SIAM J. MATRIX ANAL. APPL.
Vol. 45, No. 1, pp. 685–711

AN ESCAPE TIME FORMULATION FOR SUBGRAPH DETECTION AND PARTITIONING OF DIRECTED GRAPHS*

ZACHARY M. BOYD†, NICOLAS FRAIMAN‡, JEREMY L. MARZUOLA§, PETER J. MUCHA¶, AND BRAXTON OSTING‖

Abstract. We provide a rearrangement based algorithm for detection of subgraphs of k vertices with long escape times for directed or undirected networks that is not combinatorially complex to compute. Complementing other notions of densest subgraphs and graph cuts, our method is based on the mean hitting time required for a random walker to leave a designated set and hit the complement. We provide a new relaxation of this notion of hitting time on a given subgraph and use that relaxation to construct a subgraph detection algorithm that can be computed easily and a generalization to K-partitioning schemes. Using a modification of the subgraph detector on each component, we propose a graph partitioner that identifies regions where random walks live for comparably large times. Importantly, our method implicitly respects the directed nature of the data for directed graphs while also being applicable to undirected graphs. We apply the partitioning method for community detection to a large class of models and real-world data sets.

Key words. escape time, partitioning, bang-bang

MSC code. 68R10

DOI. 10.1137/23M1553790

1. Introduction.

Subgraph detection and graph partitioning are fundamental problems in network analysis, each typically framed in terms of identifying a group or groups of vertices of the graph so that the vertices in a shared group are well connected or "similar" to each other in their connection patterns while the vertices in different groups (or the complement group) are "dissimilar." The specific notion of connectedness or similarity is a modeling choice, but one often assumes that edges connect similar vertices, so that, in general, the detected subgraph is dense and the "communities" identified in graph partitioning are very often more connected within groups than between groups (assortative communities). In the present work, we will propose and analyze a novel and natural subgraph detection model based on escape times, after surveying briefly some of the major existing paradigms.

*Received by the editors February 16, 2023; accepted for publication (in revised form) by D. J. Higham November 3, 2023; published electronically February 26, 2024.
https://doi.org/10.1137/23M1553790
Funding: The work of the first author was supported by the National Science Foundation (NSF) of grant DMS-2137511. The work of the first and fourth authors was supported by the ARO under MURI award W911NF-18-1-0244. The work of the third author was supported by the NSF grants DMS-1909035 and FRG DMS-2152289. The work of the fourth author was supported by the NSF grant BCS-2140024. The work of the fifth author was supported by the NSF grants DMS-1752202 and DMS-2136198.
†Department of Mathematics, Brigham Young University, Provo, UT 84602 USA (boyd@mathematics.byu.edu).
‡Department of Statistics and Operations Research, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599 USA (fraiman@email.unc.edu).
§Department of Mathematics, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599 USA (marzuola@math.unc.edu).
¶Department of Mathematics, Dartmouth College, Hanover, NH 03755 USA (peter.j.mucha@dartmouth.edu).
‖Department of Mathematics, University of Utah, Salt Lake City, UT 84112 USA (osting@math.utah.edu).

The identification of subgraphs with particular properties is a long-standing pursuit of network analysis with various applications. Dense subgraphs as assortative communities might represent coordinating regions of interest in the brain [48, 8] or social cliques in a social network [49]. In biology, subgraph detection plays a role in discovering DNA motifs and in gene annotation [31]. In cybersecurity, dense subgraphs might represent anomalous patterns to be highlighted and investigated (e.g., [80]). See [44] for a recent survey and a discussion of alternative computational methods. As noted there, some of the existing algorithms apply to directed graphs, but most do not.

In the corresponding computer science literature, much of the focus has been on approximation algorithms since the dense k-subgraph is NP-hard to solve exactly (a fact easily seen by a reduction from the k-clique problem). An algorithm that on any input (G, k) returns a subgraph of order k (that is, k vertices or "nodes"; note, we will sometimes refer to the "size" of a graph or subgraph to be the number of vertices, not the number of edges) with average degree within a factor of at most $n^{1/3-\delta}$ from the optimum solution, where n is the order of graph G and $\delta \approx 1/60$ was proposed in [25]. This approximation ratio was the best known for almost a decade until a log-density based approach yielded $n^{1/4+\varepsilon}$ for any $\varepsilon > 0$ [10]. This remains the state-of-the-art approximation algorithm. On the negative side it has been shown [46], assuming the exponential time hypothesis, that there is no polynomial-time algorithm that approximates to within an $n^{1/(\log\log n)^c}$ factor of the optimum. Variations of the problem where the target subgraph has size at most k or at least k have also been considered [1].

Depending on the application of interest, one might seek one or more dense subgraphs within the larger network, a collection of subgraphs to partition the network (i.e., assign a community label to each node), or a set of potentially overlapping subgraphs (see, e.g., [79]). While the literature on "community detection" is enormous (see, e.g., [27, 28, 29, 60, 63] as reviews), a number of common thematic choices have emerged. Many variants of the graph partitioning problem can be formalized as a (possibly constrained) optimization problem. One popular choice minimizes the total weight of the cut edges while making the components roughly equal in size [64]. Another common choice maximizes the total within-community weight relative to that expected at random in some model [55]. Other proposed objective functions include ratio cut weight [13], and approximate "surprise" (improbability) under a cumulative hypergeometric distribution [70]. However, most of these objectives are NP-hard to optimize, leading to the development of a variety of heuristic methods for approximate partitioning (see the reviews cited above for many different approaches).

Some of the methods that have been studied are based on the Fielder eigenvector [26], multicommunity flows [42], semidefinite programming [5, 6, 7], expander flows [4], single commodity flows [38], or Dirichlet partitions [57, 56, 76]. We note, in particular, the classic WalkTrap method [59], which, similar to the methods presented in this paper, emphasizes the idea of "escape" from communities; though in contrast to the present work, in addition to its limitation to undirected graphs, WalkTrap is typically used to identify candidates that are then comparatively evaluated with other measures (e.g., modularity). A final relevant heuristic is Personal PageRank and its variants [32], which find localized clusters of "similar" nodes in the sense that random walkers starting as a seed node will visit the "similar" nodes relatively frequently on a particular timescale. This is, of course, different from escape time as such, but Personal PageRank has been used in various network clustering schemes [33, 21, 78]

and has been connected theoretically to (sub)graph conductance and spectral graph theory (e.g., [2]).

Whichever choice is made for the objective and heuristic, the identified communities can be used to describe the mesoscale structure of the graph and can be important in a variety of applications (see, e.g., the case studies considered in [63]). Subgraphs and communities can also be important inputs to solving problems like graph traversal, finding paths, trees, and flows; while partitioning large networks is often an important subproblem for complexity reduction or parallel processing in problems such as graph eigenvalue computations [11], finding stationary distributions [22] breadth-first search [16], triangle listing [19], PageRank [62], Personalized PageRank [3], and related algorithms [30, 34]. In addition, clustering methods potentially related to our own for directed graphs have been introduced in [61] using the map equation, as well as in [9] using Motif Adjacency matrices. More recently, the ideas of doubly stochastic scaling have been implemented in [40], which potentially could also be useful for discovering small scale directed features. A review of the results up to 2013 on directed graph clustering can be found in [45].

Developed in parallel with the many different computational methods for identifying communities and small dense subgraphs, theoretical results have provided insights about the limits of detecting such structures in various settings, including the detectability limits for weak community separation in simple stochastic block models (see, e.g., [23, 54, 68]) and for small cliques planted in random graphs (see, e.g., [53, 67]). Of most relevance to the problems we study in the present contribution, small cliques planted in Erdős–Rényi graphs of N nodes with mean density ρ are undetectable by spectral methods for cliques with fewer than $\sqrt{N\rho/(1-\rho)}$ nodes [53].

In the present work, we consider a different formulation of the subgraph detection problem, wherein we aim to identify a subgraph with a long mean exit time—that is, the expected time for a random walker to escape the subgraph and hit its complement. Importantly, this formulation inherently respects the possibly directed nature of the edges. This formulation is distinct from either maximizing the total or average edge weight in a dense subgraph and minimizing the edge cut (as a count or suitably normalized) that is necessary to separate a subgraph from its complement. Furthermore, explicitly optimizing for the mean exit time to identify subgraphs may in some applications be preferred as a more natural quantity of interest. For example, in studying the spread of information or a disease on a network, working in terms of exit times is more immediately dynamically relevant than structural measurements of subgraph densities or cuts. Similarly, the development of respondent-driven sampling in the social survey context (see, e.g., [51, 75]) is primarily motivated by there being subpopulations that are difficult to reach (so we expect they often also have high exit times on the directed network with edges reversed). We thus argue that the identification of subgraphs with large exit times is at least as interesting—and typically related to—those subgraphs with large density and or small cut. Indeed, random walker diffusion on a network and assortative communities are directly related in that the modularity quality function used in many community detection algorithms can be recovered as a low-order truncation of a "Markov stability" auto-correlation measurement of random walks staying in communities [39]. However, the directed nature of the edges is fully respected in our escape time formulation of subgraph detection presented here (cf. random walkers moving either forward or backward along edges in the Markov stability calculation [52] that rederives modularity for a directed network [41]).

From an optimization point of view, the method presented here can be viewed as a rearrangemnet method or a Merriman–Bence–Osher (MBO) scheme [47] as applied to

Poisson solves on a graph. Convergence of MBO schemes is an active area of research in a variety of other scenarios; see [18, 35] in the case of continuum mean curvature flows, [15, 74] in a graph Allen–Cahn type problem, and [36] for a volume constrained MBO scheme on undirected networks. Similarly, proving convergence rates for our algorithm by determining quantitative bounds on the number of interior iterations required for a given ε is an important question for the numerical method and its applications to large data sets. Importantly, the method for subgraph detection that we develop and explore, and then extend to a partitioner, is inherently capable of working on directed graphs without any modification. Also, searching for related graph problems where this type of rearrangement algorithm for optimization can be applied will be an important endeavor.

1.1. A new formulation in graphs. Let $G = (V, E)$ be a (strongly) connected graph (undirected or directed; we use the term "graph" throughout to include graphs that are possibly directed), with adjacency matrix A with element A_{ij} indicating presence/absence (and possible weight) of an edge from i to j. We define the (out-)degree matrix D to be diagonal with values $D_{ii} = \sum_j A_{ij}$. For weighted edges in A this weighted degree is typically referred to as "strength," but we will continue to use the word "degree" throughout to be this weighted quantity. Consider the discrete time Markov chain M_n for the random walk described by the (row stochastic) *probability transition matrix,* $P := D^{-1}A$. The *exit time from* $S \subset V$ is the stopping time $T_S = \inf\{n \geq 0 : M_n \in S^c\}$. The *mean exit time from* S *of a node* i is defined by $\mathbb{E}_i T_S$ (where $\mathbb{E}_i$ is the expectation if the walker starts at node i) and is given by v_i, where v is the solution to the system of equations

$$(I - P)_{SS} v_S = 1_S, \tag{1.1a}$$

$$v_{S^c} = 0\,, \tag{1.1b}$$

where the subscript S represents restriction of a vector or matrix to the indices in S. The *average mean escape time (MET) from* S is then

$$\tau(S) = \frac{1}{|V|} \sum_{\nu \in V} v_\nu, \tag{1.2}$$

representing the mean exit time from S of a node chosen uniformly at random in the graph (noting that $v_\nu = 0$ for $\nu \in S^c$). We are interested in finding vertex sets (of fixed size) having large MET, as these correspond to sets that a random walker would remain in for a long time. Thus, for fixed $k \in \mathbb{N}$, we consider the *subgraph detection problem,*

$$\max_{\substack{S \subset V \\ |S| = k}} \tau(S). \tag{1.3}$$

Multiplying (1.1a) on the left by D, we obtain the equivalent system

$$Lv = d \text{ on } S, \tag{1.4a}$$

$$v = 0 \text{ on } S^c\,, \tag{1.4b}$$

where $L = D - A$ is the (unnormalized, out-degree) graph Laplacian, e is the vector of 1's, and $d = De$ is the out-degree vector. We denote the solution to (1.4) by $v = v(S)$. While this problem could be viewed as an interesting problem in and of itself, the authors are unaware how to locate optimizers of such a problem without

using NP–hard algorithms. As a result, we propose below a regularized version of (1.3) for which we can develop efficient optimization algorithms. To regularize, for $\varepsilon > 0$, we will consider the approximation to (1.4),

$$\left[L + \varepsilon^{-1}(1-\phi)\right] u = d, \tag{1.5}$$

where ϕ is a vector and action by $(1-\phi)$ on the left is interpreted as multiplication by the diagonal matrix $I - \mathrm{diag}(\phi)$. Note that for $(1-\phi)(v) > 0$ for any $v \in V$, $L + \varepsilon^{-1}(1-\phi)$ is invertible since the kernel of L is given by then vector of all 1's, denoted by e from henceforward. We denote the solution $u = u_\varepsilon$. Formally, for $\phi = \chi_S$, the characteristic function of S, as $\varepsilon \to 0$, the vector $u_\varepsilon \to v_S$ where v_S satisfies (1.4). We can also define an associated approximate MET,

$$E_\varepsilon(\phi) := \frac{1}{|V|}\|u_\varepsilon\|_{\ell^1(V)} = \frac{1}{|V|}\left\|\left[L + \varepsilon^{-1}(1-\phi)\right]^{-1} d\right\|_{\ell^1(V)}, \tag{1.6}$$

where as $\varepsilon \to 0$, we have that $E_\varepsilon(\chi_S) \to \frac{1}{|V|}\|v_S\|_{\ell^1(V)} = \tau(S)$. We then arrive at the following *relaxed subgraph detection problem:*

$$\max_{\substack{0 \le \phi \le 1 \\ \langle \phi, 1\rangle = k}} E_\varepsilon(\phi), \tag{1.7}$$

which we solve and study in this paper. For small $\varepsilon > 0$, we will study the relationship between the subgraph detection problem (1.3) and its relaxation (1.7).

We are also interested in finding node partitions with high MET in the following sense: Given a vertex subset $S \subset V$, a random walker that starts in S should have difficulty escaping to S^c and a random walker that starts in S^c should have difficulty escaping to S.[1] This leads to the problem $\max_{V = S \amalg S^c} \tau(S) + \tau(S^c)$. More generally, for a vertex partition, $V = \amalg_{\ell \in [K]} S_\ell$ with $[K] = \{1, 2, \ldots, K\}$, we can consider

$$\max_{V = \amalg_{\ell \in [K]} S_\ell} \sum_{\ell \in [K]} \tau(S_\ell). \tag{1.8}$$

Unfortunately, (1.8) is trivially maximized when all nodes fall in the same partition element, giving an infinite MET. In subsection 2.2.2, we argue that the alternative formulation that tries to solve

$$\min_{V = \amalg_{\ell \in [K]} S_\ell} \sum_{\ell \in [K]} \frac{1}{1 + \delta |V| \tau(S_\ell)}, \tag{1.9}$$

where $\delta > 0$ is more appropriate. Similar to the subgraph detection problem, an algorithm for efficient optimization of (1.9) is unknown to the authors. Hence, we seek a regularized version where we can similarly compute optimizers using combinatorially simple, scalable methods. The solution embodies the idea that in a good partition a random walker will transition between partition components very infrequently. A first approximation to (1.8) is

$$\max_{V = \amalg_{\ell \in [K]} S_\ell} \sum_{\ell \in [K]} E_\varepsilon(\chi_{S_\ell}). \tag{1.10}$$

[1] This assumption of symmetry is perhaps not always necessary in applications, and there is nothing mathematically necessary about it. The advantage of this assumption is that it allows us to consider high escape time partitions without privileging one partition element over another.

We can make an additional approximation by relaxing the constraint set. Define the admissible class

$$\mathcal{A}_K = \left\{ \{\phi_\ell\}_{\ell\in[K]} \colon \phi_\ell \in \mathbb{R}_+^{|V|} \text{ and } \sum_{\ell\in[K]} \phi_\ell = 1 \right\}.$$

Observe that the collection of indicator functions for any K-partition of the vertices is a member of $\mathcal{A}_K$. Furthermore, we can see that $\mathcal{A}_K \cong (\Delta_K)^{|V|}$, where Δ_K is the unit simplex in K dimensions. Thus, the extremal points of $\mathcal{A}_K$ are precisely the collection of indicator functions for a K-partition of the vertices. For $\delta > 0$, a relaxed version of the modified graph partitioning problem (1.9) can be formulated as

(1.11)
$$\min_{\{\phi_\ell\}_{\ell\in[K]}\in\mathcal{A}_K} \tilde{E}_{\delta,\varepsilon}\left(\{\phi_\ell\}_{\ell\in[K]}\right), \quad \text{where} \quad \tilde{E}_{\delta,\varepsilon}\left(\{\phi_\ell\}_{\ell\in[K]}\right) = \sum_{i=1}^{K} \frac{1}{1+\delta|V|E_\varepsilon(\phi_i)}.$$

For small $\varepsilon > 0$, we will study the relationship between the graph partitioning problem (1.9) and its relaxation (1.11). An important feature of (1.11) is that it can be optimized using rearrangement methods that effectively introduces a volume normalization for the partition sets, while optimization of (1.8) results in favoring one partition being full volume. We will discuss this further in section 2.2.2 below.

1.2. Outline of this paper. In section 2, we lay the analytic foundation for rearrangement methods for both the subgraph detection and partitioning problems. We prove the convergence of the methods to local optimizers of our energy functionals in both cases and establish the fact that our numerical methods improve the energy in the desired fashion. To begin, we establish properties of the gradient and Hessian of the functionals $E_\varepsilon(\phi)$ for vectors $0 \le \phi \le 1$. Then, using those properties, we introduce rearrangement methods for finding optimizers and prove that our optimization schemes improve the energy. Then, we discuss how to adapt these results to the partitioning problem. Lastly, we demonstrate how one can easily add a semisupervised component to our algorithm.

In section 3, we apply our methods to a variety of model graphs, as well as some empirical data sets to assess their performance. In the subgraph setting, we consider how well we do detecting communities in a family of model graphs related to stochastic block models, made up of a number of random Erdős–Rényi (ER) communities of various sizes and on various scales. The model graphs are designed such that the overall degree distribution is relatively similar throughout. We demonstrate community detectability and algorithm efficacy thresholds by varying a number of parameters in the graph models. We also consider directed graph models of cycles connected to ER graphs, on which our methods perform quite well. For the partitioners, we also consider related performance studies over our model graph families, as well as on a large variety of clustering data sets.

We conclude in section 4 with a discussion including possible future directions and applications of these methods.

2. Analysis of our proposed methods. In this section, we first analyze the relaxed subgraph detection problem (1.7) and the relaxed graph partitioning problem (1.11). Then, we propose and analyze computational methods for the problems. As noted above, we assume throughout that the graph is (strongly) connected.

2.1. Analysis of the relaxed subgraph detection problem and the relaxed graph partitioning problem. For fixed $\varepsilon > 0$ and $\phi \in [0,1]^{|V|}$, denote the operator on the left-hand side (LHS) of (1.5) by $L_\phi := D - A + \frac{1}{\varepsilon}(1-\phi)$.

LEMMA 2.1 (discrete maximum principle). *Given the regularized operator L_ϕ and a vector $f > 0$, we have $(L_\phi^{-1} f)_\nu > 0$ for all $\nu \in V$. Without strong connectivity, this result still holds (with $>$ replaced by $\geq$) as long as there are no leaf nodes.*

Proof. Writing $L_\phi = \left(D + \frac{1}{\varepsilon}(1-\phi)\right) - A$, we observe that

$$\begin{aligned} L_\phi^{-1} &= \left(\left(D + \frac{1}{\varepsilon}(1-\phi)\right)\left(I - \left(D + \frac{1}{\varepsilon}(1-\phi)\right)^{-1} A\right)\right)^{-1} \\ &= \left(I - \left(D + \frac{1}{\varepsilon}(1-\phi)\right)^{-1} A\right)^{-1} \left(D + \frac{1}{\varepsilon}(1-\phi)\right)^{-1} \\ &= \sum_{n=0}^{\infty} \left[\left(D + \frac{1}{\varepsilon}(1-\phi)\right)^{-1} A\right]^n \left(D + \frac{1}{\varepsilon}(1-\phi)\right)^{-1}. \end{aligned}$$

Since all entries in the corresponding matrices are positive (by strong connectivity), the result holds. ∎

For simplicity of notation in the following derivations, we denote the potential as $X := \varepsilon^{-1}(1-\phi)$ as well as using X and diag X interchangeably where needed. We can then consider the related energy functional

$$E(X) := \left\| [L+X]^{-1} d \right\|_{\ell^1(V)} = \|u\|_{\ell^1(V)}. \tag{2.1}$$

In particular, with this choice of notation, $E(X) = E_\varepsilon(\phi)$.

LEMMA 2.2. *The gradient of $E(X)$ with respect to X is given by*

$$\nabla E = -u \odot v, \tag{2.2}$$

where $\odot$ denotes the Hadamard product and

$$u = (L+X)^{-1} d, \quad v = (L+X)^{-T} e, \tag{2.3}$$

and here and throughout we take $A^{-T} = (A^{-1})^T$. The Hessian of $E(X)$ with respect to X is then given by

$$H = \nabla^2 E = (L+X)^{-1} \odot W + (L+X)^{-T} \odot W^T, \tag{2.4}$$

where

$$W := u \otimes v,$$

where $\otimes$ is the Kronecker (or outer) product.

Proof. Write e_j as the indicator vector for the jth entry. First, differentiating (2.3) with respect to X_j, we compute

$$(L+X)\frac{\partial u}{\partial X_j} = -e_j \odot u \qquad \Longrightarrow \qquad \frac{\partial u}{\partial X_j} = -\langle e_j, u \rangle (L+X)^{-1} e_j.$$

Taking the second derivative, we obtain

$$\begin{aligned}(L+X)\frac{\partial^2 u}{\partial X_j \partial X_k} &= -e_j \left\langle e_j, \frac{\partial u}{\partial X_k} \right\rangle - e_k \left\langle e_k, \frac{\partial u}{\partial X_j} \right\rangle \\ &= e_j \langle e_k, u\rangle \left\langle e_j, (L+X)^{-1} e_k \right\rangle + e_k \langle e_j, u\rangle \left\langle e_k, (L+X)^{-1} e_j \right\rangle,\end{aligned}$$

which implies that

$$\begin{aligned}&\frac{\partial^2 u}{\partial X_j \partial X_k} \\ &\quad = \left\langle e_j, (L+X)^{-1} e_k \right\rangle \langle e_k, u\rangle (L+X)^{-1} e_j + \left\langle e_k, (L+X)^{-1} e_j \right\rangle \langle e_j, u\rangle (L+X)^{-1} e_k.\end{aligned}$$

By the maximum principle (Lemma 2.1), u is positive and we can write $E(X) = \|u\|_{\ell^1(V)} = \langle e, u\rangle$. Thus, the gradient is

$$\begin{aligned}\frac{\partial E}{\partial X_j} &= \left\langle e, \frac{\partial u}{\partial X_j} \right\rangle \\ &= -\langle (L+X)^{-T} e, e_j\rangle \langle u, e_j\rangle,\end{aligned}$$

or in other words

$$\nabla_X E = u \odot v$$

for u and v as in (2.3).

For the Hessian, we have

$$\begin{aligned}\frac{\partial^2 E}{\partial X_j \partial X_k} &= \left\langle e, \frac{\partial^2 u}{\partial X_j \partial X_k} \right\rangle \\ &= \left\langle e_k, (L+X)^{-1} e_j \right\rangle \langle u, e_j\rangle \langle e_k, v\rangle + \left\langle e_j, (L+X)^{-1} e_k \right\rangle \langle v, e_j\rangle \langle e_k, u\rangle.\end{aligned}$$

Thus, the Hessian can be written

$$H = \nabla^2 E = (L+X)^{-1} \odot W + (L+X)^{-T} \odot W^T,$$

where

$$W := u \otimes v,$$

as claimed. ∎

Remark 2.3. If L is symmetric, the above statements can be simplified greatly to give

$$H = \nabla^2 E = (L+X)^{-1} \odot (W + W^T),$$

where

$$W + W^T := u \otimes v + v \otimes u = \frac{1}{2}(u+v) \otimes (u+v) - \frac{1}{2}(u-v) \otimes (u-v).$$

PROPOSITION 2.4. *Fix some $X_\infty > 0$. For $f > 0$ fixed, let u satisfy $(L+X)u = f$. The mapping $X \mapsto E(X) = \|u\|_{\ell^1(V)}$ is strongly convex on $\{X_\infty \geq X \geq 0,\ X \neq 0\}$.*

Proof. We wish to show that

$$E(X) = e^T (L+X)^{-1} d$$

is convex on $[0, X_\infty]^n$ (excluding the origin) for fixed constant X_∞. Defining $\tilde{X} = D + X$ and using the identity $L = D - A$, this is equivalent to

$$e^T(\tilde{X} - A)^{-1}d$$

being convex on $\{V : d_i + X_\infty \geq \tilde{X}_i \geq d_i\}$. Expanding, we have

$$e^T\left(I - \tilde{X}^{-1}A\right)^{-1}\tilde{X}^{-1}d = e^T\sum_{k=0}^{\infty}\left(\tilde{X}^{-1}A\right)^k\tilde{X}^{-1}d.$$

Since the sum of convex functions is convex, it is enough to show that

$$e^T\left(\tilde{X}^{-1}A\right)^k\tilde{X}^{-1}d$$

is convex for each $k > 0$. For fixed k, and using the fact that e, d, and A are nonnegative, the preceding term is a nonnegative linear combination of terms of the form

$$F(x) = \prod_i x_i^{-\alpha_i}.$$

Therefore, we seek to show that F is convex when all entries are positive for any $\alpha = (\alpha_1, \ldots, \alpha_n)$ with each $\alpha_j \geq 0$ and at least one $\alpha_j > 0$ for $1 \leq j \leq n$.[2] To prove this, we check whether the Hessian is strictly positive definite. Computing second derivatives gives

$$\frac{\partial^2 F}{\partial X_i \partial X_i}(X) = F(X)\alpha_i(\alpha_i + 1)X_i^{-2}$$

and

$$\frac{\partial^2 F}{\partial X_i \partial X_j}(X) = F(X)\alpha_i\alpha_j X_i^{-1}X_j^{-1}.$$

So the Hessian of F is

$$F(X)\left[(\alpha X^{-1})(\alpha X^{-1})^T + \operatorname{diag}(\alpha X^{-2})\right],$$

which is positive semidefinite, being the sum of positive semidefinite matrices.[3]

To prove positive definiteness (rather than semidefiniteness), recognize that the $k = 0$ term contributes a term to the Hessian of the form DX^{-2}, which is strictly positive definite on the domain in question. This proves strong convexity. ∎

Proposition 2.4 gives that $\phi \to E_\varepsilon(\phi)$ is strongly convex on $\mathbb{R}_+^{|V|}$, so $\{\phi_\ell\}_{\ell\in[K]} \mapsto \tilde{E}_\varepsilon\left(\{\phi_\ell\}_{\ell\in[K]}\right)$ is also convex on $\mathcal{A}_K$. The following corollary is then immediate.

COROLLARY 2.5 (Bang-bang solutions). *Every maximizer of* (1.7) *is an extreme point of* $\{\phi \in [0,1]^{|V|} : \langle\phi, 1\rangle = k\}$*, i.e., an indicator function for some vertex set* $S \subset V$ *with* $|S| = k$.

Thus, in the language of control theory, Corollary 2.5 shows that (1.7) is a bang-bang relaxation of (1.3) and that (1.11) is a bang-bang relaxation of (1.8).

[2]Note, $f(x, y) = xy$ is not convex, but $x^{-1}y^{-1}$ is convex for $x, y > 0$.

[3]The first term is positive semidefinite (PSD) since it is the outer product of a nonnegative vector with itself, and the second term is PSD since it is diagonal with positive entries.

COROLLARY 2.6. *Since the set of values* $(x_1, \ldots, x_n) \in \mathbb{R}^n_+$ *with which we are concerned is convex and* E *is* C^2 *in* X*, the resulting Hessian matrix* H *is positive definite.*

Remark 2.7. Note that though the Hadamard product of two positive definite matrices is positive definite, Corollary 2.6 is not obvious from the structure of the Hessian, given that the matrix W is indefinite when u and v are linearly independent. As a result, this positive definiteness is strongly related to the structure of the $L + X$ matrix and its eigenvectors.

2.2. Optimization scheme.

2.2.1. Subgraph detector. We solve (1.7) using rearrangement ideas as follows. After initializing S (randomly in our experiments), we use the gradient (2.2) to find the locally optimal next choice of S, and then iterate until convergence (typically < 10 iterations in our experiments). More explicitly, we follow these steps:

$$Lu + \varepsilon^{-1}(1 - \chi_{S^0})u = d, \tag{2.5}$$

$$L^T v + \varepsilon^{-1}(1 - \chi_{S^0})v = 1. \tag{2.6}$$

The update, S^1, then contains those nodes ℓ that maximize $u_\ell v_\ell$.

Pseudocode for this approach is given in Algorithm 1, which has the following ascent guarantee.

Algorithm 1 Subgraph detector.

Input $S^0 \subset V$.
while $S^t \neq S^{t-1}$ **do**
Solve (2.5) and (2.6) for u and v.
Assign vertex ℓ to subgraph S^1 if $\nabla_\phi E$ is optimized. That is, solve the following subproblem:

$$\max_{|S|=k} \sum_{\ell \in S} u(\ell) \cdot v(\ell). \tag{2.7}$$

(Note that (2.7) is easily solved by taking the k indices corresponding to the largest values of $u(\ell) \cdot v(\ell)$, breaking ties randomly if needed.)
Reset now, building on $S^1 \subset V$ accordingly and repeat until $S^n = S^{n-1}$.
end while

PROPOSITION 2.8. *Every nonstationary iteration of Algorithm* 1 *strictly increases the energy* E_ε*. Algorithm* 1 *terminates in a finite number of iterations.*

Proof. Let S^0 and S^1 be the vertex subsets for successive iterations of the method. Define $W_{1,0} = \chi_{S^1} - \chi_{S^0}$. Assuming $W_{1,0} \neq 0$, by strong convexity (Theorem 2.4) and the formula for the gradient (2.2), we compute

$$E_\varepsilon(\chi_{S^1}) > E_\varepsilon(\chi_{S^0}) + \frac{1}{\varepsilon}\langle W_{1,0}, uv\rangle \tag{2.8a}$$

$$= E_\varepsilon(\chi_{S^0}) + \frac{1}{\varepsilon}\left(\sum_{i \in S_1} u_i v_i - \sum_{i \in S_0} u_i v_i\right) \tag{2.8b}$$

$$\geq E_\varepsilon(\chi_{S^0}). \tag{2.8c}$$

Thus, the energy is strictly increasing on nonstationary iterates. Since we assume that V is a finite size vertex set and the rearrangement method increases the energy, it cannot cycle and hence must terminate in a finite number of iterations. ∎

To avoid hand-selection of ε, we always set $\varepsilon = C/\|L\|_F$, where $\|L\|_F$ is the Frobenius norm of the graph Laplacian and $C > 1$ is typically set at $C = 50$ to make sure ε allows communication between graph vertices. If C is chosen to take a different value below, we will highlight those cases. Theoretically, this choice of ε is appropriate so that in $L + \frac{1}{\varepsilon}(1-\phi)$ the perturbation is on roughly the same scale as the Laplacian itself and is thus neither negligible nor dominating. In Figure 10, we illustrate that in empirical cases the partitioner performance is fairly stable with respect to ε in this range.

2.2.2. Graph partitioner. Given the success of the energy (1.6), one might naïvely consider partitioning the graph by maximizing an energy of the form

$$(S_1, S_2, \ldots, S_K) \mapsto \sum_{i=1}^{K} [E_\varepsilon(\chi_{S_i})]. \tag{2.9}$$

It can be seen that this energy does not properly constrain the volumes of each partition component and the solution of this problem merely puts all the vertices in a single component. Indeed, when $\varepsilon \to 0$, this is the sum of mean escape times of the partition components, which is maximized when one partition component contains the entire graph (since the escape time is then infinite). Thus, the above formulation is inadequate without a volume constraint.

As a second attempt, we considered a partition energy of the form

$$(S_1, S_2, \ldots, S_K) \mapsto \sum_{i=1}^{K} [|V| E_\varepsilon(\chi_{S_i})]^{-1}, \tag{2.10}$$

since the inverses penalize putting all nodes into the same partition by making the resulting empty classes highly costly. Intuitively, this energy functional provides an effective volume normalization of the relative gradients (similar to a K-means type scheme). However, while in practice this functional appears to work reasonably well on all graph models considered here, we were unable to prove, upon analysis of the Hessian, that rearrangements based on such an algorithm are bang-bang like the subgraph detector.

Finally, as an alternative, we consider the partition energy

$$\tilde{E}_{\delta,\varepsilon}(S_1, S_2, \ldots, S_K) = \sum_{i=1}^{K} [1 + \delta |V| E_\varepsilon(\chi_{S_i})]^{-1}. \tag{2.11}$$

Applied to functions, $\phi_j \colon V \to [0,1]$, instead of indicator functions, we consider

$$\tilde{E}_{\delta,\varepsilon}(\phi_1, \phi_2, \ldots, \phi_K) = \sum_{i=1}^{K} [1 + \delta |V| E_\varepsilon(\phi_i)]^{-1}. \tag{2.12}$$

We then have that

$$\nabla_{\phi_j} \tilde{E} = -\frac{\delta}{[1 + \delta |V| E_\varepsilon(\phi_i)]^2} \nabla_{\phi_j} (|V| E_\varepsilon(\phi_j)) \tag{2.13}$$

making the Hessian consist of blocks of the form

$$\nabla^2_{\phi_j}\tilde{E} = -\frac{\delta}{[1+\delta|V|E_\varepsilon(\phi_i)]^2}\nabla^2_{\phi_j}(|V|E_\varepsilon(\phi_j)) + 2\frac{\delta^2}{[1+\delta|V|E_\varepsilon(\phi_i)]^3}(\nabla_{\phi_j}(|V|E_\varepsilon(\phi_j)))(\nabla_{\phi_j}(|V|E_\varepsilon(\phi_j)))^T. \tag{2.14}$$

Note, the first term is negative definite (by Proposition 2.4 and its proof) and the second term is $O(\delta^2)$. Therefore, for δ sufficiently small, this Hessian is negative definite, proving that $\tilde{E}$ is concave with respect to ϕ_j. In practice, we find that taking $\delta = \varepsilon$ is sufficient both for having a negative definite Hessian and generating good results with respect to our rearrangement scheme. As such, we will generically take $\delta = \varepsilon$ henceforward. Expansion on optimality of our choice of δ is a topic for future work, but a sweep over a range of δ values demonstrated that the outcomes were quite similar for δ of this scale with respect to ε. The number of iterations required to converge and thus the overall time of implementing the algorithm could differ for δ sufficiently small, but not the overall purity measure of the examples we considered.

Our approach to the node partitioner is largely analogous to that of the subgraph detector, with the exception that we use classwise ℓ^1 normalization when comparing which values of $u \odot v$ at each node. In detail, the algorithm is presented in Algorithm 2. It is a relatively straightforward exercise applying the gradient computation for $E_\varepsilon(S_i)$ from Proposition 2.4 to prove that the energy functional (2.10) will decrease with each iteration of our algorithm as in Proposition 2.8.

2.2.3. Semisupervised learning. In cases where we have a labeled set of nodes T with labels $\hat{\phi}_v \in \{0,1\}$ indicating whether we want node i to be in the subgraph ($\hat{\phi}_v = 1$) or its complement ($\hat{\phi}_v = 0$), we can incorporate this information into our approach as follows.

For the subgraph detector, we use $E_{\varepsilon,\lambda,T}(\phi) = E_\varepsilon(\phi) + \lambda\sum_{v\in T}(\phi_v - (1-\hat{\phi}_v))^2$. Then the rearrangement algorithm needs to be modified at step 3 of Algorithm 1 to become: Assign vertex ℓ to subgraph S^1 if $\nabla_\phi E$ is optimized:

Algorithm 2 Graph partitioner.

Input $\vec{S} = \{S_1^0, \dots, S_K^0\}$ a K partition of V.
while $\vec{S}^t \neq \vec{S}^{t-1}$ **do**
For $j = 1, \dots, K$, solve the equations

$$L\mathbf{u}_j + \varepsilon^{-1}(1-\chi_{S_j^0})\mathbf{u}_j = d,$$
$$L^T\mathbf{v}_j + \varepsilon^{-1}(1-\chi_{S_j^0})\mathbf{v}_j = 1.$$

Normalize $\mathbf{u}_j = \frac{\mathbf{u}_j}{(1+\varepsilon\|\mathbf{u}_j\|_{\ell^1})^2}$, $\mathbf{v}_j = \mathbf{v}_j$.
Assign vertex ν to $\vec{S}_j^{t+1}$ where

$$j = \text{argmax}\{\mathbf{u}_1 \cdot \mathbf{v}_1(\nu), \dots, \mathbf{u}_K \cdot \mathbf{v}_K(\nu),\}$$

(that is, optimize $\nabla_\phi E$) breaking ties randomly if needed.
Set $t = t+1$.
end while

$$\max_{|S|=k} \frac{1}{\varepsilon}\sum_{\ell\in S} u(\ell)\cdot v(\ell) + 2\lambda \sum_{v\in T}[\chi_S(v) - (1-\hat{\phi}_v)],$$

where χ is the binary-valued indicator function. This again is solved by picking the largest elements (we break ties by picking the lowest-index maximizers if needed). Since the energy is still convex, the energy still increases at each iteration.

For the K-partitioner, we have a labeled set of nodes T_i with labels $\hat{\phi}_{i,v} \in \{0,1\}$ for $i = 1,\ldots,K$ indicating whether we want node v to be in partition element i, with $\sum_i \hat{\phi}_{i,v} = 1$ for $v \in \cup_i T_i$. We can incorporate this information into our approach by modifying the energy to be the concave functional

$$\tilde{E}_{\varepsilon,\lambda}(\phi_1,\ldots,\phi_K) = \tilde{E}_{\varepsilon}(\phi_1,\ldots,\phi_K) - \lambda \sum_{v\in T}\sum_{j=1}^{K}(\phi_{j,v} - (1-\hat{\phi}_{j,v}))^2 \tag{2.15}$$

with the gradient rearrangement being appropriately modified.

3. Numerical results. We test the performance of these algorithms both on synthetic graphs and an assortment of "real-world" graphs. For the synthetic tests, we use a particular set of undirected stochastic block models which we call the MultIsCale K-block Escape Ensemble (MICKEE), designed to illustrate some of the data features which our algorithms handle. A MICKEE graph consists of N nodes partitioned into $K+1$ groups of sizes N_1, ..., N_K, and $N_{K+1} = N - \sum_{j=1}^{K} N_j$, where $N_1 < N_2 < \cdots < N_K < N_{K+1}$ (see the 2-MICKEE schematic in Figure 1). The nodes in the first K groups induce densely connected ER subgraphs (from which we will study escape times) while the last group forms a sparsely connected ER background graph. Each of the K dense subgraphs is sparsely connected to the larger background graph. The goal is to recover one of the planted subgraphs, generally the smallest. In our experiments, a naïve spectral approach often found one of the planted graphs, but we know of no way to control which subgraph is recovered. In particular, the spectral method is often drawn to larger-scale clusters when multiple scales are present. Our subgraph detector method, in contrast, can be directed to look at the correct scale to recover a specific subgraph, as we will demonstrate in the 2-MICKEE example (i.e., with two planted subgraphs).

(A) Schematic of typical 2-MICKEE graph (B) Corresponding spy plot

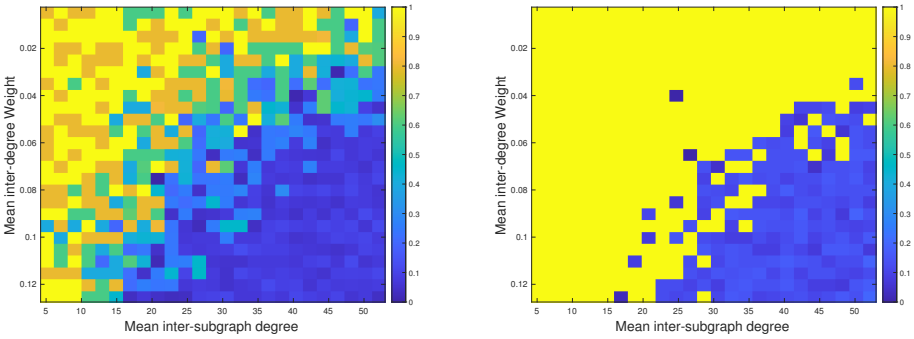
FIG. 1. *Schematic of a 2-MICKEE graph, with three dense subgraphs that are randomly connected to each other. Our subgraph detectors can identify the target subgraph, ignoring other planted subgraphs at different scales. Our partitioner correctly identifies each subgraph as a partition element, regardless of the scale.*

We explore a number of variations on the basic MICKEE theme, including (1) making the large subgraph have a power law degree distribution (with edges drawn using a loopy, multiedged configuration model), (2) adding more planted subgraphs with sizes ranging across several scales, (3) adding uniformly random noise edges across the entire graph or specifically between subgraphs, and (4) varying the edge weights of the various types of connections. For brevity, we refer to a MICKEE graph with K planted subgraphs (not including the largest one) as a K-MICKEE graph.

3.1. Subgraph detection. We explore the performance of Algorithm 1 using four benchmarks, which emphasize (1) noise tolerance, (2) multiscale detection, (3) robustness to heavy-tailed degree distributions, and (4) effective use of directed edges, respectively. In each of these tests, the target subgraph is the smallest planted subgraph.

Robustness to noise. In Figure 2 we visualize results from Algorithm 1 on 3-MICKEE graphs, varying the amount and type of noise. While it is possible to get a bad initialization and thus find a bad local optimum the subgraph detector usually finds the target exactly, except in the noisiest regime (which occurs roughly at the point where the number of noise edges is equal to the number of signal edges).

Range of scales. We generated 2-MICKEE graphs with varying sizes of the subgraphs relative to each other and the total mass. We take $1500 < N < 2500$ for the total size and vary the percentage of smallest planted subgraph as $.02N \leq N_1 \leq .15N$ with $N_2 = 2N_1$. Here, the intergroup edge density was set to .01 (in-subgraph-degree values between $0.01N(1-3p)$ for $.02 < p < .15$) with mean intergroup edge weight .05 compared to intragroup edge weights of 1. We used this framework to assess the detectability limits of sizes of the smallest components, and numerically we observe that small communities are quite detectable using our algorithm. Using the best result over five initializations, we were able to detect the smallest community over the entire



(a) Average of 5 runs.

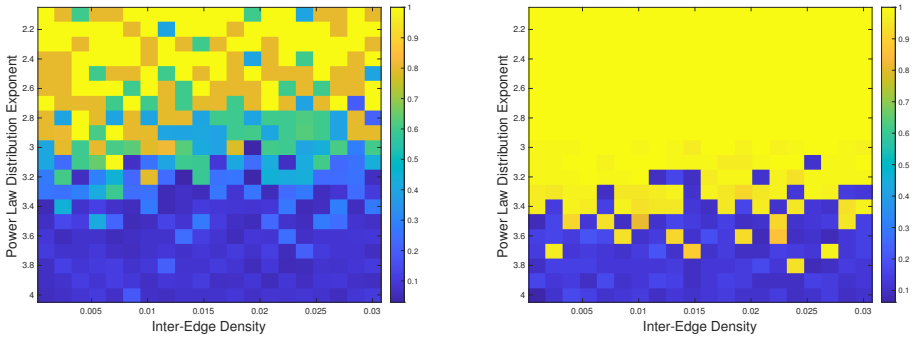
(b) Best of 5 runs.

FIG. 2. *Accuracy of Algorithm* 1 *as a function of mean intersubgraph degree (the mean taken over the nodes of the target subgraph) and mean weight of the intercomponent edges (not including nonedges) for* 3*-MICKEE graphs with planted subgraphs of sizes* 80, 160, *and* 240 *nodes, with a total of* 1,000 *nodes in the entire graph. The expected in-subgraph-degree is fixed at* 20.8 *(with intracomponent edge weights given by* 1*). (Therefore, if we define the mixing parameter as the typical proportion of a node's edges that are not in that node's community, the mixing parameter for the target subgraph ranges from* .19 *on the left to* .72*.) Intergroup edge weights are drawn from a normal distribution with maximum ranging from* .01–.25*. As long as the noise level is not too high, the subgraph detector finds the smallest planted subgraph despite the presence of "decoy" subgraphs at larger scales. This may be contrasted with spectral clustering, which is attracted to the larger scales.*

range and we did so reliably on average as well. Since the resulting figure would thus not be terribly informative for this range, we forego including a similar heat plot over this range of parameters. These results are not surprising in light of known results about the detectability limit in the related setting of small dense (unweighted) subgraphs in ER graphs. In particular, as elucidated in [67], which synthesizes prior results in the literature, when the random background graph is of density ρ and the dense subgraph is of density ρ_r, spectral methods are unable to detect dense subgraphs with fewer than $\sqrt{N\rho(1-\rho)}/(\rho_r-\rho)$ nodes. We thus expect that the small subgraphs in our MICKEE graph numerical examples under the parameters studied here are above the detectability threshold, because by comparison to the (seemingly slightly more difficult) case of planting a dense subgraph inside an ER graph the different edge weights we use here would effectively increase the $(\rho_r - \rho)$ denominator, decreasing the size of the smallest detectable dense subgraph. Nevertheless, while the smaller subgraphs are theoretically detectable (to both our method and to the naïve spectral approach), as noted above the spectral method is often drawn to larger-scale clusters when multiple scales are present.

Heavy-tailed degree distributions. For the results in Figure 3, we use a power law degree distribution in the largest component of 3-MICKEE graphs with $N_1 = 80, N_2 = 160, N_3 = 240$, and $N = 1000$. Surprisingly (at least to us), smaller power law exponents (corresponding to more skewed degree distributions) actually make the problem much easier (whereas adding noise edges had little effect). We conjecture that this is because, in the presence of very high-degree nodes, it is difficult to have a randomly occurring subgraph with high mean escape time, since connections into and out of the hubs are difficult to avoid.

Directed edge utilization. In Figure 4 we consider the problem of detecting a directed cycle appended to an ER graph. The graph weights have been arranged so that the expected degree of all nodes is roughly equal. There are many edges leading from the ER graph into the cycle, with only one edge leading back into the ER graph. This makes the directed cycle a very salient dynamical feature, but not readily detectable by undirected (e.g., spectral) methods. We considered a large number of cycle sizes relative to the ER graph and with a proper choice of ε, we were



(a) Average of 5 runs. (b) Best of 5 runs.

FIG. 3. *Accuracy of Algorithm* 1 *on a* 3*-MICKEE graph with a power law distribution as a function of the power law exponent and intercluster edge density. We observe a robustness to both the exponent and density (especially in the right panel) up to a sharp cutoff around* 3.4*. Note the low exponents (typically considered to be the harder cases) are actually easier in this problem.*

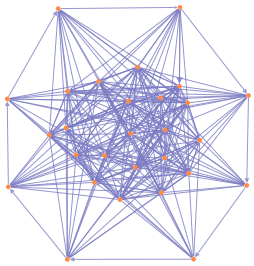
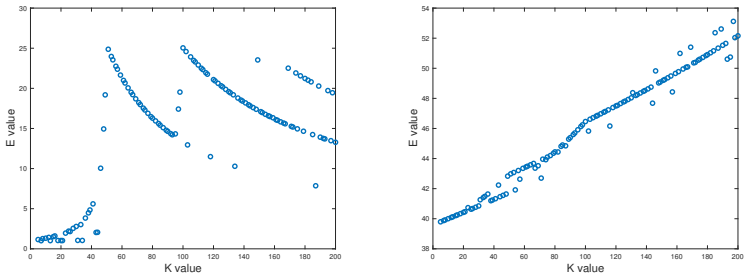


FIG. 4. *A directed ER graph with a directed cycle appended. Note that there is only one edge (in the upper left) leading from the cycle to the ER graph, with many edges going the other direction from the ER graph to the cycle. The cycle nodes have the same expected degree as the ER nodes, yet a random walker would naturally get stuck in the cycle for a long time. Detecting such a dynamical trap is a challenge for undirected algorithms, but Algorithm* 1 *detects it consistently over a wide range of cycle lengths and ER graph sizes.*



(a) True mean exit time. (b) Regularized energy.

FIG. 5. *The score of the optimal subgraph found with Algorithm* 1. *Both plots have clear shifts near* $k = 50$ *corresponding to the smallest component and* $k = 100$ *corresponding to the second smallest component. This suggests that the size of natural subgraphs within a given graph can be detected from breaks in the subgraph scores as the size of the target in Algorithm* 1 *varies.*

able to detect the cycle in all cases. Thus, this detector finds directed components very robustly due to the nature of the escape time.

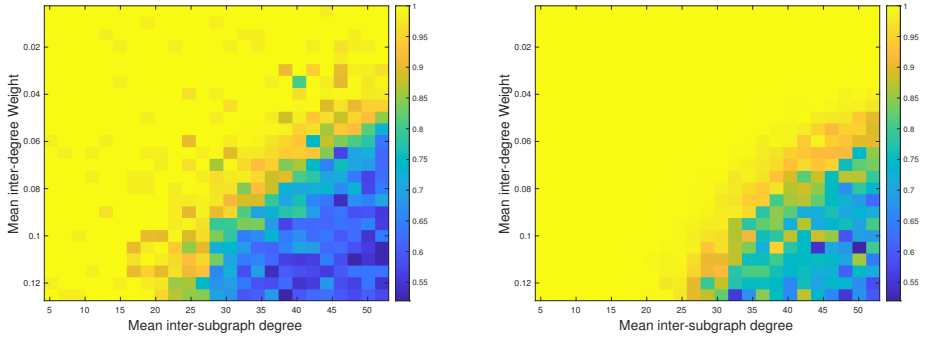
Variation over choice of N_1. In Figure 5, we consider how the Mean Exit Time as well as the regularized energy in (1.6) behaves as we vary the constrained volume of our algorithm. We considered a 2-MICKEE graph with $N_1 = 50$, $N_2 = 100$, and $N = 1000$. We took the baseline ER density .03 and the intergroup edge density was set to .025 with mean intergroup edge weight .1.

In summary, we find that the subgraph detector is able to robustly recover planted communities in synthetic graphs and is robust to a range of application-relevant factors.

3.2. K-partition method. We will now consider the performance of Algorithm 2 in a variety of settings. Throughout, we will give heat plots over the variation of the parameters to visualize the purity measure of our detected communities from our ground-truth smallest component of the graph, over five iterations of the algorithm. The purity measure is

$$\frac{1}{N}\sum_{k=1}^{K}\max_{1\le l\le K} N_k^l$$

for N_k^l the number of data samples in cluster k that are in ground truth class l.



(a) Average of 5 runs.

(b) Best of 5 runs.

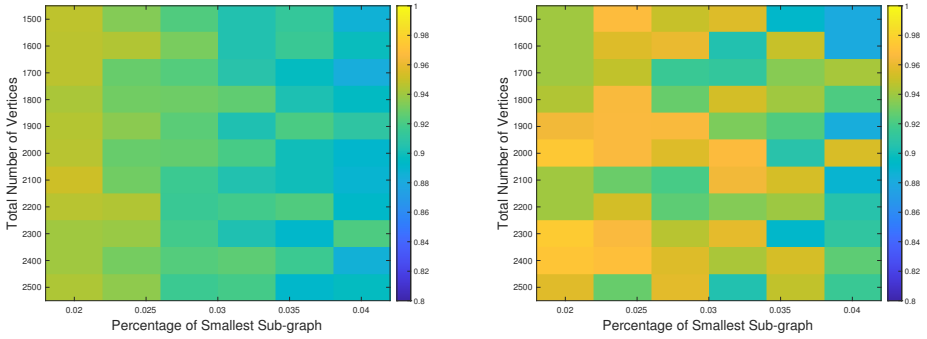
FIG. 6. *The purity measure for Algorithm* 2 *on* 3*-MICKEE graphs. We vary the density of the interregion edges and their edge weights. We observe robust (usually perfect) detection over a range of these parameters, with a sharp cutoff (especially in the left panel) when the noise levels grow too high, suggesting that detection is still possible beyond this cutoff, but the energy landscape has more bad local optima beyond this point. The left side of each figure corresponds to a mixing parameter of* .11*, and the right side corresponds to* .57*. The mixing parameter is obtained by mapping the x-axis through* $x \to x/(40+x)$.

In Figure 6 we consider a heat plot of the purity measure for a 4-partition of a 3-MICKEE graph using delocalized connections with $N_1 = 80, N_2 = 160, N_3 = 240$, and $N = 1000$, varying the density of the intercomponent edge connections ($0 < \rho < .1$) and the mean weight of the intercomponent edges ($0 < \Delta < .125$). We vary over number and strength of connecting edges between components and consider the purity measure as output. The base ER density was set to .04.

In addition, we have tested Algorithm 2 on MICKEE graphs with varying sizes of the components relative to each other and the total mass where the connections between ER graphs include more random edges with weak connection weights. Figure 7 shows results from testing the algorithm on 2-MICKEE graphs with varying sizes of the components relative to each other and the total mass. We take $1500 < N < 2500$ for the total size and vary the percentage of smallest planted subgraph as $.02N \le N_1 \le .15N$ with $N_2 = 2N_1$. Here, the intergroup edge density was set to .025 with mean intergroup edge weight .05. The question addressed in this experiment is how small can we get the components and still detect them? We heat map the average purity measure varying the number of vertices in the graph and the relative size of the smallest subgraph (i.e., N_1/N).

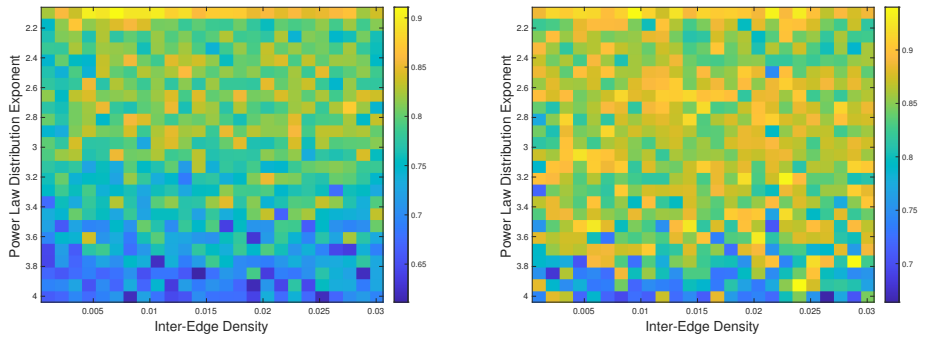
We similarly consider the partitioning problem on a version of the 3-MICKEE graph with power law degree distribution in the largest component, using delocalized connections with $N_1 = 80, N_2 = 160, N_3 = 240$, and $N = 1000$. Figure 8 provides a $\rho - q$ plot for results from varying the density ($.001 < \rho < .03$) of the edge-density of connections between the components of the graph, using a power law degree distribution for the largest component with exponent ($2.1 \le q \le 4$).

3.2.1. Graph clustering examples. We consider the family of examples as in [81] and compare the best presented purity measures from that paper to a number of settings using our algorithms. Since some of these examples are by their nature actually directed data sets, throughout we computed both the directed and undirected adjacency matrix representations as appropriate to test against. We ran the



(a) Average of 5 runs. (b) Best of 5 runs.

FIG. 7. *The purity measure for the partitioner acting on a 2-MICKEE graph with the fraction of nodes in the smaller planted subgraph varying, along with the size of the graph. We observe a generally robust partitioning.*



(a) Average of 5 runs. (b) Best of 5 runs.

FIG. 8. *Purity achieved by Algorithm 2 on 3-MICKEE graphs with a power law degree distribution, varying the exponent of the power law and inter-subgraph edge density. We observe generally robust partitioning (especially in the right panel).*

K-partitioner over a variety of scenarios for both cases. In all these runs, we chose the value of K to agree with the metadata (we avoid term "ground truth," as the node labels themselves may be noisy or not the only good interpretation of the data). However, we note that our algorithm also does a good job in a variety of settings selecting the number of partitions to fill without precisely providing this correct number a priori.

For our study, we consider a number of various options for the algorithm. First, the initial seeding sets were chosen either uniformly at random or using K-means on the first K-eigenvectors of the graph Laplacian. We consider the best result over 10 outcomes. In addition, we considered a range of values of ε, all of which were a multiplicative factor of the inverse of the Frobenius norm of the graph Laplacian, denoted $\|L\|_{\text{Fro}}$, which sets a natural scaling for separation in the underlying graph. See, for instance, the related choice in [57]. We computed a family of partitions for $\varepsilon = 50\nu/\|L\|_{\text{Fro}}$, where $\nu = e^{.2\ell}$ with $-50 < \ell < 50$. Finally, we also considered the impact of semisupervised learning by toggling between $\lambda = 0$ and $\lambda = 10^6$ in (2.15) with 10% of the nodes being included in the learning set. Clearly, there are many

ways we might improve the outcomes, by, for instance, increasing the number and method of initialization and refining our choices of ε or λ; nevertheless, we see under our current choices that our algorithm performs well over a range of such parameters, as reported in Table 1. In Figure 9, for these 45 datasets, we plot the graph edge count vs. the median time per iteration for our graph partitioning algorithm. The main cost per iteration is the preconditioned GMRES solves.

For each data set in Table 1, we report the best outcome using directed adjacency matrices to build the Graph Laplacian using both the K-means and random initializations but with no semisupervised learning (Directed); the best outcome using symmetrized adjacency matrices to build the Graph Laplacian using both the K-means and random initializations but with no semisupervised learning (Undirected); the best outcome when SemiSupervised Learning is turned on over any configuration (Semisupervision), the K-means only outcome (K-means only) and the best data from all the experiments reported in [81] (best from [81]). Our results promisingly demonstrate that our algorithm is very successful in many cases in discovering large amounts of community structure that aligns with the metadata in these explicit data sets. Given that our communities are all built around random walks in the graph, it is not clear that all ground-truth designated communities would align well with our methods. For example, we note that our results do not align well with the metadata in the POLBLOGS data set, but since this particular dataset is known to have multiple types of unaligned clustering structure present (for example, partisan split and core-periphery structure), the lack of alignment with one set of node metadata should not generally be understood as failure to capture "the" correct partition. A major takeaway from the table, however, is that in several examples we see that using the directed nature of the data provides better agreement with the metadata (as indicated by the green cells). Perhaps most striking in the table is that the best run of our algorithm, even without semisupervised learning, provides better agreement with the metadata than [81] for many of the data sets.

As a statistical summary of our findings, we had in total 39 directed datasets and 6 undirected data sets that came from a variety of domains (image, social, biological, physical, etc.). The networks are sized between 35 nodes and $98,528$ nodes, having 2–65 classes per network. Among directed networks, 21 data sets gave highest purity with the metadata with semisupervised learning turned on, while 13 have the best result from [81], and 2 have K-means only best. For 9 total data sets (green in the table), the directed version of our algorithm is more closely aligned with node metadata than the symmetrized undirected version, while 5 are tied (yellow) and for 25 the undirected method is more closely aligned (orange). When [81] is best, the median gap from our result with semisupervised learning is .05. When our algorithm with semisupervised learning is best, the median gap from [81] is .05. There is no clear relationship between data domain and performance or node count and performance. However, semisupervision generally did improve the results the most with a smaller class count (median 3) versus [81] (median 20).

When the directed algorithm aligns more closely with node metadata than the undirected version, the median gap is 0.03. Interestingly, five of the datasets where directed was more aligned with node metadata were images or sensor data, with the two largest gaps (.07 and .09) being digit datasets. When undirected was more closely aligned with node metadata, the median gap was 0.06, with the largest gap being .29, for the 20NEWS dataset. When semisupervision improves over our method (max of directed and undirected performance), the median improvement is .06, and the max

TABLE 1
Purity measure table. (Table in color online.)

Network	Domain	Vertices	Density	Classes	Directed	Undirected	Semisupervision	K-means only	Best from [81]
Directed data									
MNIST	Digit	70,000	0.00	10	0.85	0.78	**0.98**	0.84	0.97
VOWEL	Audio	990	0.01	11	0.35	0.32	**0.44**	0.34	0.37
FAULTS	Materials	1,941	0.00	7	0.44	0.42	**0.49**	0.39	0.41
SEISMIC	Sensor	98,528	0.00	3	0.60	0.59	**0.66**	0.58	0.59
7Sectors	Text	4,556	0.00	7	0.27	0.26	**0.39**	0.26	0.34
PROTEIN	Protein	17,766	0.00	3	0.47	0.46	**0.51**	0.46	0.50
KHAN	Gene	83	0.06	4	0.59	0.59	**0.61**	0.59	0.60
ROSETTA	Gene	300	0.02	5	0.78	0.78	**0.81**	0.77	0.77
WDBC	Medical	683	0.01	2	0.65	0.65	**0.70**	0.65	0.65
POLBLOGS	Social	1,224	0.01	2	0.55	0.55	**0.59**	0.51	NA
CITESEER	Citation	3,312	0.00	6	0.28	0.29	**0.49**	0.25	0.44
SPECT	Astronomy	267	0.02	3	0.79	0.80	**0.84**	0.79	0.79
DIABETES	Medical	768	0.01	2	0.65	0.67	**0.74**	0.65	0.65
DUKE	Medical	44	0.11	2	0.64	0.68	**0.73**	0.52	0.70
IRIS	Biology	150	0.03	3	0.87	0.90	**0.97**	0.67	0.93
RCV1	Text	9,625	0.00	4	0.35	0.40	**0.62**	0.32	0.54
CORA	Citation	2,708	0.00	7	0.33	0.39	**0.50**	0.32	0.47
CURETGREY	Image	5,612	0.00	61	0.23	0.29	**0.33**	0.22	0.28
SPAM	Email	4,601	0.00	2	0.64	0.70	**0.73**	0.61	0.69
GISETTE	Digit	7,000	0.00	2	0.87	0.94	**0.97**	0.81	0.94
WEBKB4	Text	4,196	0.00	4	0.42	0.53	**0.66**	0.40	0.63
CANCER	Medical	198	0.03	14	0.49	**0.55**	0.54	0.45	0.54
YALEB	Image	1,292	0.00	38	0.44	**0.54**	0.52	0.41	0.51
COIL-20	Image	1,440	0.00	20	0.74	**0.85**	0.78	0.82	0.81
ECOLI	Protein	327	0.02	5	0.79	**0.83**	0.81	0.81	**0.83**
YEAST	Biology	1,484	0.00	10	0.46	0.53	0.54	0.47	**0.55**
20NEWS	Text	19,938	0.00	20	0.20	0.49	0.62	0.16	**0.63**
MED	Text	1,033	0.00	31	0.50	0.54	0.54	0.48	**0.56**
REUTERS	Text	8,293	0.00	65	0.60	0.69	0.75	0.60	**0.77**
ALPHADIGS	Digit	1,404	0.00	6	0.42	0.48	0.48	0.46	**0.51**
ORL	Face	400	0.01	40	0.76	0.82	0.76	0.78	**0.83**
OPTDIGIT	Digit	5,620	0.00	10	0.90	0.93	0.91	0.90	**0.98**
PIE	Face	1,166	0.00	53	0.53	0.66	0.62	0.51	**0.74**
SEG	Image	2,310	0.00	7	0.54	0.64	0.59	0.51	**0.73**
UMIST	Face	575	0.01	20	0.74	0.71	0.67	0.67	**0.74**
PENDIGITS	Digit	10,992	0.00	10	0.82	0.73	0.82	0.83	**0.87**
SEMEION	Digit	1,593	0.00	10	0.86	0.82	0.77	0.81	**0.94**
AMLALL	Medical	38	0.13	2	0.92	**0.95**	0.94	**0.95**	0.92
IONOSPHERE	Radar	351	0.01	2	0.77	0.77	**0.85**	**0.85**	0.70
Undirected data									
POLBOOKS	Social	105	0.08	3	0.83	**0.85**	**0.85**	0.82	0.83
KOREA	Social	35	0.11	2	**1.00**	**1.00**	**1.00**	0.71	**1.00**
FOOTBALL	Sports	115	0.09	12	**0.94**	0.93	0.90	0.93	0.93
MIREX	Music	3,090	0.00	10	0.21	0.24	0.27	0.12	**0.43**
HIGHSCHOOL	Social	60	0.10	5	0.82	0.85	0.83	0.82	**0.95**

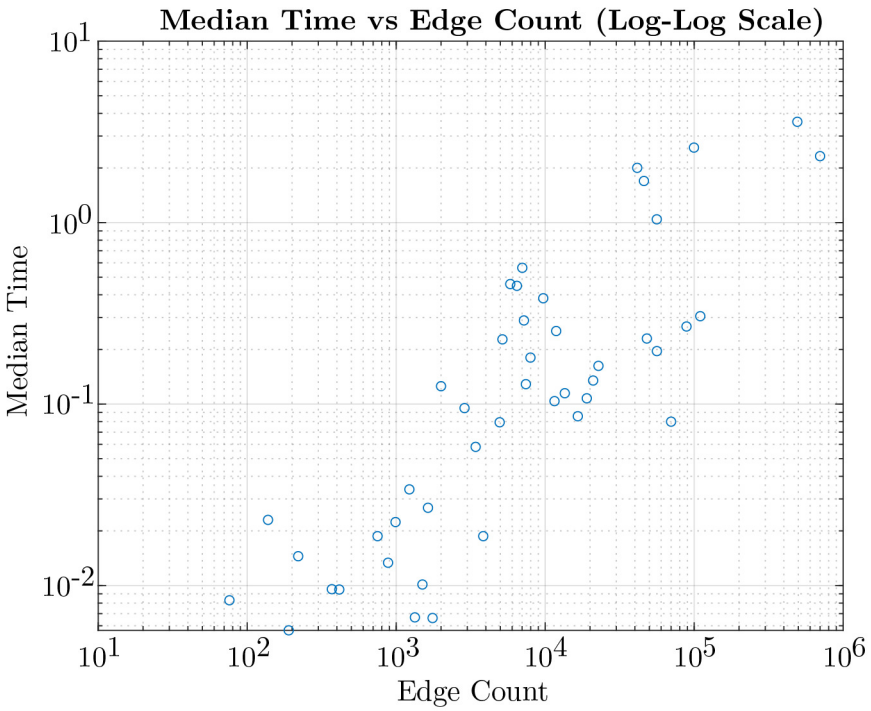
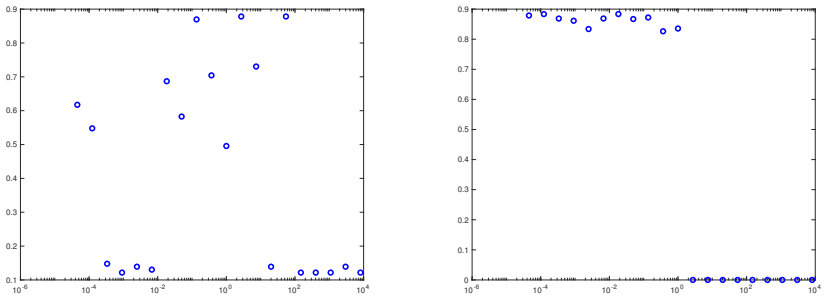


FIG. 9. *Timing for one iteration of our graph partitioner. The slope of roughly one suggests a roughly linear scaling law, which supports the argument that our approach is scalable to large datasets, as expected.*

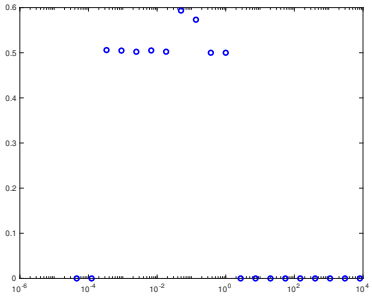
improvements were .22 and .20. There is no obvious relationship between edge density and algorithm performance.

We have discussed the output of a variety of experiments on a large number of data sets, but we also want to discuss their dependence upon the ε parameter and the percentage of nodes that are learned in the energy (2.15). To that end, we consider the output purity measure for some representative data sets and look at the outputs over a range of epsilon parameters and percentages of learning. In this case, we considered only the K-means initialization for consistency and simplicity of comparison. For the ε sweep, we recall that we considered the range $\varepsilon = 50\nu/\|L\|_{\text{Fro}}$, where $\nu = e^{\cdot 2\ell}$ with $-50 < \ell < 50$. In Figure 10 we show the variation in the purity measure with ε for a small graph (FOOTBALL), a medium sized graph (OPTDIGITS), and a large graph (SEISMIC). Similarly, in Figure 11 we visualize how results vary with the fraction of supervision (nodes with labels provided) under semisupervised learning, for the same graphs, with $\nu = .6, .8, 1.0, 1.2, 1.4, 1.6, 1.8$.

4. Discussion. Throughout our study we emphasize that our methodology operates fundamentally on the possibly directed nature of the underlying graph data. Considering the Index of Complex Networks [20] as a representative collection of widely-studied networks, we note that (as of our writing here) 327 of the 698 entries in the Index contain directed data. Whereas there are undoubtedly settings where one can ignore edge direction, there are inevitably others where respecting direction is essential. By formulating a strategy for subgraph detection and graph partitioning



(a) Football (b) Optdigits



(c) Seismic

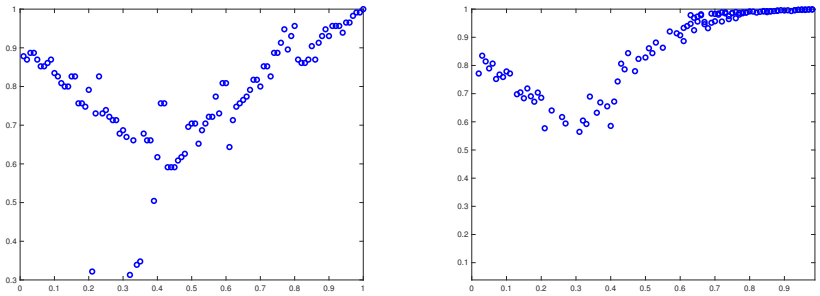
FIG. 10. *Purity measures for three selected data sets as a function of the scale parameter* $\nu = \frac{||L||_{\text{Fro}}}{50}\varepsilon$. *In all three panels, we observe a stable range (on a log scale) where purity is stably nontrivial, and in the left panel, there are two such scales.*

inherently built on processes running on the directed graph, we avoid the need for any post hoc modifications to try to respect directed edges. In particular, our method nowhere relies on any correspondingly undirected version of the graph, avoiding possible information lost in symmetrizing.

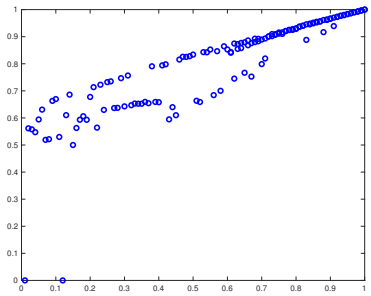
While we expect that our formulation of escape times can be useful in general, including for undirected graphs, our proper treatment of the directed graph data should prove especially useful. For example, the directed follower v. following nature of some online social networks (e.g., Twitter) is undoubtedly important for understanding the processes involved in the viral spread of (mis)information. As shown by [77] (and extended by [43]), the community structure is particularly important for identifying the virality of memes specifically because a meme that "escapes" (in our present language) its subgraph of origin is typically more likely to continue to propagate. Another application where directed escape times could be relevant is in detecting the (hidden) circulation of information, currency, and resources that is part of coordinated adversarial activity, as explored, for example, in [37, 50, 66].

To close, we highlight two related thematic areas for possible future work that we believe would lead to important extensions on the methods presented here.

4.1. Connection to distances on directed graphs. In previous work of the present authors [12] along with Weare, we construct a symmetrized distance function on the vertices of a directed graph. We recall the details briefly here, which is based



(a) Football (b) Optdigits



(c) Seismic

FIG. 11. *Purity measures on three selected data sets as a function of the fraction of supervision (nodes with labels provided) under semisupervised learning. We observe that supervision can either consistently help (as in the right panel) or can have inconsistent effects (as in the left and middle panels). One possible explanation for this is that there may be multiple clustering structures present in the data, and it takes a lot of supervision to force the partitioner to switch to a partition aligned with the metadata indicated by the supervision, rather than a different clustering structure that is better from the perspective of the optimizer.*

somewhat upon the hitting probability matrix construction used in umbrella sampling [24, 69]. For a general probability transition matrix P, we denote the Perron eigenvector as $P'\phi = \phi$. Let us define a matrix M such that $M_{ij} = \text{Prob}_i[\tau_j < \tau_i]$, the probability that for a random walker starting from site i the hitting time of j is less than the time it takes to return to i. Then, it can be proved that (see [24, 69]) $M * \text{diag}(\phi)$ (with appropriate scaling) gives a natural symmetrization of and a metric on the nodes of a (possibly directed, strongly connected) graph. Here, "natural" means that it encodes the structure of arbitrarily long cycles in a balanced way that is useful for detecting various dynamics-related structures, as in the ER + cycle example; Figure 4. A natural question to pursue is whether parsing the directed network with this approach to create the symmetrized A^{hp} matrix, then applying our clustering scheme can be used to effectively detect graph structures in a more robust manner. In particular, comparison of our clustering scheme versus K-means studies of the distance structure should be an important direction for future study.

4.2. Continuum limits. The methods presented here have a clear analog in the continuum setting to the motivated problems in the continuum discussed in the introduction. The primary continuum problem is related to the landscape function, or torsion function, on a subdomain prescribed with Dirichlet boundary conditions,

$$-\Delta u_S = 1_S, \quad u_S|_{\partial S} = 0. \tag{4.1}$$

This is known as the mean exit time from a set S of a standard Brownian motion random walker; see [58, Chapter 7]. Correspondingly, for a domain Ω with Neumann boundary conditions (to make life easier with graphs) and some $0 < \alpha < 1$, we propose the following optimization:

$$\max_{S\subset\Omega, |S|=\alpha|\Omega|} \int_S u_S \, dx\,, \tag{4.2}$$

meaning that we wish to maximize the exit time of a random walker from a given subdomain. Through the Poisson formula for the mean exit time, we have that $\int u_S = (-\Delta u_S, u_S)$, allowing us to frame things similarly via a Ginzburg–Landau like penalty term for being in a set S,

$$\min_{\substack{0\le\phi\le 1\\ \int\phi=\alpha|\Omega|}} \min_{\int u=1} \frac{1}{2}(-\Delta u_S, u_S) + \frac{1}{2\varepsilon}\langle u, (1-\phi)u\rangle.$$

Analysis of optimizers for such a continuum problem and its use in finding subdomains and domain partitions is one important direction for future study. Related results in a continuum setting have been studied, for instance, in [14, 17], but the regularization of this problem seems to be new and connects the problem through the inverse of the Laplacian to the full domain and its boundary conditions. Following works such as [56, 65, 71, 72, 73, 82], an interesting future direction would be to prove consistency of our algorithm to these well-posed continuum optimization problems.

Acknowledgments. The authors are grateful to Matthias Kurzke and Jonathan Weare for helpful discussions throughout the crafting of this result. The authors also thank the three anonymous reviewers for many helpful suggestions for improving the exposition of the manuscript.

REFERENCES

[1] R. Andersen and K. Chellapilla, *Finding dense subgraphs with size bounds*, in Proceedings of the International Workshop on Algorithms and Models for the Web-graph, Lecture Notes in Comput. Sci. 5427, Springer, Boston, 2009, pp. 25–37.

[2] R. Andersen, F. Chung, and K. Lang, *Local graph partitioning using PageRank vectors*, in Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science, IEEE, 2006, pp. 475–486.

[3] R. Andersen, F. Chung, and K. Lang, *Local partitioning for directed graphs using PageRank*, in Algorithms and Models for the Web-Graph, Lecture Notes in Comput. Sci. 4863, A. Bonato and F. R. K. Chung, eds., Springer, Berlin, 2007, pp. 166–178.

[4] S. Arora, E. Hazan, and S. Kale, *$O(\sqrt{\log n})$ approximation to SPARSEST CUT in $\tilde{O}(n^2)$ time*, SIAM J. Comput., 39 (2010), pp. 1748–1771, https://doi.org/10.1137/080731049.

[5] S. Arora, S. Rao, and U. Vazirani, *Expander flows and a $\sqrt{\log n}$-approximation to SPARSEST CUT*, in Proceedings of the 36th ACM Symposiumon Theory of Computing, Vol. 25, 2004, 29.

[6] S. Arora, S. Rao, and U. Vazirani, *Geometry, flows, and graph-partitioning algorithms*, Commun. ACM, 51 (2008), pp. 96–105.

[7] S. Arora, S. Rao, and U. Vazirani, *Expander flows, geometric embeddings and graph partitioning*, J. ACM, 56 (2009), 5.

[8] D. S. Bassett, N. F. Wymbs, M. A. Porter, P. J. Mucha, J. M. Carlson, and S. T. Grafton, *Dynamic reconfiguration of human brain networks during learning*, Proc. Natl. Acad. Sci. USA, 108 (2011), pp. 7641–7646, https://doi.org/10.1073/pnas.1018985108.

[9] A. R. Benson, D. F. Gleich, and J. Leskovec, *Higher-order organization of complex networks*, Science, 353 (2016), pp. 163–166.

[10] A. Bhaskara, M. Charikar, E. Chlamtac, U. Feige, and A. Vijayaraghavan, *Detecting high log-densities: An $O(n^{1/4})$ approximation for densest k-subgraph*, in Proceedings of the 42nd ACM Symposium on Theory of Computing, 2010, pp. 201–210.

[11] E. G. Boman, K. D. Devine, and S. Rajamanickam, *Scalable matrix computations on large scale-free graphs using 2D graph partitioning*, in Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, 2013, 50, https://doi.org/10.1145/2503210.2503293.

[12] Z. M. Boyd, N. Fraiman, J. Marzuola, P. J. Mucha, B. Osting, and J. Weare, *A metric on directed graphs and Markov chains based on hitting probabilities*, SIAM J. Math. Data Sci., 3 (2021), pp. 467–493, https://doi.org/10.1137/20M1348315.

[13] X. Bresson, T. Laurent, D. Uminsky, and J. H. von Brecht, *An Adaptive Total Variation Algorithm for Computing the Balanced Cut of a Graph*, preprint, https://arxiv.org/abs/1302.2717, 2013.

[14] T. Briancon, *Regularity of optimal shapes for the Dirichlet's energy with volume constraint*, ESAIM Control Optim. Calc. Var., 10 (2004), pp. 99–122.

[15] J. Budd and Y. V. Gennip, *Graph Merriman–Bence–Osher as a semidiscrete implicit Euler scheme for graph Allen–Cahn flow*, SIAM J. Math. Anal., 52 (2020), pp. 4101–4139, https://doi.org/10.1137/19M1277394.

[16] A. Buluç and K. Madduri, *Graph partitioning for scalable distributed graph computations*, Contemp. Math., 588 (2013), pp. 83–102.

[17] G. Buttazzo and G. Dal Maso, *An existence result for a class of shape optimization problems*, Arch. Ration. Mech. Anal., 122 (1993), pp. 183–195.

[18] A. Chambolle and M. Novaga, *Convergence of an algorithm for the anisotropic and crystalline mean curvature flow*, SIAM J. Math. Anal., 37 (2006), pp. 1978–1987, https://doi.org/10.1137/050629641.

[19] S. Chu and J. Cheng, *Triangle listing in massive networks and its applications*, in Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2011, pp. 672–680.

[20] A. Clauset, E. Tucker, and M. Sainz, *The Colorado Index of Complex Networks*, available online at https://icon.colorado.edu/, 2016.

[21] S. Comandur and D. F. Gleich, *Vertex neighborhoods low conductance cuts and good seeds for local community methods*, in Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2012, pp. 597–605.

[22] P. J. Courtois, *On time and space decomposition of complex structures*, Commun. ACM, 28 (1985), pp. 590–603.

[23] A. Decelle, F. Krzakala, C. Moore, and L. Zdeborová, *Inference and phase transitions in the detection of modules in sparse networks*, Phys. Rev. Lett., 107 (2011), 065701.

[24] A. R. Dinner, E. H. Thiede, B. V. Koten, and J. Weare, *Stratification as a general variance reduction method for Markov chain Monte Carlo*, SIAM/ASA J. Uncertain. Quantif., 8 (2020), pp. 1139–1188, https://doi.org/10.1137/18M122964X.

[25] U. Feige, D. Peleg, and G. Kortsarz, *The dense k-subgraph problem*, Algorithmica, 29 (2001), pp. 410–421.

[26] M. Fiedler, *Algebraic connectivity of graphs*, Czechoslovak Math. J., 23 (1973), pp. 298–305.

[27] S. Fortunato, *Community detection in graphs*, Phys. Rep., 486 (2010), pp. 75–174, https://doi.org/10.1016/j.physrep.2009.11.002.

[28] S. Fortunato and D. Hric, *Community detection in networks: A user guide*, Phys. Rep., 659 (2016), pp. 1–44, https://doi.org/10.1016/j.physrep.2016.09.002.

[29] S. Fortunato and M. E. J. Newman, 20 *years of network community detection*, Nat. Phys., 18 (2022), pp. 848–850, https://doi.org/10.1038/s41567-022-01716-7.

[30] K. Fountoulakis, D. Wang, and S. Yang, *p-norm flow diffusion for local graph clustering*, in Proceedings of the 37th International Conference on Machine Learning, PMLR, 2020, pp. 3222–3232.

[31] E. Fratkin, B. T. Naughton, D. L. Brutlag, and S. Batzoglou, *MotifCut: Regulatory motifs finding with maximum density subgraphs*, Bioinformatics, 22 (2006), pp. e150–e157.

[32] D. F. Gleich, *PageRank beyond the web*, SIAM Rev., 57 (2015), pp. 321–363, https://doi.org/10.1137/140976649.

[33] D. F. Gleich and C. Seshadhri, *Vertex neighborhoods, low conductance cuts, and good seeds for local community methods*, in Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2012, pp. 597–605.

[34] W. Ha, K. Fountoulakis, and M. W. Mahoney, *Statistical guarantees for local graph clustering*, J. Mach. Learn. Res., 22 (2021), pp. 6538–6591.

[35] K. Ishii, *Optimal rate of convergence of the Bence–Merriman–Osher algorithm for motion by mean curvature*, SIAM J. Math. Anal., 37 (2005), pp. 841–866, https://doi.org/10.1137/04061862X.
[36] M. Jacobs, E. Merkurjev, and S. Esedoḡlu, *Auction dynamics: A volume constrained MBO scheme*, J. Comput. Phys., 354 (2018), pp. 288–310.
[37] H. Jin, X. He, Y. Wang, H. Li, and A. L. Bertozzi, *Noisy subgraph isomorphisms on multiplex networks*, in Proceedings of the 2019 IEEE International Conference on Big Data pp. 4899–4905, https://doi.org/10.1109/BigData47090.2019.9005645.
[38] R. Khandekar, S. Rao, and U. Vazirani, *Graph partitioning using single commodity flows*, J. ACM, 56 (2009), 19.
[39] R. Lambiotte, J.-C. Delvenne, and M. Barahona, *Random walks, Markov processes and the multiscale modular organization of complex networks*, IEEE Trans. Network Sci. Eng., 1 (2014), pp. 76–90, https://doi.org/10.1109/TNSE.2015.2391998.
[40] L. Le Gorrec, S. Mouysset, and D. Ruiz, *Doubly stochastic scaling unifies community detection*, Neurocomputing, 504 (2022), pp. 141–162.
[41] E. A. Leicht and M. E. J. Newman, *Community structure in directed networks*, Phys. Rev. Lett., 100 (2008), 118703, https://doi.org/10.1103/PhysRevLett.100.118703.
[42] T. Leighton and S. Rao, *An Approximate Max-flow min-cut Theorem for Uniform Multicommodity Flow Problems with Applications to Approximation Algorithms*, Technical report, MIT Cambridge Microsystems Research Center, Cambridge, MA, 1989.
[43] W. Li, S. J. Cranmer, Z. Zheng, and P. J. Mucha, *Infectivity enhances prediction of viral cascades in twitter*, PLOS One, 14 (2019), e0214453.
[44] C. Ma, Y. Fang, R. Cheng, L. V. S. Lakshmanan, W. Zhang, and X. Lin, *Efficient algorithms for densest subgraph discovery on large directed graphs*, in Proceedings of the 2020 ACM SIGMOD International Conference on Management Data, 2020, pp. 1051–1066.
[45] F. D. Malliaros and M. Vazirgiannis, *Clustering and community detection in directed networks: A survey*, Phys. Rep., 533 (2013), pp. 95–142.
[46] P. Manurangsi, *Almost-polynomial ratio ETH-hardness of approximating densest k-subgraph*, in Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, 2017, pp. 954–961.
[47] B. Merriman, J. K. Bence, and S. Osher, *Diffusion generated motion by mean curvature*, AMS Selected Letters, Crystal Grower's Workshop, 1993, pp. 73–83.
[48] D. Meunier, R. Lambiotte, A. Fornito, K. D. Ersche, and E. T. Bullmore, *Hierarchical modularity in human brain functional networks*, Front. Neuroinform., 3 (2009), 37, https://doi.org/10.3389/neuro.11.037.2009.
[49] J. Moody, *Peer influence groups: Identifying dense clusters in large networks*, Soc. Networks, 23 (2001), pp. 261–283, https://doi.org/10.1016/S0378-8733(01)00042-9.
[50] J. D. Moorman, Q. Chen, T. K. Tu, Z. M. Boyd, and A. L. Bertozzi, *Filtering methods for subgraph matching on multiplex networks*, in 2018 IEEE International Conference on Big Data, 2018, pp. 3980–3985, https://doi.org/10.1109/BigData.2018.8622566.
[51] T. Mouw and A. M. Verdery, *Network sampling with memory: A proposal for more efficient sampling from social networks*, Sociol. Methodol., 42 (2012), pp. 206–256, https://doi.org/10.1177/0081175012461248.
[52] P. J. Mucha, T. Richardson, K. Macon, M. A. Porter, and J.-P. Onnela, *Community structure in time-dependent, multiscale, and multiplex networks*, Science, 328 (2010), pp. 876–878, https://doi.org/10.1126/science.1184819.
[53] R. R. Nadakuditi, *On hard limits of eigen-analysis based planted clique detection*, in 2012 IEEE Statistical Signal Processing Workshop (SSP), IEEE, 2012, pp. 129–132.
[54] R. R. Nadakuditi and M. E. Newman, *Graph spectra and the detectability of community structure in networks*, Phys. Rev. Lett., 108 (2012), 188701.
[55] M. E. J. Newman and M. Girvan, *Finding and evaluating community structure in networks*, Phys. Rev. E, 69 (2004), 026113, https://doi.org/10.1103/physreve.69.026113.
[56] B. Osting and T. H. Reeb, *Consistency of Dirichlet partitions*, SIAM J. Math. Anal., 49 (2017), pp. 4251–4274, https://doi.org/10.1137/16M1098309.
[57] B. Osting, C. D. White, and É. Oudet, *Minimal Dirichlet energy partitions for graphs*, SIAM J. Sci. Comput., 36 (2014), pp. A1635–A1651, https://doi.org/10.1137/130934568.
[58] G. A. Pavliotis, *Stochastic Processes and Applications: Diffusion Processes, The Fokker–Planck and Langevin Equations*, Texts in Appl. Math. 60, Springer, 2014.
[59] P. Pons and M. Latapy, *Computing communities in large networks using random walks*, in Computer and Information Sciences, P. Yolum, T. Güngör, F. Gürgen, and C. Özturan, eds., Springer, 2005, pp. 284–293.

[60] M. A. Porter, J. P. Onnela, and P. J. Mucha, *Communities in networks*, Not. AMS, 56 (2009), pp. 1082–1097 & 1164–1166.
[61] M. Rosvall, D. Axelsson, and C. T. Bergstrom, *The map equation*, European Phys. J. Special Top., 178 (2009), pp. 13–23.
[62] S. Salihoglu and J. Widom, *GPS: A graph processing system*, in SSDBM: Proceedings of the 25th International Conference on Sci. Stat. Database Management, SSDBM, ACM, 2013, 22, https://doi.org/10.1145/2484838.2484843.
[63] S. Shai, N. Stanley, C. Granell, D. Taylor, and P. J. Mucha, *Case studies in network community detection*, in The Oxford Handbook of Social Networks, R. Light and J. Moody, eds., Oxford University Press, 2021, pp. 309–333, https://doi.org/10.1093/oxfordhb/9780190251765.013.16.
[64] J. Shi and J. Malik, *Normalized cuts and image segmentation*, IEEE Trans. Pattern Anal. Mach. Intell., 22 (2000), pp. 888–905.
[65] A. Singer and H.-T. Wu, *Spectral convergence of the connection Laplacian from random samples*, Inf. Inference, 6 (2017), pp. 58–123.
[66] D. L. Sussman, Y. Park, C. E. Priebe, and V. Lyzinski, *Matched filters for noisy induced subgraph detection*, IEEE Trans. Pattern Anal. Mach. Intell., 42 (2020), pp. 2887–2900, https://doi.org/10.1109/TPAMI.2019.2914651.
[67] D. Taylor, R. S. Caceres, and P. J. Mucha, *Super-resolution community detection for layer-aggregated multilayer networks*, Phys. Rev. X, 7 (2017), 031056.
[68] D. Taylor, S. Shai, N. Stanley, and P. J. Mucha, *Enhanced detectability of community structure in multilayer networks through layer aggregation*, Phys. Rev. Lett., 116 (2016), 228301.
[69] E. Thiede, B. V. Koten, and J. Weare, *Sharp entrywise perturbation bounds for Markov chains*, SIAM J. Matrix Anal. Appl., 36 (2015), pp. 917–941, https://doi.org/10.1137/140987900.
[70] V. A. Traag, R. Aldecoa, and J.-C. Delvenne, *Detecting communities using asymptotical surprise*, Phys. Rev. E, 92 (2015), 022816, https://doi.org/10.1103/PhysRevE.92.022816.
[71] N. G. Trillos and D. Slepčev, *Continuum limit of total variation on point clouds*, Arch. Ration. Mech. Anal., 220 (2016), pp. 193–241.
[72] N. G. Trillos and D. Slepčev, *A variational approach to the consistency of spectral clustering*, Appl. Comput. Harmon. Anal., 45 (2018), pp. 239–281.
[73] N. G. Trillos, D. Slepčev, J. Von Brecht, T. Laurent, and X. Bresson, *Consistency of Cheeger and ratio graph cuts*, J. Mach. Learn. Res., 17 (2016), pp. 6268–6313.
[74] Y. van Gennip, N. Guillen, B. Osting, and A. L. Bertozzi, *Mean curvature, threshold dynamics, and phase field theory on finite graphs*, Milan J. Math., 82 (2014), pp. 3–65, https://doi.org/10.1007/s00032-014-0216-8.
[75] A. M. Verdery, M. G. Merli, J. Moody, J. A. Smith, and J. C. Fisher, *Brief report: Respondent-driven sampling estimators under real and theoretical recruitment conditions of female sex workers in China*, Epidemiology, 26 (2015), pp. 661–665, https://doi.org/10.1097/EDE.0000000000000335.
[76] D. Wang and B. Osting, *A diffusion generated method for computing Dirichlet partitions*, J. Comput. Appl. Math., 351 (2019), pp. 302–316.
[77] L. Weng, F. Menczer, and Y.-Y. Ahn, *Virality prediction and community structure in social networks*, Sci. Rep., 3 (2013), 2522, https://doi.org/10.1038/srep02522.
[78] J. J. Whang, D. F. Gleich, and I. S. Dhillon, *Overlapping community detection using seed set expansion*, in Proceedings of the 22nd ACM International Conference on Information & Knowledge Management, 2013, pp. 2099–2108.
[79] J. D. Wilson, S. Wang, P. J. Mucha, S. Bhamidi, and A. B. Nobel, *A testing based extraction algorithm for identifying significant communities in networks*, Ann. Appl. Stat., 8 (2014), pp. 1853–1891, https://doi.org/10.1214/14-AOAS760.
[80] H. Yan, Q. Zhang, D. Mao, Z. Lu, D. Guo, and S. Chen, *Anomaly detection of network streams via dense subgraph discovery*, in 2021 International Conference on Computer Communications and Networks, pp. 1–9, https://doi.org/10.1109/ICCCN52240.2021.9522263.
[81] Z. Yang, T. Hao, O. Dikmen, X. Chen, and E. Oja, *Clustering by nonnegative matrix factorization using graph random walk*, Adv. Neural Inf. Process. Syst., 25 (2012), pp. 1079–1087.
[82] A. Yuan, J. Calder, and B. Osting, *A continuum limit for the PageRank algorithm*, European J. Appl. Math., 33 (2022), pp. 472–504, https://doi.org/10.1017/s0956792521000097.