greedily through the lens of random walk dynamics. However, this method grows the periphery by one node at each step, making it impractical for large graphs.

2 Soft Partition Markov Stability

In this section, we define our proposed quality metric *soft partition Markov stability* (SPMS). We formulate an explicit random process on the graph that gives rise to SPMS. We explore the theoretical properties of this metric and show how its convergence rate, with respect to the time parameter, is inversely related to the spectral gap of the transition matrix. Finally, we motivate the use of SPMS as a clustering objective function and illustrate the significant flexibility advantages gained by describing community structure with soft partitions.

2.1 Definition of SPMS

Consider an undirected, connected, and non-bipartite graph $G = ([n], E)$, where $[n] = \{1, \dots, n\}$, with transition matrix T and unique stationary distribution π. For a soft partition P (definition 2.1), we define its SPMS (definition 2.3) to be the trace of a $k \times k$ stability matrix (definition 2.2), whose entries describe the flow of random walks between clusters as a difference of the finite-time rate and the stationary rate, weighted by the mass of each node in P.

Definition 2.1. For a k-way cluster assignment, we encode a *soft partition* of V as a matrix $P \in [0, 1]^{n \times k}$ satisfying $\mathbb{1}_n^T P = \mathbb{1}_k^T$. The distribution in column a is denoted by $P_{:,a}$.

Definition 2.2. Given a soft partition P and positive integer t, we define the *soft partition Markov stability matrix* (or just stability matrix) of P for time t as

$$R(t, P) = P^T (T^t - \pi^T \otimes \mathbb{1}_n) P \tag{3}$$

where $\otimes$ is the Kronecker product.

Definition 2.3. Define the *soft partition Markov stability* (SPMS) for the soft partition P and time step t as

$$r(t, P) = \min_{0 \le s \le t} \operatorname{Tr}[R(s, P)]. \tag{4}$$

As before, $R(t, P) \in \mathbb{R}^{k\times k}$ and each entry describes the difference of two probabilities – an intuitive discussion of the quantities is given in appendix B.2. Analytically, we see that

$$[R(t,P)]_{ab} = \sum_{i,j=1}^{n} P_{ia}P_{jb}(T^t_{ij} - \pi_j) \tag{5}$$

where $T^t_{ij} := [T^t]_{ij}$. We see that the a, b entry of $R(t, P)$ is related to the flow of finite-time random walkers that start in cluster a and end in cluster b. That is, each summand describes the probability that a random walker will go from node i to node j in t steps minus the stationary rate, weighted by the mass of nodes i and j in their respective distributions. In particular, we see that each entry of eq. (3) describes the flow of a random process between clusters as outlined in appendix B.2.

2.2 Convergence Rates for SPMS

Similarly to how Cheeger inequalities relate the second smallest eigenvalue of L_N to conductance (Mohar, 1989), we establish a connection between SPMS and the spectral gap of the transition matrix T. We use the theory of mixing times to show that the time for which the SPMS of a clustering can be large is inversely related to the spectral gap of T. All proofs are given in appendix B.3.

Suppose $\lambda_1, \ldots, \lambda_n$ are the eigenvalues of L_N, listed in increasing order, for a connected, non-bipartite graph. Recall that $\lambda_i \in [0, 2]$ for all i, $\lambda_1 = 0$, and $\lambda_i > 0$ for $i > 1$. Notice that $T = D^{-1}A$ is similar to $D^{-1/2}AD^{-1/2}$ and deduce that the eigenvalues of T are $1 - \lambda_i$ for $i \in [n]$. It follows that T has an eigenvalue of 1 and all others will be smaller in absolute value. We now give the definition of the (absolute) spectral gaps (definition 2.4) of T and L_N.

Definition 2.4. Consider an ergodic Markov chain with transition matrix T that has eigenvalues $\lambda_1, \ldots, \lambda_n$ with $\lambda_1 = 1$. The *spectral gap* of T is defined as $\lambda^*_T = 1 - \max_{i=2,\ldots,n} |\lambda_i|$. Let $\mu_1 \leq \cdots \leq \mu_n$ be the eigenvalues of L_N with $\mu_1 = 0$, then the *spectral gap* of L_N is $\lambda^*_{L_N} = \mu_2$.

Remark 2. As suggested by the Cheeger inequality (Mohar, 1989), a classical heuristic in the spectral clustering literature is that clusterable graphs exhibit a small $\lambda^*_{L_N}$. Since we are working with random walks, we wish to extend this idea to the spectrum of the transition matrix $T = D^{-1}A$. If the eigenvalue that maximizes $|\lambda_i|$ is positive, then $\lambda^*_T = \lambda^*_{L_N}$. It is easy to see that $0 \leq \lambda^*_T \leq \lambda^*_{L_N}$ always. Thus, if the spectral gap of L_N is small, then the spectral gap of T will be small. As desired, we may now restate the Cheeger inequality in terms of the transition matrix as $2h(G) \geq \lambda^*_T$, where $h(G)$ is the Cheeger isoperimetric constant. From now on, we denote λ^*_T by λ^* for brevity.

We now state our main results that relate the behavior of $r(t, P)$ to the spectral gap of the underlying Markov chain. While it is clear that $\lim_{t\to\infty} \operatorname{Tr}[R(t, P)] = 0$, theorem 2.1 shows how this convergence rate depends on λ^*.

Theorem 2.1. *Consider a connected, non-bipartite graph G that admits an unbiased random walk with transition matrix T. For any $\epsilon > 0$ and soft partition $P \in [0, 1]^{n\times k}$, there exists a time $t(\epsilon)$ such that for all $t \geq t(\epsilon)$ one has $|\operatorname{Tr}[R(t, P)]| < \epsilon$, and consequently, $r(t, P) < \epsilon$. Specifically, letting $\pi_{min} = \min_{i\in[n]} \pi_i$, we have*

$$t(\epsilon) = \frac{1}{\lambda^*} \log \frac{2kn}{\epsilon \pi_{min}}. \tag{6}$$

Remark 3. From the formula for $t(\epsilon)$, we can see that a smaller λ^* yields a larger $t(\epsilon)$. Intuitively, a good soft partition on a clusterable graph should remain stable for a long time, i.e., there should be some P for which $|\operatorname{Tr}[R(t, P)]|$ is large for long time steps. The result shows that the maximum time for which $|\operatorname{Tr}[R(t, P)]|$ can be larger than ϵ is inversely related to the spectral gap. Although $\lim_{t\to\infty} |\operatorname{Tr}[R(t, P)]| = 0$ is guaranteed, our result shows that $r(t, P)$ converges more slowly on graphs with a small spectral gap, which is evidence of a good clustering.

2.3 SPMS for Clustering

SPMS is the quantity that we will optimize to find a clustering. Hence, we should have that a good clustering will make the diagonal entries of $R(t, P)$ large with small off-diagonal entries. To accomplish this, intuitively, one should assign high mass to nodes that are central to a cluster and low mass to peripheral nodes.

When $a = b$, eq. (5) can be made large by placing high mass on nodes where $T^t_{ij} - \pi_j$ is large, which are canonically the well-connected, central vertices of a cluster. Suppose $a \neq b$ and a random walk ends at some node j in the periphery of $P_{:,b}$. Although leaking into the periphery of another cluster is not uncommon, if P_{jb} is small to reflect its place in the cluster, then the i, j contribution to $[R(t, P)]_{ab}$ will be small as desired. Thus, SPMS optimization will capture any non-uniformity in the cluster structure. Defining our metric in terms of soft partitions makes our approach strictly more flexible than traditional Markov stability, constituting a significant advantage of our method.

3 Algorithm

To find a clustering, we seek to optimize the quantity $r(t, P)$ over matrices $P \in [0, 1]^{n\times k}$ satisfying $\mathbb{1}_n^T P = \mathbb{1}_k^T$. We present a practically efficient scheme of parameterizing such matrices P as a function of the eigenspace V_k and an orthogonal matrix in a way that allows for gradient-based optimization. Moreover, we argue that the structure of the eigenspace yields distributions that are distinct and localized to communities.

3.1 Optimization Formulation

To find a k-way clustering of the graph, we maximize SPMS eq. (4) over a set of soft partitions. Analogously to remark 1, we see empirically that $\mathrm{Tr}[R(t, P)]$ is monotonically decreasing in t and that $r(t, P) \approx \mathrm{Tr}[R(t, P)]$. Thus, we take $\mathrm{Tr}[R(t, P)]$ as a proxy for $r(t, P)$ and choose it as the objective function for our optimization. For a fixed time t, we solve

$$\begin{aligned} \underset{Q\in \mathrm{SO}(k)}{\text{maximize}} \quad & \mathrm{Tr}[R(t, P)] \\ \text{subject to} \quad & P = [V_k Q]^{\odot 2} \end{aligned} \tag{7}$$

where $\odot$ is the Hadamard elementwise product and $M^{\odot 2} := M \odot M$. Since the columns of V_k are orthonormal eigenvectors corresponding to the k algebraically smallest eigenvalues of L_N, the columns of $V_k Q$ are also orthonormal. Thus, we take P to be the elementwise square of $V_k Q$, so the columns of the resulting matrix are probability distributions. Although this definition of P may seem restrictive, it is well-known that V_k contains rich structural information about the graph.

We leverage a result from Lie theory to reduce optimization over $SO(k)$ to unconstrained optimization in Euclidean space. The specific implementation is given in appendix C.1. In our experiments, we observe that the clusters returned by eq. (7) are distinct, although this is not explicitly enforced above. In appendix C.1, we detail a heuristic argument for this behavior and provide a regularized variant of eq. (7) that directly penalizes cluster similarity.

3.2 Initialization

We must choose some $Q_0 \in SO(k)$ to define an initial guess for P. Given that $r(t, P)$ is non-convex, choosing a good initialization is beneficial. We take Q_0 to be the orthogonal matrix from the polar factorization in CPQR-based spectral clustering (Damle et al., 2019). By swapping columns, we may assume without loss of generality that such $Q_0 \in SO(k)$. In appendix E.7, we empirically evaluate the CPQR initialization against more naive initial guesses, showing that it greatly facilitates convergence but is not strictly necessary for the functioning of SPMS optimization in some cases.

4 Core-Periphery Substructure

Adjacent to the study of clustering, the detection of core-periphery structure in graphs has been of interest in fields ranging from economics to social network analysis (Tang et al., 2019). This notion refers to graphs that exhibit a bipartition of the nodes into a well-connected, cohesive core and a sparsely connected periphery. In Zhang et al. (2015), the authors formalize core-periphery structure as a special case of the SBM (appendix D.1). While there has been significant work on algorithms to detect and identify core-periphery structure in graphs (Borgatti and Everett, 2000; Brusco, 2011; Yan and Luo, 2019; Zhang et al., 2015), little has been done in the context of clustering. We consider clusterable graphs whose communities exhibit core-periphery structure and propose an algorithm

to find well-connected cores within each cluster. Building off distributions returned from SPMS optimization, we define a novel dynamics-based quality metric to maximize and explore greedy methods for choosing a core. Numerical experiments highlight its efficacy. Notably, our method leverages flexibility in how we interpret soft partitions generated by optimizing SPMS.

4.1 Core Metric

Our core metric evaluates the quality of a proposed core in terms of random walk dynamics. This yields an intuitive and direct approach for analyzing the core-periphery substructure of clusters, constituting a major advantage over existing methods. We now motivate the core metric and discuss its properties. First, we define a function describing the flow of random walks between subsets of V (definition 4.1).

Definition 4.1. For a connected, non-bipartite graph $G = (V, E)$, positive integer t, and probability distribution p on V, we define the function $f_t^p : \mathcal{P}(V) \times \mathcal{P}(V) \to [0, 1]$ by

$$f_t^p(A, B) = \sum_{i \in A} \sum_{j \in B} p_i p_j T_{ij}^t. \tag{8}$$

Considering a distribution on the graph, $f_t^p(A, B)$ describes the flow of random walks from A to B, with each node weighted by p. With the SBM characterization of core-periphery structure given in appendix D.1, this flow is large within the core and smaller between the periphery and core. That is, denoting $C \subset V$ and $P \subset V$ to be the core and periphery, respectively, we should have $f_t^p(C, C) > f_t^p(P, C)$ for a suitable distribution p. In practice, for a cluster we wish to analyze, p will be the corresponding column of P^* from SPMS optimization. Now, we can define the core metric (definition 4.2), the objective function that we optimize to find a core within a given cluster.

Definition 4.2. Let $C \subset V$ be a non-empty set which we will call the core and let its complement $P = C'$ be the periphery. We define the core metric as

$$\varphi_t^p(C) = \frac{f_t^p(C, C)}{|C| + f_t^p(P, C)}. \tag{9}$$

We iteratively grow the core and seek a maximum of $\varphi_t^p(C)$, which favors cores that retain random walks originating within them. Since SPMS optimization concentrates mass on core nodes, a well-chosen core yields large $f_t^p(C, C)$ and small $f_t^p(P, C)$ – the latter reflecting low periphery-to-core transition probability and low peripheral mass. The metric also naturally penalizes extreme core sizes. An oversized core is punished by the cardinality term in the denominator, as adding non-core nodes produces diminishing returns in $f_t^p(C, C)$ while $|C|$ constantly grows. An undersized core exhibits small $f_t^p(C, C)$ and consequently, small $\varphi_t^p(C)$. Hence, we expect $\varphi_t^p(C)$ to exhibit a local maximum at an optimal core, behavior we confirm on several model graphs. We now provide an intuitive sufficient condition for increasing the core metric at an iteration (proven in appendix D.2).

Theorem 4.1. *Consider a graph $G = (V, E)$, a distribution p on V, and a positive integer t. Let $C \subset V$ be a non-empty core and let $P = C'$ be the periphery. Suppose for a node $k \notin C$ that*

$$\frac{f_t^p(C, \{k\})}{f_t^p(C, C)} \geq \frac{f_t^p(P, \{k\})}{|C|}. \tag{10}$$

Then, $\varphi_t^p(C \cup \{k\}) \geq \varphi_t^p(C)$ if k satisfies,

$$\frac{f_t^p(\{k\}, C \cup \{k\})}{\sum_{n \in C} f_t^p(\{n\}, C)} \geq \frac{1}{|C|}. \tag{11}$$

We have derived an intuitive sufficient (but not necessary) condition for increasing the core metric at each iteration: the metric is guaranteed to grow if the contribution of the added node is "better than average" in the sense of eq. (11). While theorem 4.1 characterizes when the metric improves, we have yet to specify how each additional node is chosen.

4.2 Choosing a Core

Throughout our testing, we used several greedy heuristics to iteratively grow the core and seek a maximum of φ_t^p, some enforcing core contiguity and others prioritizing efficiency. In this section,

we outline our general framework and defer specific implementation details and comparative discussion of each method to appendix D.3.

Given a graph for which we wish to identify core-periphery substructure, we first run SPMS optimization with time step t to obtain a soft partition that captures the degree inhomogeneity of each cluster. For a cluster we wish to analyze, we set the distribution p to be the corresponding column of P^*. In practice, it would be sensible to choose some lower limit ℓ and upper limit $m < n$ on the size of an acceptable core (which may vary based on the application) and iterate the chosen process to get cores $C_1, \ldots, C_m$. The optimal core will be taken as $C^* = \arg\max_{C_i, \ell \leq i \leq m} \varphi_t^p(C_i)$.

5 Numerical Results

In this section, we explore the performance of our algorithms on model graphs and real datasets. All our experiments were in Python using the NetworkX package (Hagberg et al., 2008) and the Adam optimizer (Kingma and Ba, 2014) in TensorFlow. When running gradient descent, we say that SPMS has converged if the relative change between iterations is less than a tolerance of 10^{-5}. We will first find soft clusterings of random geometric graphs arising from Gaussian mixture distributions and analyze their core-periphery substructure. Later, we demonstrate explicit flexibility advantages SPMS optimization has over baseline soft clustering algorithms. We also analyze our methods on large, real-world road network graphs to demonstrate the scalability of SPMS optimization and SPMS-CP. Empirically, we observed that the performance of our algorithms is insensitive to the choice of the time step for small t and fixed k. We take $t = 5$ for all tests and refer the reader to appendix E.6 for a discussion of t as a tunable resolution parameter for multiscale clustering.

5.1 Gaussian Random Geometric Graphs

We show how our method captures cluster inhomogeneity by considering random geometric graphs (RGGs) arising from Gaussian mixture distributions. We sample 100000 points from a mixture of four bivariate Gaussians in $\mathbb{R}^2$ and construct a graph by connecting "nearby" points with an edge (see appendix E.2 for details). This method produces graphs with inhomogeneous clusters as samples from a Gaussian concentrate near the mean. For a sampled RGG, we run SPMS optimization with $k = 4$ to yield a solution P^*. In fig. 1, we visualize the optimal distributions by plotting the sampled points in the plane with their corresponding mass on the z-axis and fitting a surface to the data. We emphasize how the optimal distribution concentrates mass at the mean of the respective Gaussian and tapers off radially. Also, note that extreme points from other Gaussians fall in the periphery of the pictured cluster and receive low, but non-trivial, mass in its distribution, further highlighting the flexibility of SPMS optimization.

RGGs are also a useful model to analyze SPMS-CP. We sample random graphs with 2000 initial data points as before and run SPMS optimization with $k = 4$. We analyze the cluster corresponding to the distribution $p = P^* e_1$, and run the Contiguous procedure (appendix D.3) to grow the core with $m = 1000$, computing φ_5^p at each iteration. In fig. 1, we see the plot of the core metric for $C_1, \ldots, C_m$ on 50 random graphs sampled from the Gaussian mixture distribution. Observe that the metric reaches a clear maximum and consistently decreases after this point. Upon repeating this experiment, the optimal core size varied, with values ranging roughly from 250 to 450. This behavior is not unexpected, as RGGs do not have a canonical ground-truth core. We emphasize that the core nodes almost always fall within one standard deviation of the mean of the respective distribution, showing the consistency of our method despite this natural variation.

5.2 Flexibility Advantages Over Row-Normalized Soft Clustering Methods

We show that SPMS optimization distinguishes between high-degree and low-degree ambiguous nodes, while row-normalized baselines cannot. Consider a 4-block SBM $G = (V, E)$ with $p = 0.7$, $q = 0.1$, and 100 nodes per cluster. We add two blocks of 50 ambiguous nodes, which we call *strong* and *weak*. Each strong (respectively, weak) node forms an edge with nodes in V with probability p_s (respectively, p_w), where $p_s \geq p_w$. For p_s large, strong nodes are well-connected with all clusters and could be considered members of each. For p_w small, weak nodes connect loosely to G and are not part of any cluster. Methods that return a distribution of cluster membership for each node, i.e., row-normalization, cannot always distinguish these two types of ambiguity. Indeed, the optimal

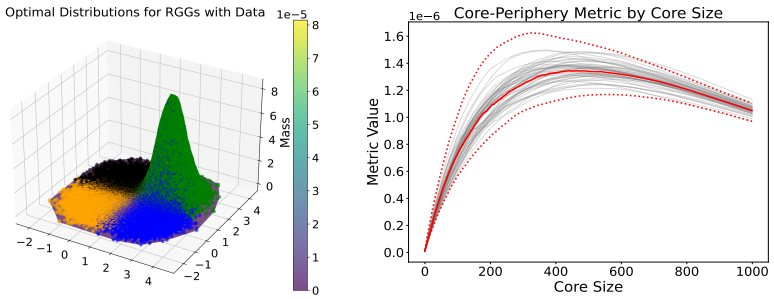


Figure 1: Visualization of the surface that was fit to the data (left). The data is colored to indicate the distribution Z_i from which it was sampled. The value of the φ_5^p for $C_1, \ldots, C_{1000}$ on 50 independent RGGs (right). A representative trial is in solid red and red dotted lines highlight extreme cases.

distributions will be roughly uniform in each block or will arbitrarily assign the planted blocks to a cluster. Column-normalized methods such as SPMS can separate these cases: strong nodes receive substantial mass in every distribution, while weak nodes receive negligible mass in all of them.

To test this claim, we fixed $p_w = 0.025$ and, for each $p_s \in \{0.025, 0.05, \ldots, 0.5\}$, sampled 200 graphs as above. For each graph and $k = 4$, we computed soft clusterings with SPMS optimization, the probabilistic method of Newman and Leicht (Newman and Leicht, 2007), FGC (Peng et al., 2021), and fuzzy k-means (Dunn, 1973) applied to the rows of V_k. For each method and each value of p_s, we trained a logistic regression model on 150 of the graphs to classify each ambiguous node as belonging to either the strong or weak block, using the corresponding rows of the soft clustering matrix as features.[1] We evaluated each classifier on the remaining 50 graphs and plotted the average AUC in fig. 2. When $p_s = p_w$, the planted blocks are indistinguishable, and each method scores around 0.5. As p_s grows, SPMS separates the planted ambiguous blocks as shown in fig. 2, with AUC approaching 1, while FGC and fuzzy k-means fail at most parameter regimes. Since the method of Newman and Leicht identifies groups of nodes that have similar connectivity patterns, it detects more than just community structure and is able to distinguish the planted blocks.

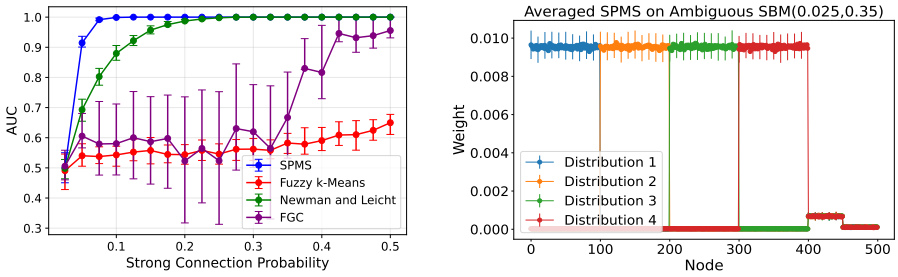


Figure 2: Plot of average AUC for each method across parameter regimes with error bars representing the 0.25 and 0.75 quantiles (left). Optimal distributions produced by SPMS optimization on model graphs with $(p_w, p_s) = (0.025, 0.35)$ averaged node-wise with standard deviation error bars reported every 10 nodes (right). Nodes $0 - 99, \ldots, 300 - 399$ represent communities, nodes $400 - 449$ represent the strong ambiguous block, and nodes $450 - 499$ represent the weak ambiguous block. Notice how SPMS strongly captures each community and separates the ambiguous blocks.

In appendix E.3, we visualize the optimal distributions produced by SPMS over several parameter regimes and show how the strong and weak planted blocks are naturally separated. We also analyze the distributions assigned to the strong and weak nodes by FGC and fuzzy k-means.

5.3 Real-World Data

We have demonstrated the flexibility and performance of SPMS optimization and SPMS-CP on model graphs; we now analyze our algorithms on a large road network encompassing San Francisco,

[1] Cluster labeling is inconsistent across trials, so we sorted each row in decreasing order before training.

San Jose, and Oakland. We obtained the data using Boeing (2025) and OpenStreetMap. We describe our cleaning procedure and some pathological properties of the data in appendix E.4.

We ran SPMS optimization with $k = 3$ to obtain an optimal soft partition P^*. The distribution capturing San Jose and a visualization of the induced hard partition are pictured in fig. 3.

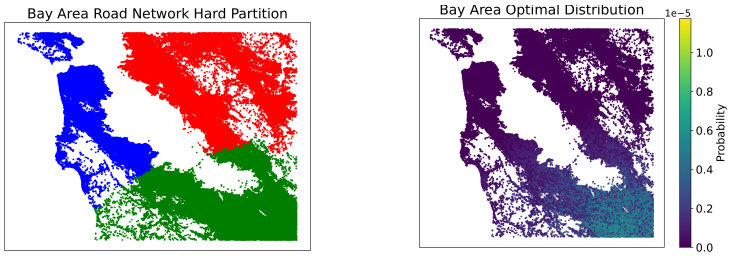


Figure 3: Hard partition from P^* (left). Distribution capturing San Jose (right). The hard partition naturally yields a cluster for each city. The optimal distribution placed high mass on the center of San Jose and tapered off at the periphery. Note the pathological edge effects discussed in appendix E.4.

Next, we analyze the core-periphery structure of the Bay Area road network with distributions coming from SPMS optimization. For efficiency, we suggest incorporating an increment parameter η and growing the core by adding η nodes at each iteration. Empirically, we have seen that $\varphi_t^p(C)$ is regular as a function of the core. So, if η is small, computing the core metric at these increments will give a good description of the graph of φ_t^p.

To avoid the edge effects discussed in appendix E.4, we choose $k = 12$ and analyze internal clusters. We choose $t = 5$ and $\eta = 25$, and look for cores of size at least $\ell = 1000$. We set the maximum core size to be $m = 75000$. On two different clusters, we iterate our algorithm using the Rapid method (appendix D.3) to obtain cores with sizes equal to each multiple of η less than m. In fig. 4, we see a visualization of the hard partition produced by SPMS optimization, as well as the optimal cores inside the clusters we analyzed. Observe how the core is naturally localized to the center of each cluster and becomes more diffuse toward the periphery.

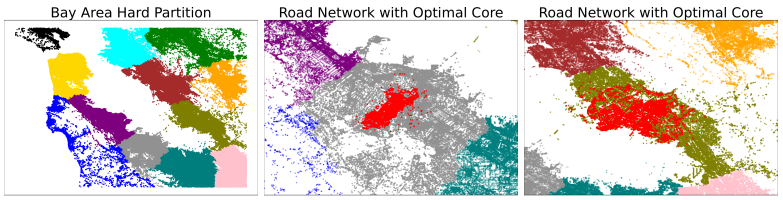


Figure 4: Visualization of the hard partition for $k = 12$ (left). Optimal cores in internal clusters (center; right). Core nodes are highlighted in red.

6 Discussion and Conclusion

We have presented a novel spectral algorithm for clustering an undirected graph with assortative community structure. Our approach utilizes a generalization of the standard Markov stability framework to output a set of localized, sufficiently disjoint probability distributions on V that recover the underlying clusters. This relaxed notion of clustering allows us to capture degree inhomogeneity and node ambiguity, structural properties that are common in real data and overlooked by standard community detection algorithms. We framed SPMS in the context of Markov chain dynamics and showed that it behaves consistently with classical intuition. We formulated our approach as an optimization problem (optionally with regularization) and showed an efficient initialization at a suitable clustering using CPQR factorizations. We demonstrated the performance and flexibility of our method on synthetic and real datasets. Moreover, we explored core-periphery substructure detection as a downstream application of SPMS optimization. We proposed a novel core metric and stated a result that justifies our modeling choices. Several methods for choosing an optimal core were discussed, and we illustrated the promising behavior of our approach on several datasets.

AI USE STATEMENT

In this work, we used generative AI tools to polish, tighten, and proofread writing in some sections. Specifically, we refined writing in the introduction, section 4, and section 5. We used AI tools to partially implement and refine experiments contained in section 5.2. AI assistance was used to partially develop the experiment in appendix E.6. AI tools were used to implement baseline algorithms. We have not used AI tools to develop theoretical models and conceptual frameworks, formulate mathematical claims, assist theorem proving and writing, clean data, interpret results, or to propose and refine hypotheses. The use of AI for translation or supporting qualitative and thematic data analysis is not applicable for this work. We have reviewed and carefully validated all AI-produced code. We take responsibility for the final content of this work, including text and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

We confirm that the research presented in this manuscript is entirely compliant with the ICLR Code of Ethics.

REPRODUCIBILITY STATEMENT

We have taken steps to ensure all claims from this paper are reproducible. Indeed, all theoretical claims and results from the main text are detailed in the appendix. We include references to all cited results and full proofs of all novel claims. Additionally, in the supplementary material, we have submitted the necessary data and code to reproduce all main figures and experiments in our manuscript. All data cleaning steps used for real data are outlined in appendix E.4.

REFERENCES

Pranjal Awasthi, Moses Charikar, Ravishankar Krishnaswamy, and Ali Kemal Sinop. The Hardness of Approximation of Euclidean k-Means. In Lars Arge and János Pach, editors, *31st International Symposium on Computational Geometry (SoCG 2015)*, volume 34 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 754–767, Dagstuhl, Germany, 2015. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISBN 978-3-939897-83-5. doi: 10.4230/LIPIcs.SOCG.2015.754. URL https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.SOCG.2015.754.

James C Bezdek. *Pattern recognition with fuzzy objective function algorithms*. Springer Science & Business Media, 2013.

James C. Bezdek†. Cluster validity with fuzzy sets. *Journal of Cybernetics*, 3(3):58–73, 1973. doi: 10.1080/01969727308546047. URL https://doi.org/10.1080/01969727308546047.

Geoff Boeing. Modeling and analyzing urban networks and amenities with osmnx. *Geographical Analysis*, 57(4):567–577, 2025. doi: https://doi.org/10.1111/gean.70009. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/gean.70009.

Stephen P Borgatti and Martin G Everett. Models of core/periphery structures. *Social Networks*, 21(4):375–395, 2000. ISSN 0378-8733. doi: https://doi.org/10.1016/S0378-8733(99)00019-2. URL https://www.sciencedirect.com/science/article/pii/S0378873399000192.

Michael Brusco. An exact algorithm for a core/periphery bipartitioning problem. *Social Networks*, 33(1):12–19, 2011. ISSN 0378-8733. doi: https://doi.org/10.1016/j.socnet.2010.08.002. URL https://www.sciencedirect.com/science/article/pii/S0378873310000390.

Mihai Cucuringu, Puck Rombach, Sang Hoon Lee, and Mason A. Porter. Detection of core–periphery structure in networks using spectral methods and geodesic paths. *European Journal of Applied Mathematics*, 27(6):846–887, 2016. doi: 10.1017/S095679251600022X.

Anil Damle, Victor Minden, and Lexing Ying. Simple, direct and efficient multi-way spectral clustering. *Information and Inference: A Journal of the IMA*, 8(1):181–203, 03 2019. ISSN 2049-8772. doi: 10.1093/imaiai/iay008. URL https://doi.org/10.1093/imaiai/iay008.

Fabio Della Rossa, Fabio Dercole, and Carlo Piccardi. Profiling core-periphery network structure by random walkers. *Scientific reports*, 3:1467, 03 2013. doi: 10.1038/srep01467.

J.-C. Delvenne, S. N. Yaliraki, and M. Barahona. Stability of graph communities across time scales. *Proceedings of the National Academy of Sciences*, 107(29):12755–12760, 2010. doi: 10.1073/pnas.0903215107. URL https://www.pnas.org/doi/abs/10.1073/pnas.0903215107.

Jean-Charles Delvenne, Michael T. Schaub, Sophia N. Yaliraki, and Mauricio Barahona. *The Stability of a Graph Partition: A Dynamics-Based Framework for Community Detection*, pages 221–242. Springer New York, New York, NY, 2013. ISBN 978-1-4614-6729-8. doi: 10.1007/978-1-4614-6729-8_11. URL https://doi.org/10.1007/978-1-4614-6729-8_11.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977. doi: https://doi.org/10.1111/j.2517-6161.1977.tb01600.x. URL https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.2517-6161.1977.tb01600.x.

Hongbo Deng and Jiawei Han. *Probabilistic Models for Clustering*, pages 61–86. 09 2018. ISBN 9781315373515. doi: 10.1201/9781315373515-3.

Ling Ding, Chao Li, Di Jin, and Shifei Ding. Survey of spectral clustering based on graph theory. *Pattern Recognition*, 151:110366, 2024. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2024.110366. URL https://www.sciencedirect.com/science/article/pii/S0031320324001171.

J. C. Dunn. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3(3):32–57, 1973. doi: 10.1080/01969727308546046. URL https://doi.org/10.1080/01969727308546046.

Maria Brigida Ferraro. Fuzzy k-means: history and applications. *Econometrics and Statistics*, 30:110–123, 2024. ISSN 2452-3062. doi: https://doi.org/10.1016/j.ecosta.2021.11.008. URL https://www.sciencedirect.com/science/article/pii/S2452306221001398.

Santo Fortunato and Marc Barthélemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104(1):36–41, 2007. doi: 10.1073/pnas.0605965104. URL https://www.pnas.org/doi/abs/10.1073/pnas.0605965104.

Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using networkx. In Gaël Varoquaux, Travis Vaught, and Jarrod Millman, editors, *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, Pasadena, CA USA, 2008.

Brian Hall. *Lie Groups, Lie Algebras, and Representations: An Elementary Introduction*. Graduate Texts in Mathematics. Springer International Publishing, 2015. ISBN 9783319134673. URL https://books.google.com/books?id=didACQAAQBAJ.

Paul W. Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 5(2):109–137, 1983. ISSN 0378-8733. doi: https://doi.org/10.1016/0378-8733(83)90021-7. URL https://www.sciencedirect.com/science/article/pii/0378873383900217.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Renaud Lambiotte and Michael T. Schaub. *Modularity and Dynamics on Complex Networks*. Elements in the Structure and Dynamics of Complex Networks. Cambridge University Press, 2022.

David Levin and Yuval Peres. *Markov Chains and Mixing Times*. American Mathematical Society, 2 edition, 2017. ISBN 1470429624.

Zijing Liu and Mauricio Barahona. Geometric multiscale community detection: Markov stability and vector partitioning. *Journal of Complex Networks*, 6(2):157–172, 04 2018. ISSN 2051-1329. doi: 10.1093/comnet/cnx028. URL https://doi.org/10.1093/comnet/cnx028.

Zijing Liu and Mauricio Barahona. Graph-based data clustering via multiscale community detection. *Applied Network Science*, 5, 01 2020. doi: 10.1007/s41109-019-0248-7.

Bojan Mohar. Isoperimetric numbers of graphs. *Journal of Combinatorial Theory, Series B*, 47(3):274–291, 1989. ISSN 0095-8956. doi: https://doi.org/10.1016/0095-8956(89)90029-4. URL https://www.sciencedirect.com/science/article/pii/0095895689900294.

M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E*, 69:026113, Feb 2004. doi: 10.1103/PhysRevE.69.026113. URL https://link.aps.org/doi/10.1103/PhysRevE.69.026113.

M. E. J. Newman and E. A. Leicht. Mixture models and exploratory analysis in networks. *Proceedings of the National Academy of Sciences*, 104(23):9564–9569, 2007. doi: 10.1073/pnas.0610537104. URL https://www.pnas.org/doi/abs/10.1073/pnas.0610537104.

OpenStreetMap. Openstreetmap. https://www.openstreetmap.org/.

Yong Peng, Xin Zhu, Feiping Nie, Wanzeng Kong, and Yuan Ge. Fuzzy graph clustering. *Information Sciences*, 571:38–49, 2021. ISSN 0020-0255. doi: https://doi.org/10.1016/j.ins.2021.04.058. URL https://www.sciencedirect.com/science/article/pii/S002002552100387X.

Michael T. Schaub, Jean-Charles Delvenne, Sophia N. Yaliraki, and Mauricio Barahona. Markov dynamics as a zooming lens for multiscale community detection: Non clique-like communities and the field-of-view limit. *PLOS ONE*, 7(2):1–11, 02 2012a. doi: 10.1371/journal.pone.0032210. URL https://doi.org/10.1371/journal.pone.0032210.

Michael T. Schaub, Renaud Lambiotte, and Mauricio Barahona. Encoding dynamics for multiscale community detection: Markov time sweeping for the map equation. *Phys. Rev. E*, 86:026112, Aug 2012b. doi: 10.1103/PhysRevE.86.026112. URL https://link.aps.org/doi/10.1103/PhysRevE.86.026112.

A. Singh. Algorithmic considerations for spectral clustering, and learning graphs from labeled data. Master's thesis, Cornell University, 2023.

Wenli Tang, Liutao Zhao, Wei Liu, Yiping Liu, and Bo Yan. Recent advance on detecting core-periphery structure: A survey. *CCF Transactions on Pervasive Computing and Interaction*, 1: 175–189, 11 2019. doi: 10.1007/s42486-019-00016-z.

U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, December 2007. doi: 10.1007/s11222-007-9033-z.

Bowen Yan and Jianxi Luo. Multicores-periphery structure in networks. *Network Science*, 7(1): 70–87, 2019.

Xiao Zhang, Travis Martin, and M. E. J. Newman. Identification of core-periphery structure in networks. *Phys. Rev. E*, 91:032803, Mar 2015. doi: 10.1103/PhysRevE.91.032803. URL https://link.aps.org/doi/10.1103/PhysRevE.91.032803.

A Background

Take a connected, undirected graph $G = (V, E)$ and consider the multi-way clustering problem of partitioning the vertices into k disjoint clusters. Suppose we know the corresponding symmetric adjacency matrix $A \in \{0,1\}^{n\times n}$. We denote the degree vector by $d = A\mathbb{1}_n$, where $\mathbb{1}_n$ is the n-dimensional all ones vector. We define the Laplacian to be $L = D - A$ and the normalized Laplacian to be $L_N = D^{-1/2}LD^{-1/2}$, where $D = \text{diag}(d)$, the diagonal degree matrix. Let $|V| = n$.

A.1 Cut Metrics

A suitable approach to finding a clustering of the graph is to minimize the normalized cut over possible partitions (von Luxburg, 2007). At a high level, the normalized cut of a partition of V will be low when few edges cross between clusters. Given a partition of the nodes $\mathcal{C} = \{C_1, \ldots, C_k\}$, define the normalized indicator vectors u_i by

$$[u_i]_j = \begin{cases} \left(\sum_{\alpha\in C_i} d_\alpha\right)^{-1/2} & , \quad \text{if } j \in C_i \\ 0 & , \quad \text{otherwise} \end{cases} \tag{12}$$

and the matrix $U = [u_1 \ldots u_k]$. Noting that $\text{Tr}\,[U^T LU]$ is the normalized cut of the partition $\mathcal{C}$ and that $U^T DU = I_k$, we could find a reasonable partition by solving the combinatorial optimization problem

$$\begin{aligned} \underset{C_1,\ldots,C_k}{\text{minimize}} \quad & \text{Tr}[U^T LU] \\ \text{subject to} \quad & U^T DU = I_k. \end{aligned} \tag{13}$$

However, this optimization problem is provably hard, so it is common to consider a relaxation of eq. (13). Letting $X = D^{1/2}U$ and relaxing the indicator constraint to let X be real-valued, we get the more feasible problem

$$\begin{aligned} \underset{X\in\mathbb{R}^{n\times k}}{\text{minimize}} \quad & \text{Tr}[X^T L_N X] \\ \text{subject to} \quad & X^T X = I_k. \end{aligned} \tag{14}$$

It is well-known that the optimal solution to eq. (14) is given by $X^* = V_k Q$ where V_k is the matrix whose columns are the eigenvectors associated with the k algebraically smallest eigenvalues of L_N and $Q \in O(k)$. Traditional spectral clustering leverages this result by embedding the nodes into $\mathbb{R}^k$ by the rows of V_k and running k-means to assign cluster membership. While this method is simple, finding the optimal clustering of the spectral embeddings is not guaranteed due to the non-convexity of the k-means objective function. Specifically, Awasthi et al. (2015) show that there is an $\epsilon > 0$ such that approximating the optimal solution to Euclidean k-means with a factor better than $1 + \epsilon$ is NP-hard. Moreover, it is unclear that finding a partition that improves the k-means objective function will also be better in terms of normalized cut. While distance-based clustering algorithms are sufficient and important to mention, we sought a metric that directly incorporates the network topology.

A.2 Choosing k

For the tests in this manuscript, we chose a suitable value of k using known information about the graphs we were analyzing. That is, we chose k to match the number of blocks in an SBM, or to match the number of cities in a road network. In an application setting, however, one may not have an intuitive guess for k.

The common solution to this problem is the *eigengap heuristic*. Suppose $\lambda_1 \leq \cdots \leq \lambda_n$ are the eigenvalues of the normalized Laplacian. Based on the Davis–Kahan theorem from matrix perturbation theory, the eigengap heuristic posits that a suitable value of k is such that λ_{k+1} is large relative to $\lambda_1, \ldots, \lambda_k$. In other words, one should choose k such that the eigengap $\lambda_{k+1} - \lambda_k$ is large. This heuristic is further explained and empirically justified on model graphs in von Luxburg (2007). Since SPMS optimization already requires the computation of the eigenspace V_k, the eigengap heuristic is a natural choice for our method.

If one prefers an algorithmic method for choosing k, an alternative is to first cluster the graph with the Louvain method (Lambiotte and Schaub, 2022) and choose k based on the output. The Louvain method is a greedy procedure that seeks to optimize modularity. Notably, the Louvain method does not require a choice of k and, instead, infers the number of communities by considering modularity at different resolutions. While the Louvain method is efficient, it comes with some disadvantages. The quality metric behind the Louvain method, modularity, provably fails to detect communities smaller than an explicit relative size (Fortunato and Barthélemy, 2007). Moreover, it is known that Louvain commonly converges to local optima.

A.3 Related Work

Our method is a soft clustering generalization of traditional Markov stability (Delvenne et al., 2010). Markov stability offers tunable cluster resolution through its time-step parameter, which controls the dynamics of a random walker on the graph. Since its introduction, several studies have used and elaborated on this framework for multiscale community detection. For instance, Schaub et al. (2012b) and Schaub et al. (2012a) leverage Markov stability to find large, non-clique-like communities in networks that previous algorithms fail to detect. Liu and Barahona (2018) reformulate Markov stability as a vector partitioning problem and provide a geometric interpretation of the method. Moreover, Liu and Barahona (2020) represent data similarity as a graph and use Markov stability to find multiscale community structure in datasets.

Beyond FGC algorithms discussed in section 1.2, methods relying on statistical inference have also been proposed to analyze soft cluster structure. In Newman and Leicht (2007), a general probabilistic framework is proposed that can be used to detect a wide variety of structural properties in graphs, including assortative community structure. They utilize ideas of probabilistic clustering (Deng and Han, 2018) and the expectation-maximization (EM) algorithm (Dempster et al., 1977) to find the parameters of a generative model that best fit the observed connectivity patterns between the nodes of the graph. This method outputs a probability distribution for each node that specifies its likelihood of belonging to each of the k groups and therefore defines a suitable soft partition of the network. Similarly to how FGC utilizes row normalization, this approach lacks the flexibility to detect nodes that do not naturally belong to any cluster.

In the discussion of fuzzy clustering algorithms, it is important to mention Fuzzy k-means (FkM) (Dunn, 1973; Bezdek†, 1973; Bezdek, 2013). We direct the reader to Ferraro (2024) for a survey of FkM and its variants. At a high level, this algorithm iteratively minimizes an objective function based on the distances between data points in Euclidean space to a set of centroids. Each summand in the objective function is weighted by its membership degree, a value in the unit interval specifying each point's affinity to a cluster. Normalizing the columns of the returned $n \times k$ membership degree matrix U to have unit sum, one would obtain probability distributions on the data that would be compatible with our discussion of SPMS. Although running FkM on the spectral embeddings may yield a good soft clustering in terms of SPMS, we would like to emphasize the main difference between our approach and FkM. That is, the objective function of FkM is only related to points in space and their pairwise distances, while SPMS optimization is based on the dynamics of random walks on the graph. Directly incorporating the graph structure into our objective function constitutes a significant advantage of our approach.

The focus of our discussion on core-periphery detection is to illustrate a downstream application of SPMS optimization that leverages the flexibility of our approach. However, core-periphery detection algorithms have been well-studied outside of the context of clustering. For a thorough discussion of the history of core-periphery structure, detection algorithms, and applications, we direct the reader to a survey paper (Tang et al., 2019). At a high level, two main perspectives on this computational problem have prevailed: binary classification (Brusco, 2011; Zhang et al., 2015) and quantization assignment (Yan and Luo, 2019). The former views core-periphery structure as a dichotomy, with every node being an element of either core or periphery. The latter perspective seeks to assign a coreness score to each node.

From the binary classification perspective, branch-and-bound methods have been successful in exactly identifying core-periphery structure on small networks (Brusco, 2011). However, this method becomes computationally intractable as the size of the graph grows. Others have looked at the problem from the perspective of maximum likelihood estimation and sought to find parameters of a

generative model that best explain the data (Zhang et al., 2015). The EM algorithm (Dempster et al., 1977) is frequently used to solve this optimization problem due to its efficiency, enabling statistical inference on graphs with millions of nodes.

From the quantization perspective, recent work has studied multi-core-periphery detection (Yan and Luo, 2019). This model describes a network with several dense cores, each sparsely connected among themselves, and a shared sparsely connected periphery and can be seen as a generalization of traditional core-periphery structure. This approach involves computing a canonical core-periphery ratio that describes how closely a partition captures multi-core structure.

Similarly to SPMS-CP, Della Rossa et al. (2013) identify core-periphery structure greedily through the lens of random walk dynamics. However, this method grows the periphery by one node at each step, making it impractical for large graphs. Cucuringu et al. (2016) propose transport-based core-periphery detection algorithms. One such method scores each node by the number of geodesic paths passing through it, so that nodes with high "betweenness" are identified as core.

B Soft Partition Markov Stability

B.1 Definition of SPMS

We remind the reader of the important definitions and formulas needed for the rest of the discussion.

Definition B.1. For a k-way cluster assignment, we encode a soft partition of V as a matrix $P \in [0,1]^{n\times k}$ satisfying $\mathbb{1}_n^T P = \mathbb{1}_k^T$. The distribution in column a is denoted by $P_{:,a}$.

Definition B.2. Given a soft partition P and positive integer t, we define the *soft partition Markov stability matrix* (or just stability matrix) of P for time t as

$$R(t,P) = P^T(T^t - \pi^T \otimes \mathbb{1}_n)P \tag{15}$$

where $\otimes$ is the Kronecker product.

Definition B.3. Define the *soft partition Markov stability* (SPMS) for the clustering P and time step t as

$$r(t,P) = \min_{0\le s\le t} \mathrm{Tr}[R(s,P)]. \tag{16}$$

As before, $R(t,P) \in \mathbb{R}^{k\times k}$ and each entry describes the difference of two probabilities—an intuitive discussion of the quantities follows in the next section. Analytically, we see that

$$[R(t,P)]_{ab} = \sum_{i,j=1}^{n} P_{ia}P_{jb}(T_{ij}^t - \pi_j) \tag{17}$$

where $T_{ij}^t := [T^t]_{ij}$.

B.2 SPMS from a Random Process

We can formulate a random process on the graph that precisely gives rise to the entries of the SPMS matrix $R(t,P) = P^T(T^t - \pi^T \otimes \mathbb{1}_n)P$. Let $P_{:,a}$ be the distribution in column a of P. Consider the following:

1. Start in a chosen cluster a
2. Sample a starting node $X_0 \sim P_{:,a}$
3. Run an unbiased random walk $X(s)$ on the graph for t steps
4. Suppose that our walk ends at $X(t) = j$ (with possibly $j = X_0$)
5. Call this walk *accepted* in cluster $P_{:,b}$ with probability P_{jb} for $b = 1,\dots,k$
6. Call this walk *unaccepted* with probability $1 - \sum_{\ell=1}^{k} P_{j\ell}$

All instances of the above random process will either be accepted in one and only one cluster or unaccepted entirely. It is easy to see that the probability that a random walker starting in cluster $P_{:,a}$

is accepted in cluster $P_{:,b}$ after t steps is given by $\sum_{i,j} P_{ia} P_{jb} T_{ij}^t$, where the sum is over $i \in P_{:,a}$ and $j \in P_{:,b}$. Thus, the a, b entry of the stability matrix eq. (15) is precisely the probability that a random walker starting in $P_{:,a}$ is accepted in $P_{:,b}$ after t steps minus the rate of this happening in the long-time limit.

Remark 4. There is a potential issue in the random process above. One can imagine a soft partition P with some rows that sum to more than 1. In this case, a probabilistic interpretation does not hold. This cannot happen in practice since we constrain P to be the elementwise square of a matrix with orthonormal columns, as discussed in appendix C.1. However, one could force this degenerate case by picking distributions poorly. In this situation, one could simply normalize the problematic rows to have unit sum and achieve a probabilistic interpretation with weights proportional to the original matrix entries. It is easy to derive an elementwise upper bound on the change of $R(t, P)$ after this normalization.

B.3 Convergence Rates for SPMS

In this section, we provide the complete proof of theorem 2.1. We first restate the setup.

Suppose $\lambda_1, \ldots, \lambda_n$ are the eigenvalues of L_N, listed in increasing order, for a connected, non-bipartite graph. Recall that $\lambda_i \in [0, 2]$ for all i, $\lambda_1 = 0$, and $\lambda_i > 0$ for $i > 1$. Notice that $T = D^{-1}A$ is similar to $D^{-1/2}AD^{-1/2}$ and deduce that the eigenvalues of T are $1 - \lambda_i$ for $i \in [n]$. It follows that T has an eigenvalue of 1 and all others will be smaller in absolute value. We now give the definition of the (absolute) spectral gaps (definition B.4) of T and L_N.

Definition B.4. Consider an ergodic Markov chain with transition matrix T that has eigenvalues $\lambda_1, \ldots, \lambda_n$ with $\lambda_1 = 1$, the *spectral gap* of T is defined as

$$\lambda_T^* = 1 - \max_{i=2,\ldots,n} |\lambda_i|. \tag{18}$$

Suppose $\mu_1 \leq \cdots \leq \mu_n$ are the eigenvalues of L_N with $\mu_1 = 0$, then the *spectral gap* of L_N is $\lambda_{L_N}^* = \mu_2$.

We now state our main results that relate the behavior of $r(t, P)$ to the spectral gap of the underlying Markov chain. While it is clear that $\lim_{t\to\infty} \operatorname{Tr}[R(t, P)] = 0$, theorem B.1 shows how this rate of convergence depends on λ^*.

Theorem B.1. *Consider a connected, non-bipartite graph G that admits an unbiased random walk with transition matrix T. For any $\epsilon > 0$ and soft partition $P \in [0, 1]^{n\times k}$, there exists a time $t(\epsilon)$ such that for all $t \geq t(\epsilon)$ one has $|\operatorname{Tr}[R(t, P)]| < \epsilon$. Specifically, letting $\pi_{min} = \min_{i\in[n]} \pi_i$, we have*

$$t(\epsilon) = \frac{1}{\lambda^*} \log \frac{2kn}{\epsilon \pi_{min}}. \tag{19}$$

Corollary B.2. *Consider a connected, non-bipartite graph G that admits an unbiased random walk with transition matrix T. For any $\epsilon > 0$ and soft partition P, we have $r(t, P) < \epsilon$ for all $t \geq t(\epsilon)$.*

Proof. This result follows immediately from theorem B.1 and the estimate

$$r(t, P) = \min_{0\leq s\leq t} \operatorname{Tr}[R(s, P)] \leq \min_{0\leq s\leq t} |\operatorname{Tr}[R(s, P)]| \leq |\operatorname{Tr}[R(t, P)]|.$$

□

Notice that $\pi^T = (2|E|)^{-1} d^T$ is the stationary distribution for $T = D^{-1}A$, i.e., it is the unique left 1-eigenvector of T that satisfies $\pi^T \mathbb{1}_n = 1$. With this, we see that our unbiased random walks are *reversible* (definition F.1) with respect to the stationary distribution since

$$T_{xy}\pi_x = \frac{1}{d_x} A_{xy} \frac{d_x}{2|E|} = \frac{A_{xy}}{2|E|} = \frac{T_{yx}}{2|E|} = T_{yx}\pi_y.$$

In our discussion of mixing times, we use total variation distance to measure the difference between probability distributions. We direct the reader to definition F.2 for a precise definition. A result from Levin and Peres (2017) provides a convenient characterization of the total variation distance δ when the underlying set is finite, which is the case of our graphs. We state it as a lemma.

Lemma B.3 (Proposition 4.2 from Levin and Peres (2017)). *For probability distributions $\mu, \nu \in [0,1]^n$, we have*

$$\delta(\mu, \nu) = \frac{1}{2}||\mu - \nu|| = \frac{1}{2}\sum_{i=1}^{n} |\mu_i - \nu_i|. \tag{20}$$

It is well-known that $\lim_{t\to\infty} T^t = \pi^T \otimes \mathbb{1}_n$, i.e., $\lim_{t\to\infty}[T^t]_{ij} = \pi_j$ for all $i, j \in V$. Hence, to describe how quickly a Markov chain converges to the stationary distribution, we consider the function

$$d(t) = \max_{i\in V} \delta(T^t_{i,:}, \pi) \tag{21}$$

where $T^t_{i,:}$ is understood to be the ith row of T^t. We are now ready to define a mixing time.

Definition B.5. For any $\epsilon > 0$, the *mixing time* is defined as

$$t_{\text{mix}}(\epsilon) = \min\{t : d(t) < \epsilon\}. \tag{22}$$

Lemma B.4. *The function $d(t)$ is non-increasing.*

Proof of Lemma B.4. Start by showing that for any Markov chain with transition matrix T on a connected graph $G = (V, E)$ and probability distributions μ and ν (interpreted as row vectors) supported on V, we have

$$\delta(\mu T, \nu T) \leq \delta(\mu, \nu).$$

Lemma B.3 allows us to start by writing

$$\delta(\mu T, \nu T) = \frac{1}{2}\sum_{i=1}^{n} \Big|[\mu T]_i - [\nu T]_i\Big| = \frac{1}{2}\sum_{i=1}^{n} \Big|\sum_{j=1}^{n} \Big(\mu_j T_{ji} - \nu_j T_{ji}\Big)\Big| \leq \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} T_{ji}\Big|\mu_j - \nu_j\Big|$$

$$= \frac{1}{2}\sum_{j=1}^{n} \Big|\mu_j - \nu_j\Big| \sum_{i=1}^{n} T_{ji} = \frac{1}{2}\sum_{j=1}^{n} \Big|\mu_j - \nu_j\Big| = \delta(\mu, \nu).$$

Let $T^s_{i,:}$ be the row vector corresponding to the ith row of T^s. Now, by the definition of d and this fact, we have that for any $t \geq 0$

$$d(t+1) = \max_{i\in V} \delta(T^{t+1}_{i,:}, \pi) = \max_{i\in V} \delta(T^t_{i,:}T, \pi T) \leq \max_{i\in V} \delta(T^t_{i,:}, \pi) = d(t).$$

□

With this property of $d(t)$, observe that for any time $t \geq t_{\text{mix}}(\epsilon)$, we have $d(t) < \epsilon$. Before proving theorem B.1, we state a result from Levin and Peres (2017) that describes the relationship between mixing times and the spectral gap.

Lemma B.5 (Theorem 12.4 from Levin and Peres (2017)). *Let T be reversible with respect to π. Then for any $\epsilon \in (0,1)$ we have*

$$t_{mix}(\epsilon) \leq \frac{1}{\lambda^*} \log\Big(\frac{1}{\epsilon \pi_{min}}\Big). \tag{23}$$

We now provide a proof of theorem B.1.

Proof. Fix some $\epsilon \in (0, 2kn)$ and let $\epsilon' = \epsilon/(2kn)$. By definition, for any $t \geq t_{\text{mix}}(\epsilon')$ we have $\delta(T^t_{x,:}, \pi) < \epsilon'$ for all $x \in V$. For such t, using Lemma B.3, we see that

$$\frac{1}{2}\sum_{y\in V} |T^t_{xy} - \pi_y| \leq \epsilon' \implies \sum_{y\in V} P_{xa}P_{ya}|T^t_{xy} - \pi_y| < 2\epsilon'.$$

Using the formula eq. (17), observe

$$|[R(t,P)]_{aa}| \leq \sum_{x,y\in V} P_{xa}P_{ya}|T^t_{xy} - \pi_y| < 2n\epsilon'.$$

Hence, for all $t \geq t_{\text{mix}}(\epsilon')$,

$$|\operatorname{Tr}[R(t,P)]| = \Big|\sum_{b=1}^{k}[R(t,P)]_{bb}\Big| < 2kn\epsilon' = \epsilon.$$

We use Lemma B.5 to find an upper bound on $t_{\text{mix}}(\epsilon')$ in terms of λ^*. Taking $t(\epsilon)$ to be this upper bound, rounded up, completes the proof in this case. Although uninteresting, when $\epsilon \geq 2kn$, the result is trivial as

$$|\operatorname{Tr}[R(t,P)]| \leq \sum_{a=1}^{k}\sum_{i,j=1}^{n} P_{ia}P_{ja}|T_{ij}^{t} - \pi_j| \leq \sum_{a=1}^{k}\sum_{i,j=1}^{n} P_{ia}P_{ja} \leq k < 2kn.$$

□

C ALGORITHM

We remind the reader of the optimization problem we solve to find optimal soft partitions. Then, we provide a thorough discussion of cluster distinctness. We theoretically demonstrate the quality of the CPQR initialization on stochastic block models.

C.1 CLUSTER DISTINCTNESS

Recall that we find an optimal soft clustering by solving

$$\begin{aligned} \underset{Q\in \text{SO}(k)}{\text{maximize}} \quad & \operatorname{Tr}[R(t,P)] \\ \text{subject to} \quad & P = [V_kQ]^{\odot 2} \end{aligned} \tag{24}$$

To optimize over the special orthogonal group $SO(k)$, we leverage a result from Lie theory (Hall, 2015). Since $SO(k)$ is compact and connected, we have that $\exp$ is a surjection from the Lie algebra, skew-symmetric matrices, to $SO(k)$. Since skew-symmetric matrices are entirely determined by their upper-triangular part, we may substitute these constraints smoothly into the objective function to reduce the problem to unconstrained optimization in $\mathbb{R}^{k(k-1)/2}$. Thus, we reformulate eq. (7) as

$$\begin{aligned} \text{maximize} \quad & \operatorname{Tr}[R(t,P)] \\ \text{subject to} \quad & U \text{ strictly upper-triangular} \\ & Q = \exp(U - U^T) \\ & P = [V_kQ]^{\odot 2}. \end{aligned} \tag{25}$$

One could argue that it would be advantageous to parameterize P as a function of the entire eigenspace, i.e., defining $P = [V_nQ]^{\odot 2}$ where $Q \in \mathbb{R}^{n\times k}$ with orthonormal columns. However, we note that choosing the first k eigenvectors of L_N makes the optimization problem simpler and more practically approachable. Moreover, our experiments show this restriction does not limit our ability to find good distributions.

A desirable property of a good soft partition is cluster distinctness. To construct a pathological example, suppose we have a distribution p that captures one cluster very well; then we could produce a matrix $P = [p \dots p] \in [0,1]^{n\times k}$ that does well in terms of SPMS but ignores the rest of the community structure. One could argue that the optimization problem eq. (25) does not explicitly enforce the optimal distributions to be distinct. As seen empirically, this goal seems to be implicitly accomplished through the structure of the eigenspace V_k and the definition of $P = [V_kQ]^{\odot 2}$. We provide a heuristic argument for this behavior in remark 5.

Remark 5. Heuristically, we can see that the structure of the eigenspace V_k enforces cluster distinctness. Since orthogonal transformations preserve norm, the row norms of V_k equal those of V_kQ. Hence,

$$||[V_k]_{i,:}||^2 = \sum_{j=1}^{k}[V_k]_{ij}^2 = \sum_{j=1}^{k}[V_kQ]_{ij}^2 = \sum_{j=1}^{k}[(V_kQ)^{\odot 2}]_{ij} = \sum_{j=1}^{k}P_{ij}.$$

Observe that $D^{1/2}\mathbb{1}_n$ is a 0-eigenvector of L_N and thus, the first column of V_k. After proper normalization, the i coordinate of this vector is precisely $\sqrt{\pi_i}$. Now, by the computation above, we conclude that the mass in each row is at least the stationary component. Intuitively, this enforces our distributions to cover the graph because any node with significant stationary mass must receive significant mass in some distribution. It is not unreasonable to say that our objective function in eq. (25) encourages cluster localization since diffuse clusters would be poor in terms of SPMS. We claim that cluster coverage coupled with localization yields the distinctness that we see empirically.

While the structure of V_k and our parameterization of P seem to yield cluster distinctness, if one would like to explicitly enforce distinctness of the columns of P^*, a regularized variant of eq. (25) could be used. Specifically, the off-diagonal terms can be penalized. That is,

$$\begin{aligned} \underset{Q \in \mathrm{SO}(k)}{\text{maximize}} \quad & \mathrm{Tr}[R(t,P)] - \gamma \sum_{a \neq b} [R(t,P)]_{ab} \\ \text{subject to} \quad & P = [V_k Q]^{\odot 2} \end{aligned} \tag{26}$$

where $\gamma > 0$ is a tuning parameter specifying the degree to which distribution overlap is penalized. Typically, we have taken $\gamma = 1/(k-1)$ since there are $k^2 - k$ off-diagonal terms and only k terms in the trace. This regularization penalizes cluster overlap because the off-diagonal terms capture information about cluster similarity. For instance, if clusters a and b place high mass on a similar set of well-connected nodes, then the flow of accepted random walkers between these clusters will be large and consequently, $[R(t,P)]_{ab}$ will be large.

Although this regularization intuitively enforces distinctness, on model graphs, the soft partitions returned by solving eq. (26) are indistinguishable from those returned by the original optimization problem. Thus, we typically just solve eq. (25). The variant without regularization has produced sufficiently distinct optimal distributions on model graphs and real datasets throughout our testing. We hypothesize that there is some equivalence between maximizing the diagonal terms and minimizing the off-diagonal terms of $R(t,P)$ in our optimization.

C.2 Initialization for Stochastic Block Models

We show that our initial guess $P_0 = [V_k Q_0]^{\odot 2}$ is close to the soft partition encoding the ground-truth cluster structure in SBMs. Theorem 1 of Damle et al. (2019) provides a bound describing how close $V_k Q_0$ is to a normalized indicator matrix, under some assumptions about the structure of the graph. Here, a normalized indicator matrix is a matrix $X \in [0,1]^{n \times k}$ satisfying $X^T X = I_k$. Precisely, this theorem assumes that $n > 4$ and there exists a normalized indicator matrix W satisfying

$$||WW^T - V_k V_k^T||_2 \leq \epsilon < \frac{2^{-k}}{\sqrt{n}}. \tag{27}$$

If Q_0 is the polar factor computed by CPQR-based spectral clustering with V_k as input, then there exists a permutation matrix $\hat{\Pi}$ such that

$$||W\hat{\Pi} - V_k Q_0||_F \leq 2\epsilon\sqrt{kn} + \mathcal{O}(\epsilon^2). \tag{28}$$

We note that eq. (27) holds with high probability in the SBM and is reasonable for clusterable graphs. We direct the reader to the original paper for more justification. Intuitively, this result shows that the subspace spanned by the columns of V_k after some orthogonal transformation is close to the subspace spanned by the indicator vectors of cluster membership. Hence, the matrix $V_k Q_0$ may be used for cluster assignment.

Since we are using $P_0 = [V_k Q_0]^{\odot 2}$ as our initialization, we may draw an analogous result to the aforementioned theorem to show that we initialize SPMS optimization near a sensible soft clustering. If W is a normalized indicator matrix, then the columns of $W^{\odot 2}$ describe probability distributions with disjoint support. If we assume eq. (27), the set of distributions defined by $W^{\odot 2}$ would capture the underlying cluster structure and be a suitable candidate for maximizing $r(t,P)$.

Theorem C.1. *Let $W \in \mathbb{R}^{n \times k}$ be a normalized indicator matrix and suppose $n > 4$. Let $V_k \in \mathbb{R}^{n \times k}$ with orthonormal columns satisfy eq. (27). If Q_0 is the polar factor computed by CPQR-based spectral clustering with V_k as input, then there exists a permutation matrix $\hat{\Pi}$ such that for $P_0 = [V_k Q_0]^{\odot 2}$, we have*

$$||(W\hat{\Pi})^{\odot 2} - P_0||_F \leq 2C\epsilon\sqrt{kn} + \mathcal{O}(\epsilon^2) \tag{29}$$

where $C = \max_{i\in[n],j\in[k]} \left|[W\hat{\Pi}]_{ij} + [V_k Q_0]_{ij}\right|$.

Proof. The result follows from

$$\begin{aligned}
||(W\hat{\Pi})^{\odot 2} - P_0||_F &= ||(W\hat{\Pi})^{\odot 2} - (V_k Q_0)^{\odot 2}||_F \\
&= \Big(\sum_{i=1}^{n}\sum_{j=1}^{k}\left([(W\hat{\Pi})^{\odot 2}]_{ij} - [(V_k Q_0)^{\odot 2}]_{ij}\right)^2\Big)^{1/2} \\
&= \Big(\sum_{i=1}^{n}\sum_{j=1}^{k}\left([W\hat{\Pi}]_{ij}^2 - [V_k Q_0]_{ij}^2\right)^2\Big)^{1/2} \\
&= \Big(\sum_{i=1}^{n}\sum_{j=1}^{k}([W\hat{\Pi}]_{ij} + [V_k Q_0]_{ij})^2([W\hat{\Pi}]_{ij} - [V_k Q_0]_{ij})^2\Big)^{1/2} \\
&\leq C||W\hat{\Pi} - V_k Q_0||_F \\
&\leq 2C\epsilon\sqrt{kn} + \mathcal{O}(\epsilon^2)
\end{aligned}$$

where we appealed to Theorem 1 of Damle et al. (2019) in the last step. □

Although the constant C is not universal, we note that it will be typically small (less than one) for real-world applications. That is, assuming that clusters are of the same size, the non-zero elements of both $W\hat{\Pi}$ and $V_k Q_0$ will be about $\sqrt{k/n}$ in absolute value. Moreover, since $W\hat{\Pi}$ and $V_k Q_0$ have orthonormal columns, then in the worst case we would have $C = 2$, which is acceptable.

Remark 6. In the case where $Q_0 \notin SO(k)$, the result of theorem C.1 holds after applying a permutation matrix A that swaps any two columns when multiplied on the right. Then, $Q_0 A \in SO(k)$ and both $P_0 = (V_k Q_0 A)^{\odot 2}$ and $(W\hat{\Pi}A)^{\odot 2}$ are column stochastic. We now have,

$$\begin{aligned}
||(W\hat{\Pi}A)^{\odot 2} - P_0||_F &= ||(W\hat{\Pi})^{\odot 2}A - (V_k Q_0)^{\odot 2}A||_F \\
&= ||(W\hat{\Pi})^{\odot 2} - (V_k Q_0)^{\odot 2}||_F \\
&\leq 2C\epsilon\sqrt{kn} + \mathcal{O}(\epsilon^2).
\end{aligned}$$

D Core-Periphery Substructure

In this section, we remind the reader of the construction of the core metric, provide additional discussion, and prove the result from section 4.1. Furthermore, we thoroughly discuss the three proposed methods for choosing cores.

D.1 Formal Definition of Core-Periphery Structure

A graph exhibits core-periphery structure if it has a densely connected, cohesive core and a sparsely connected periphery. As discussed in Borgatti and Everett (2000), although core-periphery detection has been well-studied, there is no conventional definition or generating model of this structure. To motivate our discussion, we opt for the definition given by Zhang et al. (2015), which states that a graph exhibiting core-periphery structure is an instance of a 2-Block SBM with $p_{11} > p_{21} > p_{22}$. Here, p_{ij} is the probability of an edge forming between blocks i and j. Hence, block 1 would be the core, and block 2 would be the periphery. Note that periphery nodes are more likely to form edges with core nodes than with each other. Consequently, a graph with core-periphery structure strongly exhibits degree inhomogeneity, with most of the edges concentrated in the core.

D.2 Core Metric

Our core metric evaluates the quality of a proposed core in terms of the flow of random walks. This yields an intuitive and direct approach for analyzing the core-periphery substructure of clusters, constituting a major advantage over existing methods. We now motivate the core metric and discuss

its properties. First, we define a function that describes the flow of random walks between subsets of V (definition D.1).

Definition D.1. For a connected, non-bipartite graph $G = (V, E)$, positive integer t, and probability distribution p on V, we describe the flow of accepted walks between subsets of V with the function $f_t^p : \mathcal{P}(V) \times \mathcal{P}(V) \to [0, 1]$ given by

$$f_t^p(A, B) = \sum_{i \in A} \sum_{j \in B} p_i p_j T_{ij}^t. \tag{30}$$

Notice that $f_t^p(A, B)$ gives a probability similar to the random process from appendix B.2. Considering just one distribution on the graph, $f_t^p(A, B)$ describes the probability that a starting node is sampled from subset A and the random walker is accepted in subset B after t steps. With the SBM characterization of core-periphery structure given in appendix D.1, we ought to see that this probability flow is large within the core and smaller between the periphery and core. That is, denoting $C \subset V$ and $P \subset V$ to be the core and periphery, respectively, we should have $f_t^p(C, C) > f_t^p(P, C)$ for a suitable distribution p. In practice, for a given cluster we wish to analyze, this distribution p will be the corresponding column of the output of SPMS optimization. With this, we are ready to define the core metric (definition D.2), the objective function that we optimize to find a core within a given cluster.

Definition D.2. Let $C \subset V$ be a non-empty set which we will call the core and let its complement $P = C'$ be the periphery. We define the core metric as

$$\varphi_t^p(C) = \frac{f_t^p(C, C)}{|C| + f_t^p(P, C)}. \tag{31}$$

Remark 7. When taking $P = C'$, there is a flaw in the interpretation of the periphery as it will contain the core of other clusters. We emphasize that the focus of our work is specifically on identifying a core, and that taking the periphery to be the complement is done solely for algorithmic convenience. Since these nodes outside of the cluster being analyzed will have negligible mass in p, they will not meaningfully contribute to $f_t^p(P, C)$. That is, taking $P = C'$ makes no difference from a computational perspective. If one were interested in finding a periphery after a core has been chosen, one could compute the set of nodes A that have significant mass in p (this threshold should vary by application) and set $P = A \setminus C$.

We iteratively grow the core and seek a maximum of $\varphi_t^p(C)$, which favors cores that retain random walks originating within them. Since SPMS optimization concentrates mass on core nodes, a well-chosen core yields large $f_t^p(C, C)$ and small $f_t^p(P, C)$ – the latter reflecting low periphery-to-core transition probability and low peripheral mass. The metric also naturally penalizes extreme core sizes. An oversized core is punished by the cardinality term in the denominator, as adding non-core nodes produces diminishing returns in $f_t^p(C, C)$ while $|C|$ constantly grows. An undersized core exhibits small $f_t^p(C, C)$ and consequently, small $\varphi_t^p(C)$. Hence, we expect $\varphi_t^p(C)$ to exhibit a local maximum at an optimal core, behavior we confirm on several model graphs. We now provide an intuitive sufficient condition for increasing the core metric at an iteration.

Theorem D.1. *Consider a graph $G = (V, E)$, a distribution p on V, and a positive integer t. Let $C \subset V$ be a non-empty core and let $P = C'$ be the periphery. Suppose for a node $k \notin C$ that*

$$\frac{f_t^p(C, \{k\})}{f_t^p(C, C)} \geq \frac{f_t^p(P, \{k\})}{|C|}. \tag{32}$$

Then, $\varphi_t^p(C \cup \{k\}) \geq \varphi_t^p(C)$ if k satisfies,

$$\frac{f_t^p(\{k\}, C \cup \{k\})}{\sum_{n \in C} f_t^p(\{n\}, C)} \geq \frac{1}{|C|}. \tag{33}$$

Proof of theorem D.1. Let $|C| = n$. Define $f_t^p(A) := f_t^p(A, A)$ for brevity. Since the distribution and time step are fixed, we write φ_t^p and f_t^p as φ and f, respectively, for notational clarity. Compute

$$\begin{aligned}\varphi(C \cup \{k\}) - \varphi(C) &= \frac{f(C \cup \{k\})}{n + 1 + f(P \setminus \{k\}, C \cup \{k\})} - \frac{f(C)}{n + f(P, C)} \\ &\geq \frac{f(C \cup \{k\})}{n + 1 + f(P, C \cup \{k\})} - \frac{f(C)}{n + f(P, C)} \\ &= \frac{f(C \cup \{k\})(n + f(P, C)) - f(C)(n + 1 + f(P, C \cup \{k\}))}{(n + 1 + f(P, C \cup \{k\}))(n + f(P, C))}.\end{aligned}$$

Hence, $\varphi(C \cup \{k\}) - \varphi(C) \geq 0$ if

$$f(C \cup \{k\})(n + f(P, C)) \geq f(C)(n + 1 + f(P, C \cup \{k\})) \iff \frac{f(C \cup \{k\})}{f(C)} \geq \frac{n + 1 + f(P, C \cup \{k\})}{n + f(P, C)}.$$

We now look at both sides of the inequality individually. First,

$$\begin{aligned}\frac{f(C \cup \{k\})}{f(C)} &= \frac{\sum_{i,j \in C} p_i p_j T_{ij}^t + \sum_{i \in C} p_i p_k T_{ik}^t + \sum_{j \in C} p_k p_j T_{kj}^t + p_k^2 T_{kk}^t}{\sum_{i,j \in C} p_i p_j T_{ij}^t} \\ &= 1 + \frac{\sum_{i \in C} p_i p_k T_{ik}^t + \sum_{j \in C} p_k p_j T_{kj}^t + p_k^2 T_{kk}^t}{f(C)}\end{aligned}$$

and

$$\frac{n + 1 + f(P, C \cup \{k\})}{n + f(P, C)} = \frac{n + 1 + \sum_{i \in P} \sum_{j \in C} p_i p_j T_{ij}^t + \sum_{i \in P} p_i p_k T_{ik}^t}{n + \sum_{i \in P} \sum_{j \in C} p_i p_j T_{ij}^t} = 1 + \frac{1 + \sum_{i \in P} p_i p_k T_{ik}^t}{n + f(P, C)}.$$

We can now see that $\varphi(C \cup \{k\}) - \varphi(C) \geq 0$ if and only if

$$\frac{\sum_{i \in C} p_i p_k T_{ik}^t + \sum_{j \in C} p_k p_j T_{kj}^t + p_k^2 T_{kk}^t}{f(C)} \geq \frac{1 + \sum_{i \in P} p_i p_k T_{ik}^t}{n + f(P, C)}.$$

The idea now is to deduce a stronger inequality that is intuitive. Since $f(P, C) \geq 0$, it is sufficient to have the inequality

$$\frac{\sum_{i \in C} p_i p_k T_{ik}^t + \sum_{j \in C} p_k p_j T_{kj}^t + p_k^2 T_{kk}^t}{f(C)} \geq \frac{1 + \sum_{i \in P} p_i p_k T_{ik}^t}{n} \tag{34}$$

to ensure that φ increases. Now, consider the left side of the above and use the assumed estimate eq. (32) to get

$$\frac{\sum_{i \in C} p_i p_k T_{ik}^t + \sum_{j \in C} p_k p_j T_{kj}^t + p_k^2 T_{kk}^t}{f(C)} \geq \frac{\sum_{i \in P} p_i p_k T_{ik}^t}{n} + \frac{\sum_{j \in C} p_k p_j T_{kj}^t + p_k^2 T_{kk}^t}{f(C)}.$$

Combining this estimate with eq. (34) implies that

$$\frac{\sum_{j \in C} p_k p_j T_{kj}^t + p_k^2 T_{kk}^t}{f(C)} \geq \frac{1}{n}$$

is sufficient for $\varphi(C \cup \{k\}) - \varphi(C) \geq 0$. Noticing that the numerator of the left-hand side is precisely $f(\{k\}, C \cup \{k\})$ and that $f(C) = \sum_{n \in C} f(\{n\}, C)$ yields the result. □

Remark 8. The assumed inequality eq. (32) in theorem D.1 is purely technical and crucial for the sake of the proof. Although it would be possible to produce choices of the graph, core, added node, and algorithm parameters such that eq. (32) does not hold, in practice this assumption is rather weak. Since $|f_t^p| \leq 1$, we see that the denominator on the right-hand side of eq. (32) will be significantly larger than that of the left-hand side. Moreover, for a suitable choice of C and k, we will have that the summands of $f_t^p(C, \{k\})$ will be larger than those of $f_t^p(P, \{k\})$. Hence, it is not unrealistic to assume eq. (32).

We have derived an intuitive sufficient (but not necessary) condition for increasing the core metric at each iteration: the metric is guaranteed to grow if the contribution of the added node is "better than average" in the sense of eq. (33). While theorem D.1 characterizes when the metric improves, we have yet to specify how each additional node is chosen.

D.3 Choosing a Core

In this section, we propose a greedy approach for selecting the core that contiguously adds neighbors of the previous core at each iteration. We compute φ_t^p for each iteration of the core and seek a local maximum. Alternative approaches are also explored.

Given a graph for which we wish to identify core-periphery substructure, we first run SPMS optimization with time step t to obtain a set of distributions that capture the degree inhomogeneity of each cluster. For a given cluster we wish to analyze, we set the distribution p to be the corresponding column of P^* so that we may compute φ_t^p. Denote $N_i \subseteq V \setminus C_i$ to be the set of neighbors of C_i, the core at iteration i. We now propose three different iterative methods to choose a core:

- **Contiguous:** Initialize $C_1 = \{\arg\max_{i\in[n]} p_i\}$. For $i > 1$, compute N_{i-1} and let $k = \arg\max_{j\in N_{i-1}} p_j$. Set $C_i = C_{i-1} \cup \{k\}$.
- **Rapid:** Set C_i to be the nodes corresponding to the i largest entries in p.
- **Metric-Contiguous:** Initialize $C_1 = \{\arg\max_{k\in V} \varphi_t^p(\{k\})\}$. For $i > 1$, compute N_{i-1} and let $k = \arg\max_{j\in N_{i-1}} \varphi_t^p(C_{i-1} \cup \{j\})$. Set $C_i = C_{i-1} \cup \{k\}$.

In an application setting, it would be sensible to choose some lower limit ℓ and upper limit $m < n$ on the size of an acceptable core (which may vary based on the application) and iterate the chosen process to get cores $C_1, \ldots, C_m$. The optimal core will be taken as $C^* = \arg\max_{C_i, \ell\le i\le m} \varphi_t^p(C_i)$.

The Contiguous approach presented above is natural since a core ought to be connected. As can be seen empirically in appendix E.1 and appendix E.2, this method is successful at finding suitable cores on several model graphs. Although it is sensible to choose the starting node $k \in [n]$ to be the one that maximizes $\varphi_t^p(\{k\})$, numerically, we find that this often coincides with $\{\arg\max_{i\in[n]} p_i\}$ and is not worth the extra computation.

At the cost of core contiguity, the Rapid approach circumvents the expensive computation of the neighbors at each iteration. Although this may produce disconnected cores at some iterations, the reduced computation may be worth it for very large graphs. In fact, we choose this approach when analyzing large, real-world networks in section 5.3 and appendix E.5.

The Metric-Contiguous approach is the most expensive of the proposed methods as it involves computing the core metric for each neighbor at each iteration. Although this method is not guaranteed to find the optimal subset of V for each potential core size, it does naturally seek a maximum of the core metric while enforcing contiguity. Hence, this way of choosing the core may be favorable for graphs for which the computation is tractable. In our testing, we found that "exhaustively" choosing the core in this way yields results similar to the previous, more-efficient Contiguous method, with the cores differing by only a few nodes. If computational resources are not a concern, there is a natural ordering to the quality of these methods. Canonically, Metric-Contiguous is preferred, followed by Contiguous, and subsequently by Rapid.

E Numerical Results

In this section, we provide additional tests on model graphs and real data.

E.1 Stochastic Block Model

First, we consider a stochastic block model (Holland et al., 1983) with four blocks, each containing 100 nodes. The intra-cluster edge formation probability is $p = 0.7$ and the inter-cluster edge formation probability is $q = 0.1$. The nodes are labeled so that $0, \ldots, 99$ comprises the first block, $100, \ldots, 199$ comprises the second block, and so on. We run our optimization with $k = 4$. For a given instance of this SBM, the optimal solution P^* is given in fig. 5.

For this test, if one were to produce a hard partition of the nodes from this set of distributions by assigning node j to cluster $\arg\max P^*_{j,:}$, we would achieve exact recovery. However, this method of producing a clustering loses much of the information about cluster inhomogeneity and node ambiguity that our approach is flexible enough to capture. Hence, we will not be concerned with exact

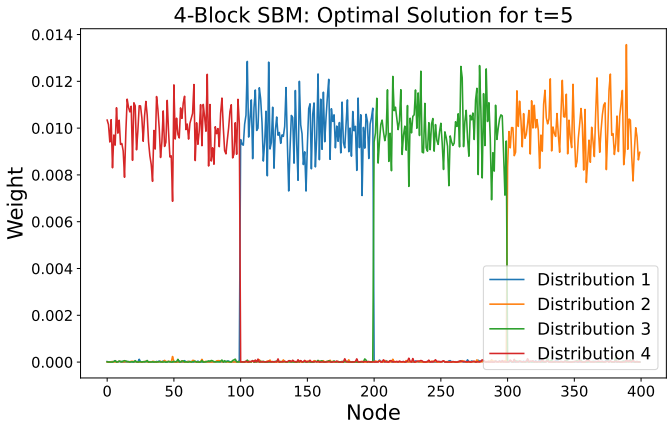


Figure 5: The optimal set of distributions P^* after gradient descent converges. We see that each column of P^* is localized to a given block, showing that our algorithm correctly identified the underlying cluster structure.

recovery in this paper and direct the reader to Damle et al. (2019) for a more thorough discussion of exact recovery in the stochastic block model.

Next, we consider a variant of the stochastic block model that highlights the ability of SPMS optimization to detect low-degree ambiguous nodes. We further develop this idea in appendix E.3. We first draw an instance of the 4-block SBM with intra-cluster edge formation probability $p = 0.7$ and inter-cluster edge formation probability $q = 0.1$. The nodes are labeled so that $0, \ldots, 99$ comprises the first block, $100, \ldots, 199$ comprises the second block, and so on. We then add ambiguous nodes labeled $400, \ldots, 499$ that have a 0.01 probability of connecting with any node in the graph independently. We solve eq. (25) with $k = 4$. In fig. 6, we visualize this graph and the optimal distributions obtained by SPMS optimization. Observe that the optimal distributions are localized to the ground-truth communities and that the 100 ambiguous nodes are assigned negligible mass in each distribution.

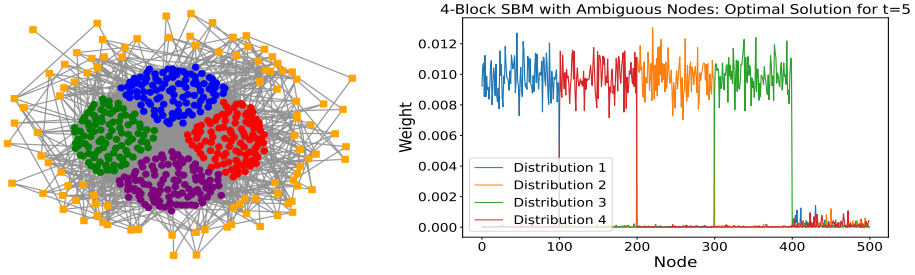


Figure 6: Visualization of the SBM after the ambiguous nodes, marked by the yellow squares, have been added (left). Optimal distributions P^* that come from SPMS optimization (right).

We may also analyze SPMS-CP on a modified formulation of the SBM. That is, we generate independent and identically distributed two-block SBMs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, each with 200 nodes, with $p_{11} = 0.4$, $p_{12} = 0.1$, and $p_{22} = 0.05$. Notice that these random graphs will exhibit core-periphery structure according to the definition from Zhang et al. (2015) described in appendix D.1. We then connect a node in G_1 to a node in G_2 with probability 0.005 independently for each pair of nodes to obtain a test graph G. This construction yields random graphs that have strong cluster structure, with communities V_1 and V_2, and core-periphery substructure.

We run SPMS optimization on G with $k = 2$ and $t = 5$ to obtain an optimal set of distributions P^*. To analyze the internal structure of G_1, we take p to be the first column of P^*. We run the Contiguous procedure described in appendix D.3 with $m = 200$ to grow a connected core and compute φ_5^p at each iteration. In fig. 7 below, we see a visualization of the optimal distributions P^*

for one sample from this distribution of random graphs. Also, in fig. 7, we plot φ_5^p as a function of the cores chosen for each acceptable size $|C_i| \in [200]$ for 50 random graphs sampled from the aforementioned procedure. Note that the nodes are labeled so that $0, \ldots, 99$ correspond to the core of G_1 and nodes $200, \ldots, 299$ correspond to that of G_2.

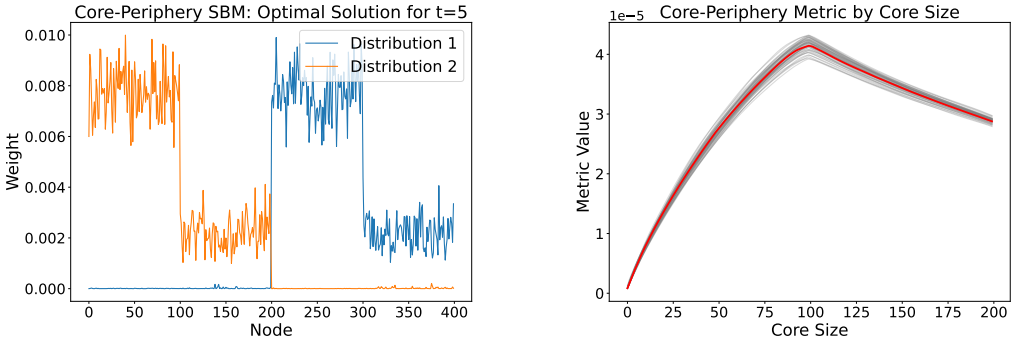


Figure 7: The optimal distributions for a soft partition of G (left). A plot of the core metric for 200 iterations of the core for 50 trials (right). The plot of the core metric corresponding to the optimal distributions on the left is highlighted in red. All other trials are in gray. Notice that SPMS optimization captures the inhomogeneity of each cluster and correctly places higher mass on the core nodes. Also note that our metric exhibits a maximum when the core size is 100.

We can see from the plot that the core metric is maximized when the core has 100 nodes with only slight variance in the optimized metric value. Since the optimal core for G_1 was $C_{100} = \{0, \ldots, 99\}$, we see that our approach recovers the "true" core-periphery substructure for this cluster. We note how the core metric monotonically increased while true core nodes were being added but strictly decreased after all such nodes were included.

E.2 Gaussian Random Geometric Graphs

We first provide the specific details for the random geometric graphs considered in section 5.1. Consider four bivariate Gaussian random variables Z_i, $1 \leq i \leq 4$, with distinct means (x_i, y_i), for $x_i, y_i \in \{0, 2\}$, and common covariance matrix $\frac{2}{5} I_2$. We sample from a multinomial distribution with 100,000 trials and 4 categories, each with equal probability, to get counts $n_1, \ldots, n_4$. We sample n_i points from Z_i to produce a set of 100,000 data points in the plane. To generate a graph, we form an edge between any two points if their distance is within $\epsilon = 0.22$ and consider the largest connected component. We note that the values of ϵ and variance were chosen by sweeping over a range of possibilities and selecting those that yielded a spectral gap indicative of a non-trivial, yet tractable, clustering problem.

In section 5.1, we saw how SPMS-CP canonically and consistently identified well-connected cores in Gaussian random geometric graphs. We may also visualize how the core grows as our algorithm iterates. We draw a random geometric graph with 5000 initial data points and run the Contiguous method to find cores of various sizes. In fig. 8, we see the original data points in $\mathbb{R}^2$ along with a visualization of the core for sizes 300, 400, and 500. The orange nodes correspond to the distribution Z_i we are analyzing, and the gray nodes correspond to the other distributions. Nodes that are colored red are those included in the core at the given iteration. We see how the chosen core nodes are concentrated around the mean of the generating distribution for core sizes that are nearly optimal. Moreover, observe how the core is contiguously and naturally grown as our algorithm iterates.

E.3 Flexibility Advantages Over Row-Normalized Soft Clustering Methods

We first show the optimal distributions produced by SPMS optimization on the model graphs discussed in section 5.2 over several parameter regimes. Once again, we note nodes $0 - 99, \ldots, 300 - 399$ represent communities, nodes $400 - 449$ represent the strong ambiguous block, and nodes $450 - 499$ represent the weak ambiguous block. We averaged the mass assigned to a given node in each distribution over all 200 trials, plotting standard deviation error bars every 10 nodes. The

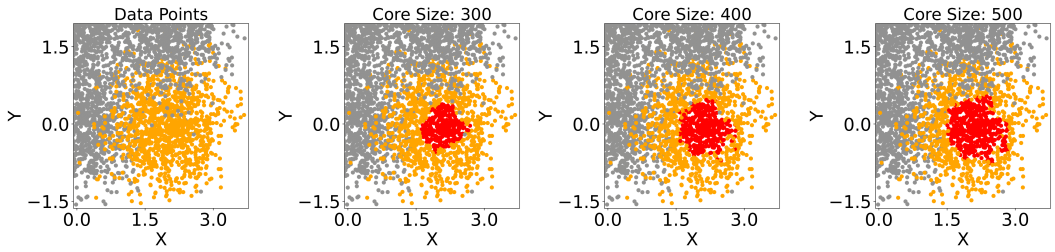
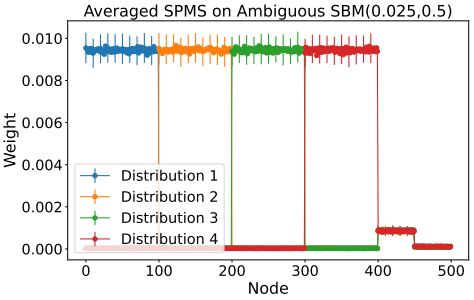
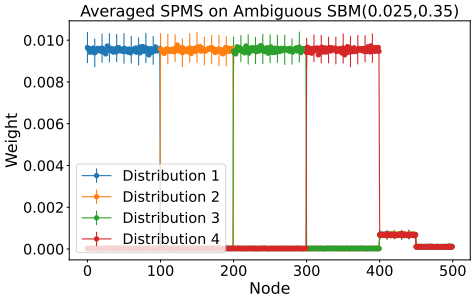


Figure 8: Visualizations of the data in $\mathbb{R}^2$ and the cores for sizes 300, 400, and 500. The optimal core contained 321 nodes.

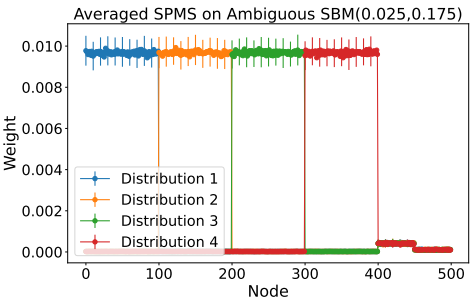
results are summarized in fig. 9. Notice for large values of p_s, SPMS assigns higher, but still negligible, mass to the strong planted block than it assigns to the weak planted block. The ambiguous nodes were consistently identified, as evidenced by their relatively low variance. If one observed this output, one would be able to separate the low-degree ambiguous nodes from the high-degree ambiguous nodes.



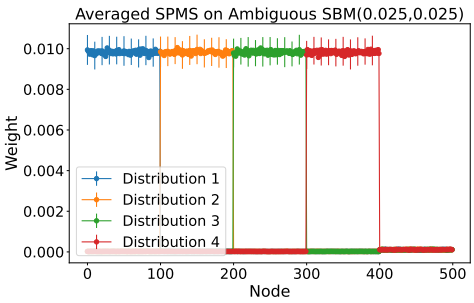
Optimal distributions for $(p_w, p_s) = (0.025, 0.5)$.



Optimal distributions for $(p_w, p_s) = (0.025, 0.35)$.



Optimal distributions for $(p_w, p_s) = (0.025, 0.175)$.



Optimal distributions for $(p_w, p_s) = (0.025, 0.025)$.

Figure 9: Optimal distributions produced by SPMS optimization on SBM with planted blocks over several parameter regimes.

We saw in section 5.2 that row-normalized methods like FGC and fuzzy k-means failed to distinguish the strong and weak planted blocks at some parameter regimes. In tables 1 to 4, we see the average mass assigned to each node in each distribution across both types of ambiguous nodes, for both FGC and fuzzy k-means. This data is reported for $p_s = 0.5, 0.25$ with standard deviations in parentheses. Once again, we sorted the optimal distributions in decreasing order since the cluster labeling produced by these methods is inconsistent across trials. Also, the average total variation distance (definition F.2) to the uniform distribution is reported.

We can see that fuzzy k-means returned roughly uniform distributions for both the strong and weak ambiguous nodes in both parameter regimes. As such, it is intuitive that the logistic regression could not differentiate the planted blocks. At $(p_s, p_w) = (0.25, 0.025)$, the optimal distributions returned by FGC were nearly identical, but non-uniform, for the strong and weak blocks. Therefore, it is

Table 1: Fuzzy k-means with $(p_s, p_w) = (0.5, 0.025)$.

Node Type	Distribution 1	Distribution 2	Distribution 3	Distribution 4	TVD from Uniform
Strong	0.303; (0.029)	0.259; (0.016)	0.232; (0.016)	0.207; (0.018)	0.065; (0.029)
Weak	0.324; (0.045)	0.255; (0.023)	0.223; (0.021)	0.198; (0.019)	0.086; (0.042)

Table 2: FGC with $(p_s, p_w) = (0.5, 0.025)$.

Node Type	Distribution 1	Distribution 2	Distribution 3	Distribution 4	TVD from Uniform
Strong	0.656; (0.085)	0.262; (0.092)	0.054; (0.044)	0.028; (0.022)	0.454; (0.048)
Weak	0.411; (0.078)	0.311; (0.046)	0.226; (0.079)	0.051; (0.038)	0.246; (0.073)

Table 3: Fuzzy k-means with $(p_s, p_w) = (0.25, 0.025)$.

Node Type	Distribution 1	Distribution 2	Distribution 3	Distribution 4	TVD from Uniform
Strong	0.316; (0.038)	0.258; (0.020)	0.227; (0.019)	0.199; (0.020)	0.079; (0.037)
Weak	0.319; (0.041)	0.255; (0.021)	0.225; (0.019)	0.201; (0.018)	0.080; (0.039)

Table 4: FGC with $(p_s, p_w) = (0.25, 0.025)$.

Node Type	Distribution 1	Distribution 2	Distribution 3	Distribution 4	TVD from Uniform
Strong	0.385; (0.097)	0.332; (0.072)	0.221; (0.113)	0.062; (0.091)	0.247; (0.146)
Weak	0.401; (0.120)	0.327; (0.079)	0.212; (0.111)	0.059; (0.088)	0.257; (0.153)

reasonable that its corresponding classifier achieved an AUC near 0.5. However, in fig. 2, we see that FGC was able to distinguish the planted blocks for large p_s. When $(p_s, p_w) = (0.5, 0.025)$, we see that the maximum mass assigned to a strong ambiguous node was higher than that of a weak ambiguous node. Hence, FGC was naturally able to separate the planted blocks in this extreme parameter regime.

E.4 Data Cleaning

Before providing additional road network tests, we would like to comment on the data cleaning procedure and some pathological behavior induced by our data generation method. Our data was obtained using OSMnx (Boeing, 2025), a package to download and analyze road networks using data from OpenStreetMap. Choosing a suitable center and sufficient radius, we generated a graph encompassing the aforementioned cities in which nodes represent intersections and edges represent roads. The package also returns additional node data, including spatial coordinates, and edge data, including road type and length. Occasionally, OSMnx produces edges that are too long to be reasonable[2]. Thus, we removed all edges over 1500 meters in length. We obtained the final data set by taking the largest connected component and removing self-loops. However, inherent to the way the data was generated, the road networks exhibit pathological behavior near the periphery.

We note that our data generation method introduced artificial boundary effects. For example, in the San Jose distribution pictured in fig. 3, we observe an unnaturally high concentration of mass in the bottom right corner of the graph. Since OSMnx only captures data within a specified radius, missing data beyond this boundary blocks the outflow of random walks to the southeast. Thus, the bottom right of the graph appears more central to the cluster than it actually is.

E.5 More Tests on Real Data

We analyze the performance of our algorithm on additional large road networks. As part of this, we discuss a memory-efficient implementation of SPMS that will be useful to recreate our experiments.

[2]That is, upon inspection we identified edges that were several kilometers long and connected nodes across the entirety of the graph, which were clearly erroneous.

For real-world applications, we would like SPMS optimization to be scalable to graphs with millions of nodes. Clearly, sparse implementations of common matrix operations should be used to leverage the sparse structure of real networks. Unfortunately, explicitly storing large dense matrices like $\pi^T \otimes \mathbb{1}_n$ is problematic. To circumvent this, we write the objective function of eq. (25) in terms of low-dimensional and sparse objects. Specifically, we suggest

$$\mathrm{Tr}[R(t,P)] = \mathrm{Tr}[P^T T^t P] - \langle P^T \mathbb{1}_n, P^T \mathrm{diag}(\pi)\mathbb{1}_n \rangle. \tag{35}$$

With this, we are free to run SPMS optimization on large data sets. In fact, the most expensive operation in our method is the computation of the eigenspace V_k. To run these experiments, it suffices to have 80 GB of memory and 20 hours of compute time on an NVIDIA A6000 GPU. We have seen empirically that the optimization problem eq. (25) converges quickly.[3]

Similarly to the experiment in section 5.3, we ran SPMS optimization on a road network encompassing Ithaca, Cortland, and Binghamton in upstate New York. We ran SPMS optimization with $k = 2$ and $t = 5$ to obtain an optimal P^*. Note that $k = 3$ is also a suitable choice for this graph, but due to the relatively small size of Cortland and its close proximity to Ithaca, we opt for $k = 2$. Figure 10 is a visualization of the induced hard partition and the distribution capturing Binghamton.

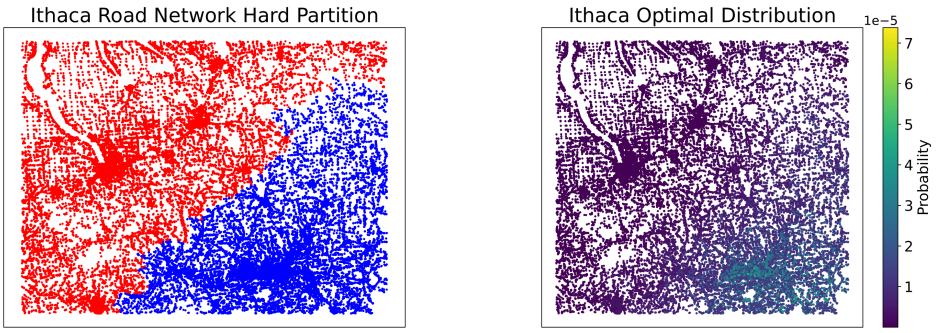


Figure 10: The hard partition induced by P^* naturally divides the graph into a cluster containing Binghamton and a cluster containing both Ithaca and Cortland (left). The distribution capturing Binghamton (right).

As a final road network test, we analyze the region of Texas encompassing Fort Worth and Dallas. We cleaned the OSMnx data exactly as before and ran SPMS optimization with $k = 2$ and $t = 5$. Once again, in fig. 11 we see a visualization of the hard partition along with one of the optimal distributions.

We were also interested in seeing how SPMS optimization behaves as the number of clusters increases. We consider the Bay Area road network. Using the eigengap heuristic discussed in appendix A.2, we observed that $k = 12$ and $k = 20$ would also be sensible choices. Appendix G contains table 5 which shows the standardized eigengaps. We ran SPMS optimization for both choices. Appendix G also includes fig. 15, a visualization of the hard partitions for these values of k.

Finally, we include plots of the core metric as a function of core size on the Bay Area road network (with $k = 12$) analyzed in section 5.3. To efficiently compute φ_t^p, we write $f_t^p(A,B)$ as a matrix-vector product $p_A^T([T^t]_{A,B})p_B$.[4] We choose our time step to be $t = 5$, our increment parameter to be $\eta = 25$, and look for cores of size at least $\ell = 1000$. We set the maximum acceptable core size to be $m = 75000$. On two different clusters, which we call cluster A and cluster B, we iterate our algorithm using the Rapid method from appendix D.3 to obtain cores with sizes equal to each

[3] In appendix E.7, fig. 14 displays the value of SPMS as a function of the gradient descent iteration for varied initializations on real networks. We see that SPMS converges after roughly 40 iterations in each case.

[4] Here, for non-empty $A, B \subseteq V$, we denote p_A to be the vector of length $|A|$ consisting of entries of p corresponding to the indices in A (we define p_B the same way) and denote $[T^t]_{A,B}$ to be the sub-matrix of T^t with rows and columns specified by index sets A and B, respectively.

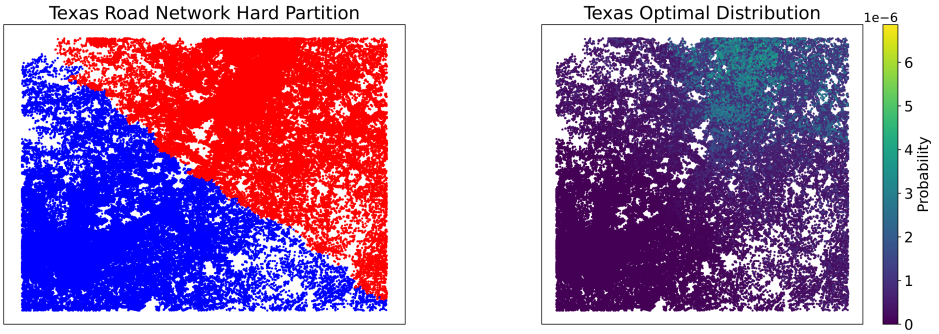


Figure 11: The hard partition induced by P^* naturally divides the graph into a cluster containing Dallas and a cluster containing Fort Worth (left). The distribution capturing Dallas (right).

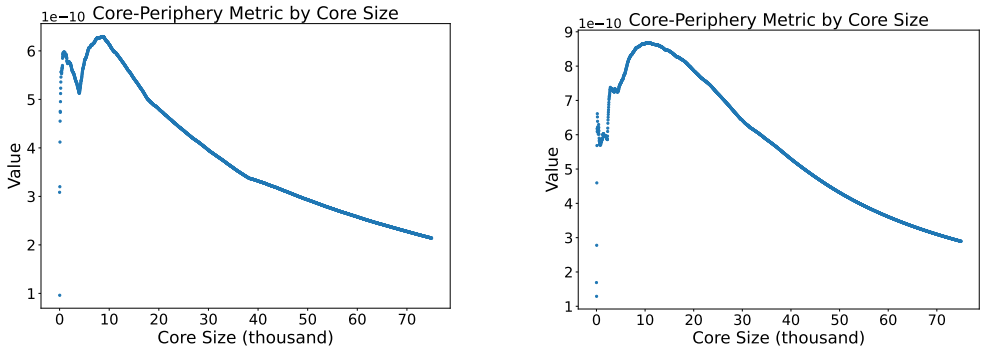


Figure 12: Plots of φ_t^p as a function of core size for each multiple of η less than m. Note the bimodality of the core metric on cluster A (left) and the unimodality of the core metric on cluster B (right).

multiple of η less than m. The optimal core in cluster A contained 8625 nodes. Figure 12 shows plots of the core metric as a function of core size for each cluster we analyzed. For cluster A, notice that φ_t^p attains two local maxima, but the larger core is favored. On the other hand, the core metric exhibits only one local maximum on cluster B.

E.6 Cluster Resolution and SPMS Optimization Across Time Scales

It is well known that the hyperparameter t in the traditional Markov stability framework allows for tunable cluster resolution. Indeed, as t increases, random walks are free to explore more of the graph, favoring larger clusters. This idea is explored in Delvenne et al. (2010). We show that SPMS optimization behaves similarly, with t determining the cluster resolutions that SPMS supports.

To test this, we consider hierarchical stochastic block models, which we call HSBMs. These model graphs are special cases of a traditional SBM, consisting of eight blocks of 50 nodes each, with a three-level hierarchy. Blocks 1-4 and blocks 5-8 each form a super-community. Within each super-community, pairs of blocks form two smaller communities. For instance, in the first super-community, blocks 1 and 2 form a community and blocks 3 and 4 form a community. Within each community, each block forms a sub-community. The edge formation probability within each sub-community is 0.7. For nodes in distinct blocks in the same community, the edge formation probability is 0.3. For nodes in the same super-community but different communities, the edge formation probability is 0.1. Finally, nodes in different super-communities form an edge with probability 0.02. This way, an HSBM has natural community structure at different resolutions. As per the construction, the canonical choices of k are 2, 4, or 8.

We note that the value of SPMS is not comparable across choices of k. Indeed, suppose an optimal set of distributions has already been chosen for some k_1. If $k_2 > k_1$, then the soft partition with k_2 distributions can always yield a higher objective value by repeating the distributions of k_1.[5] For the sake of this test, we define a proxy for the gain in objective value incurred by considering cluster structure at a finer resolution. That is, suppose $k_1 < \cdots < k_N$ are the candidate numbers of clusters that we are searching for. We define the *level gain* at level k_i and time t as

$$g(k_i, t) = \begin{cases} \mathrm{Tr}[R^*(t, k_i)] - \frac{k_i}{k_{i-1}} \mathrm{Tr}[R^*(t, k_{i-1})], & 1 < i \leq N \\ \mathrm{Tr}[R^*(t, k_i)], & i = 1 \end{cases}$$

where $R^*(t, k)$ is the optimized stability matrix (definition 2.2) at the given values of t and k. When $1 < i \leq N$, the normalizing term accounts for the natural increase in the metric that could arise due to cluster duplication. Hence, the level gain describes the increase in the metric caused by the graph structure at a finer resolution.

We drew 30 instances of HSBMs from the above distribution and ran SPMS optimization for $k_1 = 2$, $k_2 = 4$, and $k_3 = 8$ and for all $t \in \{1, \ldots, 30\}$. We computed the level gain for each value of k and t. To ensure all values are on the same scale and to isolate the effect of t, we computed $g(k_i, t)/g(k_i, 1)$. We averaged these values over the 30 trials and plotted the means, with standard deviation error bars, as a function of t. The results are shown in fig. 13.

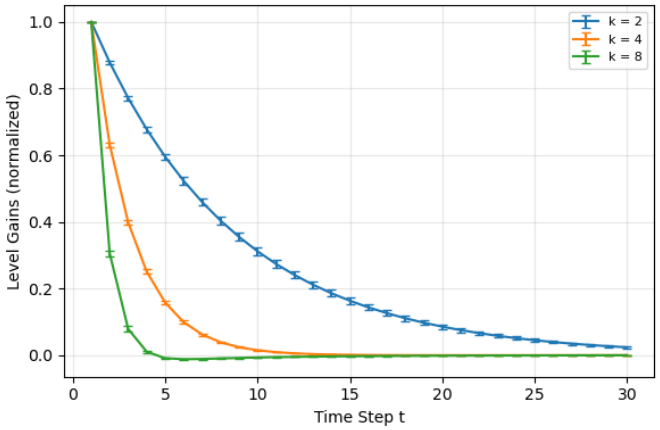


Figure 13: The normalized level gains of SPMS optimization across different community sizes and time steps.

We can see that the time step t determines the cluster resolutions that SPMS supports. Indeed, for the finest cluster resolution, $k = 8$, the level gain over the $k = 4$ resolution level goes to zero quickly as t increases. That is, as the random walks explore more of the graph, the finer clusters become less stable and have diminishing contributions to the objective. A similar pattern occurs with the $k = 4$ resolution scale compared to the coarse $k = 2$ resolution scale. Finally, we see that the coarsest resolution remains stable for the longest time.

E.7 SPMS Optimization with Varied Initializations

In this section, we analyze how the value of SPMS changes as the Adam optimizer iterates, depending on the initialization. For each of the road network graphs and parameter choices discussed, we run SPMS optimization for 60 iterations of gradient descent, with three different initializations. We first take $Q_0 = I$, which corresponds to the upper-triangular matrix $U = 0$. We randomly initialize Q_0 by taking the upper-triangular entries in U to be draws from the uniform distribution on $[0, 1)$. Finally, we consider the usual CPQR initialization as described in Damle et al. (2019). One can see how SPMS optimization is influenced by its initialization below in fig. 14.

[5] We note that we always observed distinct clusters numerically since we parameterize soft partitions as a function of the eigenspace V_k.

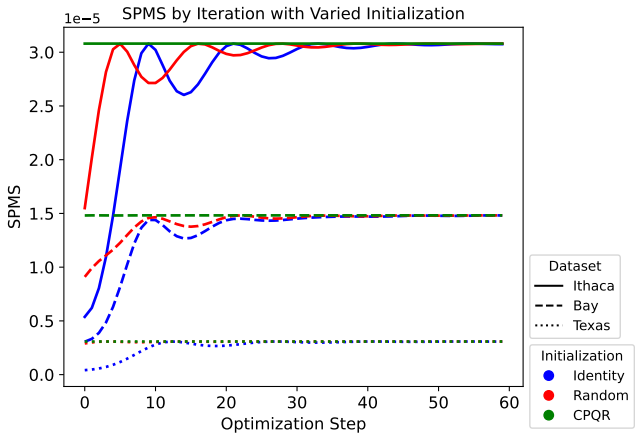


Figure 14: The SPMS value for 60 iterations of gradient descent with differing initialization on the three road network graphs.

Unsurprisingly, we see that the identity initialization results in poor clustering for early iterations. In this case, $P_0 = [V_k]^{\odot 2}$. Thus, the first cluster places high mass on nodes with large stationary component, which is generally a poor clustering. However, this is hardly a problem as SPMS converges after roughly 40 iterations in each case. The random initialization seems to perform better than the identity for early iterations, but converges after roughly the same number of gradient descent steps. In our testing, we never observed convergence to local maxima under random initializations. After computing 500 random initial guesses and running gradient descent for 50 steps, the distribution of optimized SPMS values was contained in a relatively small set for each road network. Interestingly, we see that the initial guesses coming from CPQR-based spectral clustering are close to optimal, with the objective value barely changing compared to the other initializations. From these tests, we conclude that the use of the CPQR initialization is not necessary but drastically accelerates the convergence of SPMS optimization. These empirical results, coupled with the theoretical discussion of appendix C.2, justify our use of the CPQR initialization.

F Definitions

Definition F.1. An ergodic Markov chain with transition matrix T is *reversible* with respect to the stationary distribution π if the *detailed balance condition* holds, i.e., for all $x, y \in V$ we have

$$T_{xy}\pi_x = T_{yx}\pi_y.$$

Definition F.2. Consider a measurable space $(X, \mathcal{F})$ with probability measures μ and ν. The *total variation distance* between μ and ν is

$$\delta(\mu, \nu) = \sup_{A \in \mathcal{F}} |\mu(A) - \nu(A)|. \tag{36}$$

G NUMERICAL RESULTS

k	1	2	3	4	5	6	7	8
Standardized Eigengap	-1.0905	-1.0218	0.3239	-1.1822	0.2887	-0.3417	0.5661	0.3613

k	9	10	11	12	13	14	15	16
Standardized Eigengap	-0.8668	0.0853	-1.0463	2.1608	0.4039	-0.2089	-0.3795	-0.6764

k	17	18	19	20	21	22	23	24
Standardized Eigengap	1.6351	-0.9273	0.7074	2.7090	-0.6210	0.1258	-0.4630	-0.5418

Table 5: A standardization of the eigengaps $\lambda_{k+1} - \lambda_k$ for $k = 1, \ldots, 24$ for the Bay Area road network. We use the usual z-score standardization to produce this dataset. Notice the relatively large eigengaps for $k = 3$, 12, and 20 clusters.

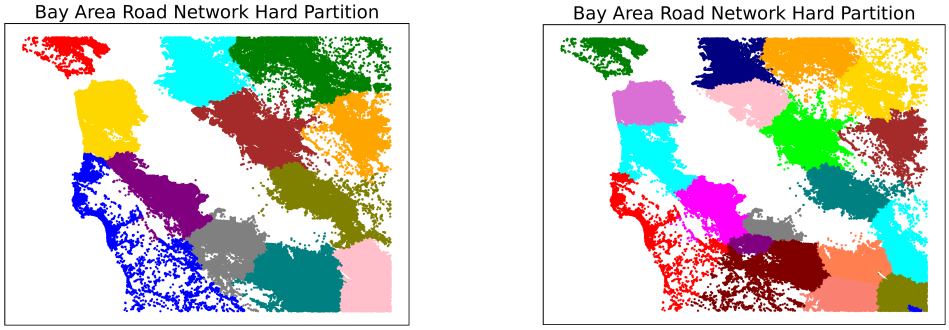


Figure 15: The hard partition corresponding to $k = 12$ (left). The hard partition corresponding to $k = 20$ (right). Notice how SPMS optimization naturally divides the network into k contiguous sections. Although there is no natural geographic interpretation of these clusters, we emphasize that SPMS optimization behaves appropriately as k increases.