STOCHASTIC BLOCK MODELS ARE A DISCRETE SURFACE TENSION*

ZACHARY M. BOYD†, MASON A. PORTER‡, AND ANDREA L. BERTOZZI§

Abstract. Networks, which represent agents and interactions between them, arise in myriad applications throughout the sciences, engineering, and even the humanities. To understand large-scale structure in a network, a common task is to cluster a network's nodes into sets called "communities" such that there are dense connections within communities but sparse connections between them. A popular and statistically principled method to perform such clustering is to use a family of generative models known as stochastic block models (SBMs). In this paper, we show that maximum likelihood estimation in an SBM is a network analog of a well-known continuum surface-tension problem that arises from an application in metallurgy. To illustrate the utility of this bridge, we implement network analogs of three surface-tension algorithms, with which we successfully recover planted community structure in synthetic networks and which yield fascinating insights on empirical networks from the field of hyperspectral video segmentation.

Key words. networks, community structure, data clustering, stochastic block models, Merriman–Bence–Osher (MBO) scheme, geometric partial differential equations

AMS subject classifications. 65K10, 49M20, 35Q56, 62H30, 91C20, 91D30, 94C15

1. Introduction.

The study of networks, in which nodes represent entities and edges encode interactions between entities [61], can provide useful insights into a wide variety of complex systems in myriad fields, such as granular materials [68], disease spreading [69], criminology [34], and more. In the study of such applications, the analysis of large data sets — from diverse sources and applications — continues to grow ever more important.

The simplest type of network is a graph, and empirical networks often appear to exhibit a complicated mixture of regular and seemingly random features [61]. Additionally, it is increasingly important to study networks with more complicated features, such as time-dependence [36], multiplexity [46], annotations [63], and connections that go beyond a pairwise paradigm [67]. One also has to worry about "features" such as missing information and false positives [44]. In this paper, we restrict attention to undirected, unweighted graphs for simplicity.

To try to understand the large-scale structure of a network, it can be very insightful to coarse-grain it in various ways [25, 73, 75, 77, 78]. The most popular type of clustering is the detection of assortative "communities," in which dense sets of nodes are connected sparsely to other dense sets of nodes [25, 75]. One of the most popular approaches, which is statistically principled, is to treat community detection as a statistical inference problem using a model such as a stochastic block model (SBM) [73]. The detection of communities has given fascinating insights into a variety of applications, including brain networks [8], social networks [83], granular networks [5], protein interaction networks [4], political networks [74], and many others.

One of the most popular frameworks for many recent approaches for detecting communities is to use an SBM, a generative model that can produce networks with community structure [25, 73].[1] One uses an SBM for community detection by fitting an observed graph to a statistical model to attempt to infer the most probable community assignment for each node. SBMs can incorporate a variety of features, including degree heterogeneity [42], hierarchical structure [70], and metadata [63]. The benefits of an SBM approach include statistical defensibility, theoretical tractability, asymptotic consistency under certain conditions, definable transitions between solvable and unsolvable regimes, and theoretically optimal algorithms [58, 73].

Recently, Newman showed that one can interpret modularity maximization [60, 64], which is still among the most popular approaches for community detection, as a special case of an SBM [62]. In another paper [38], it was shown that one can also interpret modularity maximization in terms of total-variation (TV) minimization. The latter connection allows the application of methods from geometric partial differential equations (PDEs) and ℓ^1 minimization to community detection. This raises the possibility of formulating

*Submitted to the editors DATE.

†Department of Mathematics, UCLA, Los Angeles, CA (zach.boyd@math.ucla.edu).

‡Department of Mathematics, UCLA, Los Angeles, CA (mason@math.ucla.edu).

§Department of Mathematics, UCLA, Los Angeles, CA (bertozzi@math.ucla.edu).

[1] Networks that are generated from an SBM can also have other types of block structures, depending on the choice of parameters; see subsection 2.1 for details.

SBM maximum-likelihood estimation (MLE) in terms of TV.[2] In this paper, we develop such a formulation using ideas from models of surface tension and crystal growth.

The main result of the present work is the establishment of an equivalence between SBMs and surface-tension models from the literature on PDEs that model crystal growth. Crystal growth is an important aspect of certain annealing processes in metallurgy [45, 59]. It is a consolidation process, wherein the many crystals in a metal grow and absorb each other to reduce the surface-tension energy that is associated to the interfaces between them. The various processes involved have been modeled from many perspectives, including molecular dynamics [17], front tracking [28], vertex models [91], and many others. (See [45] for a much more extensive set of references.) It has been observed experimentally that the interface between any two grains evolves according to motion by mean curvature [80]. Because mean-curvature flow is the gradient descent of the TV energy, this leads naturally to formulations in terms of level sets [66], phase fields [9], and threshold dynamics [56]. Although the interfaces follow mean-curvature flow, each different interface can evolve at a different rate, as there are different surface-tension densities between each pair of crystals. In realistic cases, surface tensions are both inhomogeneous and anisotropic, and they require careful adaptation of standard mean-curvature-flow approaches [21, 39], especially for dealing with the topological challenges that arise at crystal junctions, which routinely form and disappear.

Recently, Jacobs showed how to apply techniques from models of crystal growth to graph-cut problems from semisupervised learning [39]. (See also [40] for related work.) We also note that several recent papers, which do not directly involve surface tension, have used ideas from perimeter minimization and/or TV minimization for graph cuts and clustering in machine learning [7]. Three of those papers are concerned explicitly with ideas from network science [10, 38, 85].

Each community in a network is analogous to a crystal, and the set of edges between nodes from a pair of communities is akin to the topological boundary between a pair of crystals. The surface-tension densities correspond to the differing affinities between each pair of communities. To demonstrate the relevance of this viewpoint, we develop and test discrete analogs of surface-tension numerical schemes on several real and synthetic networks, and we find that straightforward analogs of the continuum techniques successfully recover planted community structure in synthetic networks and uncover meaningful structure in the real networks. We also prove a theoretical result, in terms of Γ-convergence, that one can meaningfully approximate the SBM MLE problem by smoother energies. Finally, we introduce three algorithms — inspired by work on crystal growth — that we test on synthetic and real-world networks.

The rest of this paper is organized as follows. In section 2, we present background information about stochastic block models, total variation, and surface tension. In section 3, we state and prove our main result, which establishes an equivalence between discrete surface tension and maximum likelihood estimation via an SBM. In section 4, we discuss three numerical approaches for performing SBM MLE: mean-curvature flow, Γ-convergence, and threshold dynamics. We discuss our results on both synthetic and real-world networks in section 5. In section 6, we conclude and discuss our results. We give additional technical details in appendices.

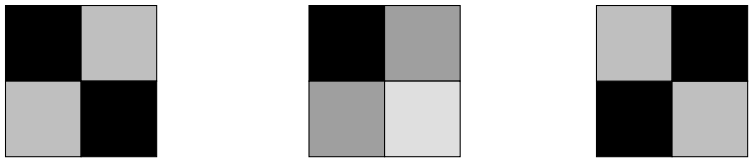
2. Background.

2.1. Stochastic Block Models (SBMs). The most basic SBM has N nodes and an assignment $g : \{1, \dots, N\} \to \{1, \dots, \hat{n}\}$ that associates each node with one of $\hat{n}$ sets. It also has an $\hat{n} \times \hat{n}$ symmetric, nonnegative matrix ω. One generates an undirected, unweighted graph as follows: for each pair of nodes, i and j, we place an edge between them with probability $\omega_{\alpha\beta}$, where α and β, respectively, denote the community assignments of nodes i and j. Similar models have been studied and rediscovered many times [18, 23, 25, 27, 35, 73, 81]. In the present paper, we use the SBM from [62].

There is considerable flexibility in the choice of ω, which leads in turn to flexibility in the SBMs themselves [25, 73]. Three examples of ω, using $\hat{n} = 2$, will help illustrate the diversity of block structures.

1. If $\omega_{11} = \omega_{22} > \omega_{12}$, one obtains traditional assortative community structure, in which nodes have a larger probability to be adjacent to nodes in the same community instead of ones in different communities.
2. If $\omega_{11} = \omega_{22} < \omega_{12}$, nodes tend to associate more with ones that are in other communities. As $\omega_{12} \to 0$, the graph becomes increasingly bipartite.

[2] Another recent paper [85] used total variation for maximizing modularity, although it was not phrased in those terms.



Community structure Core–periphery structure Bipartite structure

FIGURE 1. *Examples of different connectivity patterns that one can generate using stochastic block models. Each panel corresponds to a different kind of structure. In each panel, the upper-left and lower-right squares represent the density of connections between nodes in the same set, and the upper-right and lower-left squares represent the density of connections between nodes in different sets. Darker squares represent more densely connected sets of nodes. In (assortative) community structure, nodes are densely connected to other nodes in the same community but sparsely connected to nodes in other communities. In the core–periphery structure, core nodes (as illustrated by the dark square in the upper left) are densely both to other core nodes and somewhat densely connected to peripheral nodes, but the latter predominantly have connections only to core nodes. In bipartite block structures, a set of nodes is more densely connected to nodes in other sets than to nodes in its own set. One can also model other structures, such as hierarchical and role-based structures, using SBMs. See subsection 2.1 for additional discussion. [This figure is inspired by a figure from [41].]*

3. If $\omega_{11} > \omega_{12} > \omega_{22}$, there is a core–periphery (CP) structure: nodes from set 1 are connected densely to many nodes, but nodes from set 2 are connected sparsely to other nodes [19, 77].

These three examples are also illustrated in Figure 1. To simplify our presentation, we refer to latent block structures as "community structure," regardless of the form of ω.

The above SBM is not realistic enough for many applications, largely because each node has the same expected degree [42]. To address this issue, one can suppose that one knows the degree sequence $\{k_i\}$ and then define connection probabilities to take this information into account. The easiest approach is to model the adjacency-matrix elements A_{ij} as Poisson-distributed with parameter $\omega_{g_i g_j} \dfrac{k_i k_j}{2m}$, where m is the number of edges in the network. An important point to note is that this allows both multi-edges and self-edges. Although such edges can have important effects [26], we neglect them for simplicity.

Given an observed network, one can attempt to infer some sort of underlying community structure by statistical fitting methods. There are several ways to do this, including maximum-likelihood estimation (MLE), maximum a posteriori (MAP) estimation, and maximum marginal likelihood (MML). In MLE, one chooses the parameters g and ω under which an observed network is most probable (without using a prior), MAP yields the most probable parameter configuration under a Bayesian prior, and MML yields the best community assignment for each node individually by integrating out all of the other variables [58, 73]. We use MLE, which is the simplest approach. In mathematical terms, the problem is stated as

$$\operatorname*{argmax}_{g,\omega} P(A|g,\omega)\,, \tag{1}$$

where P is the probability density function. Because we determine the edges independently, P is given by

$$P(A|g,\omega) = \prod_{i\le j} P(A_{ij}|g,\omega) = \prod_{i\le j} P\left(A_{ij}\,\middle|\, w_{g_i g_j}\frac{k_i k_j}{2m}\right)\,.$$

We use a Poisson distribution, so

$$P(A_{ij}|\lambda) = \begin{cases} \dfrac{\lambda^{A_{ij}}}{A_{ij}!}e^{-\lambda}\,, & i \neq j\,, \\ \dfrac{\lambda^{A_{ij}/2}}{(A_{ij}/2)!}e^{-\lambda}\,, & i = j\,, \end{cases}$$

where the need for cases arises from the convention that $A_{ii} = 2$ if a self-edge is present. To solve (1), one can equivalently maximize the logarithm of $P(A|g,\omega)$. Conveniently, this changes the multiplicative structure into additive structure and allows one to drop irrelevant constants. The resulting objective function is

$$\operatorname*{argmax}_{g,\omega} \sum_{i,j}\left[A_{ij}\log(\omega_{g_i g_j}) - \omega_{g_i g_j}\frac{k_i k_j}{2m}\right]\,. \tag{2}$$

The exact optimization of (2) is NP-hard, so one needs to use a heuristic. Possibilities include greedy ascent [42], Kernighan–Lin (KL) node swapping [42,43], and coordinate descent [62]. As far as we are aware, the theory of these approaches has not received much attention, although the associated papers generally include positive results. In light of the extreme nonconvexity of the modularity objective function [33] (which is known to be related to the planted-partition form of the SBM [62]), we expect that multiple random initializations are needed for any local algorithm. (Ideas from consensus clustering may also be helpful [25].)

Ways to elaborate the above SBM include incorporating overlapping and hierarchical communities [70, 72], generalizing to structures such as time-dependent and multilayer networks [71], or incorporating metadata [63]. There are also Bayesian models and pseudo-likelihood-based methods [1, 73]. We do not consider such embellishments in this paper, although we conjecture that our approach will generalize to some of these settings.

2.2. Total Variation. We briefly introduce total variation (TV) and why it is interesting to establish a connection between SBM MLE and TV. Consider a smooth function $f : \Omega \subset \mathbb{R}^d \to \mathbb{R}$ for some d. The TV of f is

$$|f|_{\mathrm{TV}} = \int_{\Omega} |\nabla f| \mathrm{d}x \,. \tag{3}$$

For $d = 1$, (3) describes the total amount of increase and decrease of the function f. If f is smooth except for jump discontinuities, one can interpret the derivative of f in a generalized sense, yielding

$$|f|_{\mathrm{TV}} = \int_{\mathbb{R}^d - \Gamma} |\nabla f| \mathrm{d}x + \int_{\Gamma} |[f]| \, \mathrm{d}x \,,$$

where Γ is the union of all curves of discontinuity and $[f]$ is the height of the jump across the discontinuity. The first integral uses a d-dimensional measure, and the second uses a $(d-1)$-dimensional Hausdorff measure. In the particular case in which $d = 2$ and f is the characteristic function of some set S, we see that $|f|_{\mathrm{TV}}$ is the perimeter of S. Similarly, when $d = 3$, we obtain surface area.

Total variation is an important regularizer in machine learning. It is worth contrasting it with the Dirichlet energy $\int_{\Omega} |\nabla f|^2 \mathrm{d}x$, which has minimizers that satisfy $\Delta f = 0$, a condition that guarantees smoothness. However, minimizers of TV need not be smooth, as they can admit jump discontinuities. In image denoising, for instance, regularization using Dirichlet energy tends to blur edges to remove discontinuities, whereas a TV regularizer leaves the edges intact [14, 79].

Another use of TV energy is in relaxations, in which one can transform a nonconvex problem involving piecewise-constant constraints into a convex problem with the same minimizers [14, 54]. A common heuristic explanation for this phenomenon (see Figure 2) uses the shape of the 1-norm unit ball. The simplest case is in two dimensions, where the 1-norm ball is diamond-shaped, and minimizing the 1-norm over certain domains (e.g., a line) gives a sparse solution, in the sense that most components of the solution vector are 0. In this case, minimizing the 1-norm, constrained to a line, is the same as minimizing the number of nonzero elements of the vector, subject to the same constraint.

In the context of TV minimization, we take the 1-norm of a function's gradient, rather than of the function itself. Thus, instead of promoting sparsity of the function values, we promote sparse gradients, thereby incentivizing piecewise-constant minimizers for TV. Our discussion above is heuristic, but the ideas therein can be treated rigorously [14].

Algorithmically, one can minimize TV using, for example, phase-field models [9] or threshold dynamics [56], both of which rely on the fact that the gradient descent of TV is a generalization of mean-curvature flow. The alternating-directions method of multipliers (ADMM) [32] and graph-cut methods, such as the one in [12], also solve similar problems very effectively.

Thus far, we have restricted our discussion of TV to a continuum setting. There are graph analogs of the mathematical objects — gradients, measures, integrals, tangent spaces, divergences, and so on — that one uses to define TV in a continuum setting. For instance, for any function f on the nodes of a graph and any edge between nodes i and j, the discrete derivative at i in the direction j is

$$\nabla f(i, j) = f(j) - f(j) \,.$$

Using the inner products

$$\langle f, g\rangle = \sum_{i=1}^{N} f_i g_i \,,$$
$$\langle \phi, \psi\rangle = \sum_{i,j} A_{ij}\phi_{ij}\psi ij$$

on the spaces of functions on the nodes and edges, respectively, gives the divergence as the adjoint of the gradient:

$$(\operatorname{div}\phi)_i = \sum_j A_{ij}\phi_{ji}\,.$$

In a continuum, an alternative definition of TV is

$$|f|_{\mathrm{TV}} = \sup\langle \operatorname{div}\phi, f\rangle\,, \tag{4}$$

where the supremum is over an appropriate set of test functions. For a graph, (4) is equivalent to

$$|f|_{\mathrm{TV}} = \frac{1}{2}\sum_{i,j} A_{ij}|f(i) - f(j)|\,.$$

See [31,87] for a detailed justification of these definitions.

Some methods for graph clustering (e.g., see [90]) rely on the combinatorial graph Laplacian $L = \operatorname{diag}(k) - A$, which is a discrete analog of the continuum Laplacian Δ. The continuum Laplacian arises in solutions to constrained optimization problems that involve the Dirichlet energy, so it is reasonable to expect minimizers of energies that involve the combinatorial graph Laplacian to have analogous properties to minimizers of the Dirichlet energy. Indeed, it is well-known that the minimizers that arise from spectral methods are usually smooth, instead of having sharp interfaces, so one needs to threshold them in some way. Such thresholding is a major source of difficulties for attempts to obtain theoretical guarantees about the nature of minimizers after thresholding. In contrast, methods that use graph TV can directly accommodate piecewise-constant solutions [54], which do not require thresholding to give classification information. Several previous papers have exploited this property of TV on graphs [6,38,84,93].

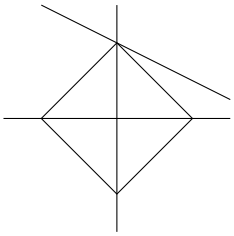


FIGURE 2. *Image of the* 1*-norm unit ball and a line in the plane. The point on the line with the smallest* 1*-norm is almost always on one of the axes.*

In the case of SBMs, one can use TV to express (2), but we find a more natural formulation in terms of surface-tension energy (a related notion).

2.3. Surface Tension. Very roughly, one can consider a metal object as being composed of a large number of crystals that range in size from microscopic to macroscopic [3]. Each crystal is a highly-ordered lattice; and there is a thin, disordered interface between crystals. The sizes and orientations of these crystals affect material properties, and one goal of annealing processes is to allow crystals to reorganize to produce a useful metal.

The potential energy of a given crystal configuration is roughly

$$\sum_{\alpha,\beta} \sigma_{\alpha\beta}\mathrm{Area}(\Gamma_{\alpha\beta})\,, \tag{5}$$

where $\Gamma_{\alpha\beta}$ is the interface between crystals α and β, and $\sigma_{\alpha\beta}$ is the surface tension energy density between these crystals. Each $\sigma_{\alpha\beta}$ is different, based on physical considerations that involve the exact offset between the orientations of the lattices in each pair of crystals. When prepared and heated appropriately, the individual crystals decrease (5) by growing to consume their neighboring crystals. See [21,39,45] for further background information.

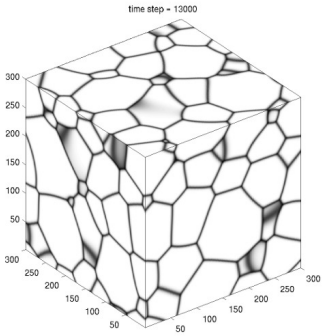


FIGURE 3. *An example arrangement of crystals. The interfaces between pairs of crystals grow into each other according to motion by mean curvature. [This image from Cenna/Wikimedia Commons/Public Domain [15].]*

We exploit the appearance of surface area in (5) to cast it as a TV problem. Mathematically, we model the metal as a region of space that is partitioned into $\hat{n}$ regions, corresponding to the crystals in the metal. Let u^α and u^β, respectively, denote the characteristic functions of the regions α and β. Therefore,

$$\text{Area}_{\alpha\beta} = |u^\alpha|_{\text{TV}} + |u^\beta|_{\text{TV}} - |u^\alpha + u^\beta|_{\text{TV}} \,.$$

Each interface between two regions evolves according to mean-curvature flow, which is the gradient descent of TV. Thus, the surface-tension flow is locally mean-curvature flow, except at the junction of three or more crystals [21,39]. Because of this connection, one can use some of the ideas (such as phase-field and threshold-dynamics methods [21]) from TV minimization to perform surface-tension minimization. When using threshold dynamics, it is possible to do theoretical analysis in the form of Lyapunov functionals, Γ-convergence, and descent conditions [39].

3. An Equivalence Between SBM MLE and Discrete Surface Tension. We now present a mathematical result that connects SBM MLE and discrete surface tension.

PROPOSITION 3.1. *Maximizing the likelihood of the parameters g and ω in the degree-corrected SBM is equivalent to minimizing*

$$\sum_{\alpha,\beta} \left[W_{\alpha\beta} \text{Cut}_{g,A}(\alpha,\beta) + e^{-W_{\alpha\beta}} \frac{\text{vol}_{g,A}(\alpha)\,\text{vol}_{g,A}(\beta)}{2m} \right] , \tag{6}$$

where $\text{Cut}_{g,A}(\alpha,\beta) = \sum_{\substack{g_i=\alpha \\ g_j=\beta}} A_{ij}$, $\text{vol}_{g,A}(\alpha) = \sum_{g_i=\alpha} k_i$, *and* $W_{\alpha\beta} = -\log\omega_{\alpha\beta}$.

The analogy with continuum surface tension is as follows. Graph cuts are analogous to surface area: given a domain in $\mathbb{R}^3$, one can superimpose a fine grid on space and count the number of edges that cross the boundary to estimate its surface area. In the limit of an infinitely fine grid, this estimate converges to the surface area under appropriate conditions [11]. Similarly, graph volumes are analogous to continuum volumes. The quantities $W_{\alpha\beta}$ play the role of surface tensions $\sigma_{\alpha\beta}$, so the first set of terms is analogous to (5). One can view the second set of terms as a soft volume constraint. A constraint is "soft" if violating it adds a finite penalty on an objective function, so minimizers will usually approximately satisfy the constraint. Volume-constrained versions of (5) have received a great deal of attention [40,45].[3]

Proof. In [62], it was shown that maximizing the log-likelihood of the parameters g and ω for a particular version of the degree-corrected SBM amounts to maximizing (2). Let $\Pi(G,\hat{n})$ be the set of partitions of the nodes of a graph G (associated with an adjacency matrix A) into at most $\hat{n}$ sets. Substituting $W_{\alpha\beta} = -\log\omega_{\alpha\beta}$ into (2) gives

$$\underset{\substack{W_{\alpha\beta}\in\mathbb{R} \\ g\in\Pi(G)}}{\text{argmin}} \sum_{i,j} \left[A_{ij} W_{g_i g_j} + \frac{k_i k_j}{2m} e^{-W_{g_i g_j}} \right] .$$

Rearranging the summations gives

$$\underset{\substack{W_{\alpha\beta}\in\mathbb{R} \\ g\in\Pi(G,\hat{n}) \\ \hat{n}\in\mathbb{N}}}{\text{argmin}} \left[\sum_{\alpha,\beta} \sum_{\substack{g_i=\alpha \\ g_j=\beta}} A_{ij} W_{\alpha\beta} + \sum_{\alpha,\beta} \sum_{\substack{g_i=\alpha \\ g_j=\beta}} \frac{k_i k_j}{2m} e^{-W_{\alpha\beta}} \right] ,$$

[3] As far as we are aware, our formulation of SBM MLE in terms of graph cuts and volumes is novel, although similar formulas have appeared previously in the literature; see, e.g., [73].

where the inner sums are over all nodes i and j such that $g_i = \alpha$ and $g_j = \beta$. Rearranging again gives

$$\underset{\substack{W_{\alpha\beta}\in\mathbb{R}\\ g\in\Pi(G,\hat{n})\\ \hat{n}\in\mathbb{N}}}{\operatorname{argmin}} \left[\sum_{\alpha,\beta} W_{\alpha\beta} \sum_{\substack{g_i=\alpha\\ g_j=\beta}} A_{ij} + \sum_{\alpha,\beta} e^{-W_{\alpha\beta}} \sum_{\substack{g_i=\alpha\\ g_j=\beta}} \frac{k_i k_j}{2m} \right] .$$

Using the definition of $\mathrm{Cut}_{g,A}$ in the first set of terms and summing over the j index independently in the second set of terms gives

$$\underset{\substack{W_{\alpha\beta}\in\mathbb{R}\\ g\in\Pi(G,\hat{n})\\ \hat{n}\in\mathbb{N}}}{\operatorname{argmin}} \left[\sum_{\alpha,\beta} W_{\alpha\beta}\mathrm{Cut}_{g,A}(\alpha,\beta) + \sum_{\alpha,\beta} e^{-W_{\alpha\beta}} \sum_{g_i=\alpha} \frac{k_i}{2m}\mathrm{vol}_{g,A}(\beta) \right] .$$

Finally, we sum over the i index in the second set of terms to obtain

$$\underset{\substack{W_{\alpha\beta}\in\mathbb{R}\\ g\in\Pi(G,\hat{n})\\ \hat{n}\in\mathbb{N}}}{\operatorname{argmin}} \sum_{\alpha,\beta} \left[W_{\alpha\beta}\mathrm{Cut}_{g,A}(\alpha,\beta) + e^{-W_{\alpha\beta}} \frac{\mathrm{vol}_{g,A}(\alpha)\,\mathrm{vol}_{g,A}(\beta)}{2m} \right] . \tag{7}$$

□

One difference between (6) and (5) is that in the latter (i.e., for a graph), one performs optimization over the $W_{\alpha\beta}$, whereas in the former (i.e., in a continuum), one ordinarily treats the surface-tension densities as fixed by the choice of material that one is modeling. Another difference is that the surface-tension coefficients in the graph setting can be any element of $(-\infty, \infty]$, subject only to the symmetry condition $W_{\alpha\beta} = W_{\beta\alpha}$. In contrast, for a continuum, further restrictions are necessary to ensure well-posedness. Esedoglu and Otto [21] proved the following sufficient conditions for well-posedness:

(1) $\sigma_{\alpha\beta} \geq 0$,
(2) $\sigma_{\alpha,\alpha} = 0$,
(3) $\sigma_{\alpha\gamma} + \sigma_{\gamma\beta} \geq \sigma_{\alpha\beta}$.

In a graph setting, one can use a straightforward change of variables to make W satisfy requirement (2).[4] In general, however, at least one of requirements (1)–(3) are not necessarily satisfied for a graph.[5]

4. Mean-Curvature Flow (MCF), Γ-Convergence, and Threshold Dynamics. We now outline three algorithmic approaches that illustrate how one can use tools from surface-tension theory to solve SBM MLE problems. Our three algorithms are graph versions of mean-curvature flow (MCF), Allen–Cahn (AC) evolution, and Merriman–Bence–Osher (MBO) dynamics. In section 5, we conduct several numerical experiments to demonstrate that these algorithms can effectively solve (2). We expect the performance of these algorithms to be good relative to other algorithms for SBM MLE, though a full evaluation of this claim is beyond the scope of our paper. We have posted our code at http://www.math.ucla.edu/~zach.boyd/code/SBM.zip. In the next three subsections, we describe how we infer g when ω is fixed. We then describe how to jointly infer ω and g.

4.1. Mean-Curvature Flow. Surface-tension dynamics are governed by mean-curvature flow except at junctions. Intuitively, this means that each point on a surface moves in the direction normal to the surface at a speed given by the mean curvature at that point. In the two-phase case, such dynamics have been well-studied, and notions of viscosity solutions and regularity theory have been developed [52]. In the multiphase case, the situation is much more complicated, notably because of the topological changes that can occur and the issue of defining the behavior at the junction of three or more phases. In two-phase surface-tension dynamics, it was shown in [12] that one can approximate the flow by solving a discrete-time minimizing-movements problem. Let C_n be one of the two regions at time $n\,dt$, where dt is the time step.

[4] See section A for the change of variables, which causes the sum in (6) to instead be over all $\alpha \neq \beta$, so that there are no "internal" surface tensions.

[5] Requirement (1) is false whenever some component of W is negative; this occurs exactly when ω has a component that is larger than 1. Requirement (3) may not hold, because the the components of W can assume any real value. Thus, it is possible to pick some W that violates requirement (3) and generate a network using it.

We then have

$$C_{n+1} = \operatorname{argmin}\left[\text{SurfaceArea}(C) + \frac{1}{dt}\int_{C_n \Delta C} \hat{\rho}(p, C_n)\mathrm{d}p\right], \tag{8}$$

where

$$\hat{\rho}(p, C_n) = \inf_{x \in \partial C_n} \|x - p\|,$$

the operation Δ denotes the symmetric difference, and ∂ is the topological boundary operator. The idea behind this approach is, at each time step, to shorten the curve as much as possible without straying too far from the curve location at the previous time step.

In the setting of graphs, a similar approach was developed in [87], where the mean-curvature flow was given by

$$C_{n+1} = \operatorname{argmin}\left[\text{Cut}_{g,A}(C, C^c) + \frac{1}{dt}\sum_{i \in C_n \Delta C} \rho(i, \partial(C_n))\right], \tag{9}$$

the operation Δ is again the symmetric difference, and $\rho(i, \partial(C_n))$ is the shortest-path distance from node i to the boundary of C_n. In this context, the boundary of a set of nodes is the set of nodes in C_n with at least one neighbor in C_n^c along with the nodes in C_n^c that have at least one neighbor in C_n. We use the term *boundary node* for any node that lies on the boundary. In the limit of small dt, (9) may still evolve, as opposed to the MBO scheme (which we use later), which becomes "stuck" when the time step is too small. Such evolution can still occur, because the penalty (associated with moving any node in $\partial(C_n)$) induced by the second set of terms in (9) is 0, regardless of the value of dt. Conveniently, this implies for sufficiently small dt that the only acceptable moves at each time step are allowed to change only the boundary nodes themselves. This makes it possible to drastically reduce the search space when solving (9).

Because careful studies in the spirit of [87] are not yet available for multi-way graph partitioning, we resort to a heuristic approach based on what is known in the binary case. Specifically, we are motivated by situation in which time steps are sufficiently small that only boundary nodes can change their community assignment. Ideally, we wish to compute an optimal reassignment of all boundary nodes jointly in order to minimize (6). To save computation time and facilitate implementation, we instead decouple the computations in the following manner: During a single time step, for each boundary node, we compute an optimal assignment of that node, assuming that all other nodes keep their assignment from the beginning of the time step. After this (but before the end of the time step), we assign each boundary node to its community, as computed previously in the time step. Because most nodes are boundary nodes[6] in our SBM-generated graphs, we find it more efficient and easier to consider reassigning all nodes in each time step rather than maintaining and referencing a separate data structure to track the boundary. In Algorithm 1, we give pseudocode for this graph MCF procedure.

4.2. Allen–Cahn (AC) Evolution. Another approach for studying mean-curvature flow, that is popular due to its simple implementation and the existence of unconditionally-stable numerical methods [6], is approximation by a Ginzburg–Landau (GL) functional. In the binary case, the GL functional is

$$\int_\Omega \left[\epsilon\,|\nabla u|^2 + \frac{1}{2\epsilon}u^2(1-u)^2\right]\mathrm{d}x, \tag{10}$$

where $u : \Omega \subset \mathbb{R}^N \to \mathbb{R}$ is a smooth function and ϵ is a small parameter. The L^2 gradient descent of the GL functional is

$$u_t = \epsilon\Delta u - \frac{1}{2\epsilon}\frac{\mathrm{d}}{\mathrm{d}u}\left[u^2(1-u)^2\right],$$

[6]Recall that a node is a boundary node if it shares an edge with a node that lies outside of its own community, so most reasonable partitions of many real graphs have many boundary nodes. Additionally, because we use a random initialization of g, most nodes will initially be boundary nodes for most graphs.

Algorithm 1 Modified graph mean-curvature flow (MCF) for SBM MLE (2).

Input A, W, $\hat{n}$.
Initialize g uniformly at random.
Let $eW = e^{-W}$ be the entry-wise exponential of W.
while not converged **do**
  Let $U_{i\alpha} = \delta_{g_i \alpha}$ for each i, α.
  Let $X = AU$. // Counts the number of neighbors that each node has in each class
  Let $\mathrm{vol}_{g,A} = (k^T U)$.
  for $a' = 1$ to $\hat{n}$ **do**
    Let $I_{a'}$ be the set of nodes currently assigned to group a'.
    for a=1 to $\hat{n}$ **do**
      Let I be the indices $1, \ldots, \hat{n}$ aside from a and a'.
      Let Delta$(I_{a'}, a)$ be given by the following formula:

$$\begin{aligned}\mathrm{Delta}(I_{a'}, a) = {} & 2X(I_{a'}, I)W(I, a) \\ & - 2X(I_{a'}, I)W(I, a') \\ & + 2X(I_{a'}, a)W(a, a) \\ & - 2X(I_{a'}, a)W(a, a') + 2X(I_{a'}, a')W(a, a') \\ & - 2X(I_{a'}, a')W(a', a') \\ & + \frac{1}{2m}\left(2k(I_{a'})\mathrm{vol}_{g,A}(eW(:, a) - eW(:, a')) \right. \\ & \left. + k(I_{a'})^2(eW(a, a) + eW(a', a') - 2eW(a, a'))\right).\end{aligned}$$

    end for
  end for
  for $i = 1$ to N **do**
    $g_i = \mathrm{argmin}(\mathrm{Delta(i, :)})$. // [Choose uniformly at random in case of a tie.]
  end for
end while
Output g.

which is the Allen–Cahn (AC) equation. The minimizers of the GL energy are mostly constant, with $O(\epsilon)$-width transition layers between the constant regions. Further, one can show that the GL energy Γ-converges to the TV energy as $\epsilon \to 0$, assuming that $\int_\Omega u \, \mathrm{d}x = \mathrm{const}$ [57]. Consequently, if u_ϵ is a minimizer of the constrained GL energy with parameter ϵ and the minimizers converge in bounded-variation (BV) space as $\epsilon \to 0$, then the accumulation point is a minimizer of the TV energy.

In the setting of graphs, the first use of AC schemes for TV minimization was in [6]. One can invoke the combinatorial graph Laplacian $L = \mathrm{diag}(k) - A$ to obtain the functional

$$U^T L U + \frac{1}{\epsilon} U^2 (1 - U^2), \tag{11}$$

where U is a function on the graph nodes (i.e. an N-element vector), and ϵ again a positive number. Equation (11) Γ-converges to graph TV [86].

In the multiphase case, we represent a partition g by an $N \times \hat{n}$ matrix whose i, α entry is $\delta(g_i = \alpha)$, where δ is the Kroneker delta. Then, instead of a double-well potential, we use a multi-well potential on $\mathbb{R}^{N \times \hat{n}}$ that is small for arguments with exactly one nonzero entry in each row. For example, [29] found that the following potential works well:

$$T(U) = \sum_{i=1}^{N} \left(\prod_{\alpha=1}^{\hat{n}} \frac{1}{4} \left\| U_i - e_k \right\|_{\ell^1}^2 \right),$$

where U_i is the ith row of the $N \times \hat{n}$ matrix U and e_k is an $\hat{n}$-element vector that is equal to 0 except for a

1 in the kth entry.

For the particular case of surface-tension dynamics, we proceed as follows. Given a partition of a network, if U is the corresponding $N \times \hat{n}$ matrix, one can show that $W. * (U^T L U) = -W. * (U^T A U)$, where $.*$ is the entry-wise product. Therefore, an appropriate GL functional for our problem is

$$\sum_{\alpha,\beta} \left[-W_{\alpha\beta} U_\alpha^T L U_\beta + \frac{\mathrm{vol}_{g,A}(\alpha)\,\mathrm{vol}_{g,A}(\beta)}{2m} \right] + \frac{1}{2\epsilon} T(U)\,. \tag{12}$$

Because $k^T u$ gives the vector of volumes, one can rewrite (12) as

$$\sum_{\alpha,\beta} \left[-W_{\alpha\beta} U_\alpha^T L U_\beta + \frac{k^T U_\alpha e^{-W}_{\alpha\beta} U_\beta^T k}{2m} \right] + \frac{1}{2\epsilon} T(U)\,, \tag{13}$$

where e^{-W} is the entry-wise exponential.

As in a continuum setting, one can prove Γ-convergence.

THEOREM 4.1. *The functionals in* (13) Γ*-converge to* (6) *as* $\epsilon \to 0$.

See section B for a proof. As far as we are aware, this is the first Γ-convergence result for a multiphase graph energy.

The resulting AC equation is

$$U_t = LUW - \frac{1}{2m} k k^T U e^{-W} - \frac{1}{\epsilon} T'(U)\,. \tag{14}$$

See section C for further details on the numerical solution of (14).

Algorithm 2 A two-phase, continuum MBO scheme.

Input the initial domain.
Initialize u as the characteristic function of the initial domain.
for $i = 1, \ldots$ **do**
  $u^{i+1/2}$ is the solution at time dt of $u_t = \Delta u$ with initial condition u^i.
  $u^{i+1} = \lfloor u^{i+1/2} + 0.5 \rfloor$, where $\lfloor \cdot \rfloor$ is the floor function.
end for
Output the set of points for which $u = 1$.

4.3. MBO Iteration. In [56], Merriman, Bence, and Osher showed that continuum mean-curvature flow is well-approximated by the simple iteration in Algorithm 2. In a rectangular domain, the iteration is extremely efficient, as one can use a fast Fourier transform when solving the heat equation. Esedoglu and Otto [21] developed a generalized version of the MBO scheme (see Algorithm 3) for computing the evolution of multiphase systems modeled by (5).

One can apply the MBO idea to community detection in networks by replacing the continuum Laplacian with the (negative) combinatorial graph Laplacian, replacing σ with W, changing u to U, and adding appropriate forcing terms for the gradient descent of the volume-balance terms. See section C for additional implementation details.

4.4. Learning ω. The MCF, AC, and MBO algorithms are able to yield a good partition of a network, given W, but they do not include a way to find W. A simple way to address this issue is to use an expectation-maximization (EM) algorithm, in which one alternates between an update of g with fixed W and an update of W with fixed g. One can find the optimal W, given g, in closed form by differentiating (6) with respect to any component of W and setting the result to 0 [42].

One must be careful, however, because the optimal $W_{\alpha\beta}$ when $\mathrm{Cut}_{g,A}(\alpha,\beta) = 0$ is infinite. This is a problem, because once one of the entries in W is infinite, it prevents g in subsequent iterations from taking any nonzero value of $\mathrm{Cut}_{g,A}(\alpha,\beta)$, which gives bad results in our test examples. (See section 5 for a discussion of these examples.) We address this issue by modifying the EM algorithm to reset all infinite values of W to $1.1 \times W_{\max}$, where $W_{\max}$ is the largest non-infinite element of W.

Algorithm 3 A multiphase, continuum MBO scheme.

Input the initial state of the domain.
Initialize $u_1, \dots, u_{\hat{n}}$ as the characteristic functions of the initial domains.
for $i = 1, \dots$ **do**
  for $\alpha = 1, \dots, \hat{n}$ **do**
    $u_\alpha^{i+1/2}$ is the solution at time dt to $u_{\alpha,t} = \Delta u_\alpha$ with initial condition u_α^i.
  end for
  for each point x **do**
    $\hat{\alpha} = \operatorname{argmin}_\alpha \sum_\beta \sigma_{\alpha\beta} u_\beta(x)$. // [Choose uniformly at random in case of a tie]
    $u_{\hat{\alpha}}(x) = 1$ and $u_\beta(x) = 0$ if $\beta \neq \hat{\alpha}$.
  end for
end for
Output u.

We also need to address another practical issue for an EM approach to work. Specifically, the algorithm that we have described thus far in this section often finds bad local minima, in which two communities are merged erroneously or a single community is split inappropriately. To overcome this issue, we implement a wrapper function (see Algorithm 4) that checks each community that is returned by MCF, AC, or MBO for further possible splitting or merging with other communities. Whenever we call MCF, AC, or MBO on a subgraph, we use the values of k and m for the whole graph rather than for a subgraph.[7]

There is also a danger of overfitting by setting $\hat{n} = N$, which gives a likelihood of 1 in (2). The proper selection of $\hat{n}$ is a complicated problem, both algorithmically and theoretically [65, 76]. For our tests, we were very successful by using a simple heuristic approach. (However, our framework in this paper is also compatible with more sophisticated methods for selecting $\hat{n}$.) For each data set, one supplies an expected value of $\hat{n}$ for that data set, and one adds a quadratic penalty to the objective value whenever $\hat{n}$ differs from its expected value. This helps curtail overfitting, while still allowing our algorithms to perform merges and splits to escape bad local minima.

Algorithm 4 The splitting–merging wrapper that we use to escape bad local minima.

Input A, $\hat{n}_{\text{expected}}$.
Place all nodes in the same community and add this community to a queue.
while the queue is not empty **do**
  Save $g_{\text{old}} = g$ and $W_{\text{old}} = W$.
  Save the current objective value as Q_{old}.
  Partition the next community in the queue (as a graph in its own right) into $\min(\hat{n}_{\text{expected}}, \sqrt{N})$ communities using MCF, AC, or MBO with $w_{\alpha\beta} = \begin{cases} 1, & \alpha = \beta, \\ 0.1, & \alpha \neq \beta. \end{cases}$
  while it is possible to improve the objective by merging **do**
    Perform the merge that most improves the objective.
  end while
  if the objective is better than Q_{old} **then**
    Add any newly created communities to the queue.
  else
    Set $g = g_{\text{old}}$ and $W = W_{\text{old}}$.
    Remove the current community from the queue.
  end if
end while
Output g, W.

[7] A similar choice was used for the recursive partitioning procedures in [38, 60].

5. Empirical Results. In this section, we discuss our results from several numerical experiments to (1) confirm that our algorithms can successfully recover g and ω from networks that we generate using SBMs and (2) explore their applicability to real-world networks. In our experiments, we use three different families of SBMs, three Facebook networks (whose community structure is partly understood [82,83]), and an example related to hyperspectral video segmentation. Because of the random initialization in our approach, we perform three trials on each of the networks for each algorithm, and we report the best result in each case.[8] For comparison, we also report the results of a Kernighan–Lin (KL) algorithm, which was reported in [42] to be effective. We summarize our results in Table 1, and we highlight that we consistently recover the underlying structure in the synthetic examples. For the real networks, we compare our results with a reference partition based on metadata that is thought to be correlated with the community structure. We find that the MCF scheme performs the best among our three schemes on these networks, and it finds partitions with a larger likelihood than the reference partition. We implement our methods in MATLAB, so one should interpret our computation time as indicative that the run time is reasonable for networks with millions of edges and perhaps on larger networks, given a careful implementation in a compiled language. For an example of code for a similar problem that was solved by an MBO scheme at large scale, see [53].

We briefly describe the three families of SBM-generated networks that we use in our numerical experiments.

- Planted partition (PP) is a 16,000-node graph that consists of 10 equally-sized communities. It is produced by the method that was described in [42]. It builds a degree-corrected SBM with a truncated power-law degree distribution with exponent 2. The parameter λ from Equation (27) in [42] is 0.001, indicating a fairly clear separation between communities.
- Lancichinetti–Fortunato–Radicchi (LFR) is a standard benchmark SBM network [48]. We construct 1000-node LFR graphs with a power-law degree distribution (with exponent 2), mean degree 20, maximum degree 50, power-law-distributed community sizes (with exponent 1), community sizes between 10 and 50 nodes, and mixing parameter 0.1.
- Multiscale SBM (MS). To construct such a graph, we take a union of disjoint components: a 10-clique, a 20-clique, and a sequence of Erdős–Rényi (ER) graphs (drawn from the $G(n,p)$ model with expected mean degree 20) of sizes 40, 80, 160, ..., 5120; there are a total of $10,230$ nodes. We connect the components to each other by adding a single edge, from nodes chosen uniformly at random, between each consecutive clique or ER graph. This construction tests whether an algorithm can find communities of widely varying sizes in the same graph [2, 24].

The hyperspectral video is a recording of a gas plume as it was released at the Dugway Proving Ground [30, 51, 55]. A hyperspectral video is different from an RGB video, in that each pixel in the former encodes the intensity of light at a large number (e.g., 129, in this case) of different wavelengths rather than only 3. We consider the classification problem of identifying pixels that include similar materials (such as dirt, road, grass, and so on). This problem is difficult, because of the diffuse nature of the gas, which leads to a faint signal that spreads out among many wavelengths and with boundaries that are difficult to determine. We construct a graph representation of this video using "nonlocal means," as described in [13]. Specifically, we use the following construction. For each pixel p and in each of 7 frames, we construct a vector v_p by concatenating the data in a 3×3 window that is centered at p. We then use a weighted cosine distance as a similarity measure on these $(3 \times 3 \times 129)$-component vectors, where we give the most weight to the components from the center of the window.[9] Finally, using the VLFEAT software package [88], we build a 10-nearest-neighbor graph using the similarity measure and a k-dimensional tree (with $k = 10$). We see from Figure 4 that partitions with small values of (6) correspond to meaningful segmentations of the image.

In Table 3, we include an example of a W matrix that we obtain from an MS network to illustrate that

[8] We chose to use three trials to illustrate that our algorithms do not require a large number of attempts to reach a good optimum. In most of our trials, even a single run of the solver is likely to give good results. In Table 1, we report our best scores. Our worst scores for MCF are 0.00, 0.00, 0.00, −0.14, and 0.01 for the the PP, MS, LFR, Caltech, and Princeton networks, respectively. We did not record the worst score for Penn. St. or the plume network. Our corresponding worst scores for AC and MBO, respectively, are 0.00, 0.00, 0.01, 0.22, 0.86 and 0.15, 0.00, 0.02, 0.53, 1.12. Comparing these results with Table 1, we see that our best and worst scores are often similar to each other.

[9] We weight the center pixel components by 1, the components from adjacent pixels by 0.5, and the components from corner pixels by 0.25. That is, we let v_{ij} be the 129-element vector at pixel (i,j), and we define w_{ij} as the concatenation of v_{ij}, $.5v_{i+1,j}$, $.5v_{i-1,j}$, $.5v_{i,j+1}$, $.5v_{i,j-1}$, $.25v_{i+1,j+1}$, $.25v_{i+1,j-1}$, $.25v_{i-1,j+1}$, and $.25v_{i-1,j-1}$. We then calculate the cosine similarity between each pair of w_{ij} vectors.

		PP	LFR	MS	Caltech	Princeton	Penn. St.	Plume
	Nodes	16,000	1,000	10,230	762	6,575	41,536	284,481
	Edges	2.9×10^5	9.8×10^3	1.0×10^5	16,651	293,307	1,362,220	2,723,840
	Communities	10	40	10	8	4	8	5
Score	MCF	0	0	0	−0.16	−0.02	−0.56	−1.41
	AC	0	0	0	0.21	0.58	−0.04	−1.23
	MBO	0	0	0	0.53	1.12	0.40	−1.21
	KL	0.28	0.03	0.04	−0.16	0.11	−0.55	−1.38
	Reference	0	0	0	0	0	0	0

TABLE 1

Results of several tests on several synthetic and empirical networks. We use three surface-tension-based methods (mean-curvature flow, Allen–Cahn, and Merriman–Bence–Osher) and the Kernighan–Lin algorithm from [42] to partition three synthetic networks (Planted Partition, LFR, and Multiscale SBM) and the largest connected components of three empirical networks (Caltech36, Princeton12, and Penn94) from the FACEBOOK100 *data set [83]. The score is the difference between the recovered surface-tension energy (6) and the corresponding energy of a reference partition, divided by the absolute value of the energy of the reference partition. Smaller values indicate better performance, and* 0 *corresponds to a partition that is of comparable quality as the reference partition. For the synthetic networks, we use the planted (and hence ground-truth) community structure as the reference partition. For the Facebook networks, we use metadata that is positively correlated with community structure (namely, House affiliation for Caltech and graduation year for the others). For the plume video, our reference partition is to assign all nodes to the same community. The edge counts on the synthetic networks give the order of magnitude, because the exact number differs across realizations.*

FIGURE 4. *Segmentation of a hyperspectral video using graph MCF. The gas plume is clearly represented in the yellow and orange pixels. The two bottom blue classes are the ground, and the other two are the sky. This image is frame 3 of 7.*

we recover different surface tensions between different pairs of communities.[10]

6. Conclusions and Discussion. We have shown that a particular stochastic block model (SBM) maximum likelihood estimation problem is equivalent to a discrete version of a well-known surface-tension problem. The equivalence associates graph cuts to surface areas and SBM parameters to physical surface tensions. This gives new geometric and physical interpretations to SBM MLE problems, which are traditionally viewed from a statistical perspective. We used the new connection to adapt three well-known surface-tension-minimization algorithms to community detection in graphs. Our subsequent computations suggest that the result algorithms are able to successfully find underlying community structure in SBM-generated graphs. When applied to graphs that are constructed from empirical data, our mean-curvature-flow method performs very well, but the other two methods face some issues (which will be interesting to explore in future studies). We also proved a Γ-convergence result that gives theoretical justification for our algorithms and is the first multiphase Γ-convergence result of which we are aware.

Although the focus of our paper has been a specific form of an SBM and an associated MLE problem, our techniques should also be insightful for other SBM problems. One straightforward adaption is to consider SBMs without degree correction, although that is more interesting for theoretical work than for applications. Additionally, one can likely incorporate priors on the values of g and ω as regularizers in the surface-tension energy. Another viable extension is to incorporate a small amount of supervision into the community-inference process using techniques (such as quadratic fidelity terms) from image processing. A similar idea

[10] For this example, we have applied the change of variables from section A to eliminate the diagonal elements.

	PP	LFR	MS	Caltech	Princeton	Penn. State	Plume
MCF	5.36	17.71	3.47	1.39	1.46	38.91	77.91
AC	5.37	26.27	7.28	8.84	480.4	3853	268.7
MBO	4.27	11.05	1.73	0.67	7.43	382.31	270.0
KL	16,566	176	5,117	20	662	95,603	980,520

TABLE 2
Computation times (in seconds).

0	5.22	∞	∞	∞	∞	∞	∞	∞	∞
5.22	0	6.1817	∞	∞	∞	∞	∞	∞	∞
∞	6.1817	0	6.8471	∞	∞	∞	∞	∞	∞
∞	∞	6.8471	0	7.6316	∞	∞	∞	∞	∞
∞	∞	∞	7.6316	0	8.362	∞	∞	∞	∞
∞	∞	∞	∞	8.362	0	9.0869	∞	∞	∞
∞	∞	∞	∞	∞	9.0869	0	9.7926	∞	∞
∞	∞	∞	∞	∞	∞	9.7926	0	10.4911	∞
∞	∞	∞	∞	∞	∞	∞	10.4911	0	11.1869
∞	∞	∞	∞	∞	∞	∞	∞	11.1869	0

TABLE 3
Optimal surface tensions for the MS SBM example. (Note that the entries are heterogeneous, so there are different surface tensions between different pairs of communities.) The infinite entries correspond to sets with no observed edge between them.

was suggested for modularity maximization in [38] and was tested in [10]. The introduction of supervision helps alleviate severe nonconvexity by penalizing local minima that do not agree with the supervision. It is also important to generalize our approach to more complicated types of networks, such as multilayer [46] and temporal networks [37], and to incorporate metadata [63] into our inference methodology. For example, given our successful results on the hyperspectral video, it may be particularly interesting to use temporal-network clustering to analyze time-dependent communities in the video.

Approaches such as inference using SBMs and modularity maximization are also related to other approaches for community detection, and the results in this paper may help further illuminate those connections. These include recent work that relates SBMs to local methods for community detection that are based on personalized PageRank [47] and very recent work that established new connections between modularity maximization and several other approaches [89]. We expect that further mapping of the relations between the diverse available perspectives for community detection (and other problems in network clustering) will yield many new insights for network theory, algorithms, and applications.

Acknowledgements. ZMB and ALB were funded by NSF grants DMS-1737770 and DMS-1417674, as well as ONR grant N00014-16-1-2119. ZMB was also supported by the Department of Defense (DoD) through the National Defense Science & Engineering Graduate Fellowship (NDSEG) Program. MAP and ALB were also funded by DARPA award number FA8750-18-2-0066. We thank Kevin Miller, Brent Edmunds, and Robert Hannah for helpful discussions.

A. Eliminating the Diagonal Elements of W. It is difficult to determine the parameters $W_{\alpha\alpha}$ in the context of (6) and our surface-tension analogy, because they correspond to "internal" surface tensions of a single crystal. In this appendix, we use a change of variables to eliminate these diagonal terms and replace them with additional volume terms, which are much easier to interpret.

We begin with the identity

$$\sum_{\alpha,\beta} W_{\alpha\beta}\mathrm{Cut}_{g,A}(\alpha,\beta) = \sum_{\alpha}\sum_{\beta\neq\alpha} W_{\alpha\beta}\mathrm{Cut}_{g,A}(\alpha,\beta) + \sum_{\alpha} W_{\alpha\alpha}\mathrm{Cut}_{g,A}(\alpha,\alpha)\,, \tag{15}$$

and we compute

$$\sum_{\alpha} W_{\alpha\alpha}\mathrm{Cut}_{g,A}(\alpha,\alpha) = \sum_{\alpha} W_{\alpha,\alpha} \sum_{g_i=\alpha, g_j=\alpha} w_{ij}$$

$$
\begin{aligned}
&= \sum_{\alpha} W_{\alpha,\alpha} \left(\sum_{g_i=\alpha, j=1,\dots,N} w_{ij} - \sum_{g_i=\alpha, g_j\neq\alpha} w_{ij} \right) \\
&= \sum_{\alpha} W_{\alpha,\alpha} \left(\sum_{g_i=\alpha} k_i - \sum_{\beta\neq\alpha, g_i=\alpha, g_j=\beta} w_{ij} \right) \\
&= \sum_{\alpha} W_{\alpha,\alpha} \left(\mathrm{vol}_{g,A}(\alpha) - \sum_{\beta\neq\alpha} \mathrm{Cut}_{g,A}(\alpha,\beta) \right) . \qquad (16)
\end{aligned}
$$

Combining (16) with (15) yields

$$
\sum_{\alpha,\beta} W_{\alpha\beta} \mathrm{Cut}_{g,A}(\alpha,\beta) = \sum_{\alpha\neq\beta} (W_{\alpha\beta} - W_{\alpha\alpha}) \, \mathrm{Cut}_{g,A}(\alpha,\beta) + \sum_{\alpha} W_{\alpha\alpha} \mathrm{vol}_{g,A}(\alpha) \, . \qquad (17)
$$

This formulation has eliminated the diagonal at the cost of making W asymmetric. We can fix this issue by replacing (17) with

$$
\begin{aligned}
\sum_{\alpha,\beta} W_{\alpha\beta} \mathrm{Cut}_{g,A}(\alpha,\beta) &= \sum_{\alpha\neq\beta} \left(W_{\alpha\beta} - \frac{1}{2} W_{\alpha\alpha} - \frac{1}{2} W_{\beta\beta} \right) \mathrm{Cut}_{g,A}(\alpha,\beta) + \sum_{\alpha} W_{\alpha\alpha} \mathrm{vol}_{g,A}(\alpha) \\
&= \sum_{\alpha\neq\beta} \hat{\sigma}_{\alpha\beta} \mathrm{Cut}_{g,A}(\alpha,\beta) + \sum_{\alpha} W_{\alpha\alpha} \mathrm{vol}_{g,A}(\alpha) \, , \qquad (18)
\end{aligned}
$$

where $\hat{\sigma}_{\alpha\beta} = W_{\alpha\beta} - \frac{1}{2} W_{\alpha\alpha} - \frac{1}{2} W_{\beta\beta}$. The matrix $\hat{\sigma}$ is symmetric and has 0 values on the diagonal.

Finally, we expand a bit on the role of the volume terms in (6). The term

$$
\sum_{\alpha} W_{\alpha\alpha} \mathrm{vol}_{g,A}(\alpha) \qquad (19)
$$

is the inner product of the vector of volumes with the diagonal of W. We minimize (19), subject to the constraints $\sum_{\alpha} \mathrm{vol}_{g,A}(\alpha) = 2m$ and $\mathrm{vol}_{g,A}(\alpha) \geq 0$, by placing all of the nodes in the community that corresponds to the smallest[11] entry in the diagonal of W. Thus, these terms incentivize placing more mass in the communities with the smallest volume penalty.

B. Γ-Convergence of the Ginzburg–Landau Approximation of (6). Gamma-convergence is defined as follows:

DEFINITION B.1. *Let Y be a metric space, and let F_n be a sequence of functionals that take values in $\mathbb{R} \cup \{\infty\} \cup \{-\infty\}$. We say that F_n* Γ-converges *to another functional F if for all $x \in Y$, the following bounds hold:*

1. *(Lower bound) For every sequence $x_n \to x$, we have $F(x) \leq \liminf_{n\to\infty} F_n(x_n)$.*
2. *(Upper bound) For every $x \in Y$, there is a sequence $x_n \to x$ such that $F(x) \geq \limsup_{n\to\infty} F_n(x_n)$.*

We now prove Theorem 4.1.

Proof. We largely follow [86], though we generalize to account for the multiphase nature of our problem.

Observe that all of the terms that do not involve the potential T are continuous and independent of ϵ, so they cannot interfere with the Γ-convergence [20]. Therefore, it suffices to prove that $\frac{1}{\epsilon} T$ Γ-converges to

$$
\chi(U) = \begin{cases} 0 \, , & \text{if } U \text{ corresponds to a partition} \, , \\ +\infty \, , & \text{otherwise} \, . \end{cases}
$$

[11] When referring to "smallest" eigenvalues in the appendices, we mean the smallest positive or most-negative values rather than those that are smallest in magnitude.

To prove the lower bound, let $U_n \to U$ and $\epsilon_n \to 0$. If U corresponds to a partition, then $\chi(U) = 0$, which is automatically less than or equal to $\frac{1}{\epsilon_n} T(U_n)$ for each n. If U does not correspond to a partition, then $\chi(U) = +\infty$. Pick N_1 such that whenever $n > N_1$, the distance from U_n to the nearest feasible point is at least $c > 0$. Let T_c be the infimum of T on all of $\mathbb{R}^{N \times \hat{n}}$ except for the balls of radius c that surround each feasible point (so, in particular, $T_0 > 0$). It follows that $\lim\inf_{n\to\infty} \frac{1}{\epsilon_n} T(U_n) \geq \lim_{n\to\infty} \frac{1}{\epsilon_n} T_0 = +\infty$. Thus, the lower bound always holds.

To prove the upper bound, let U be any $N \times \hat{n}$ matrix. If U corresponds to a partition, then letting $U_n = U$ for all n gives the required sequence. If u does not correspond to a partition, then $U_n = U$ for all n still satisfies the upper bound.

Thus, both the upper and lower bound requirements hold, and we have proven Γ-convergence. ∎

C. Additional Notes on the AC and MBO Schemes. In this appendix, we discuss some practical details regarding our implementation of the AC and MBO solvers.

The choice of ϵ in AC is important, because it selects a characteristic scale of the transition. If it is too small, the barrier to transition is large, and no evolution occurs. If it is too large, the transition layer becomes wide enough that a large part of the graph is caught in it, such that U does not approximately correspond to a partition of the graph. Furthermore, Theorem 4.1 asserts only that the minimizers of (6) and (13) are related when ϵ is sufficiently small. In our numerical experiments, we set $\epsilon = 0.004$, a choice that we selected by hand-tuning using our synthetic networks. There is no reason to believe that the same value should work for all networks. For example, for the well-known Zachary Karate Club network [92], we obtain much better results for $\epsilon = 0.04$. A very interesting problem is to determine a correct notion of distance and accompanying quantitative estimates to allow an automated selection of ϵ to obtain a transition layer with an appropriate width to give useful results. We discretize the AC equation via convex splitting [22]:

$$(1 + c\,dt)U^{n+1} + LU^{n+1}W = -dt\left(cU^n + T'(U^n) + \frac{1}{2m} kk^T U e^{-W}\right),$$

where $c > 2/\epsilon$ [50]. Using the constant c leads to an unconditionally stable scheme, which negates the stiffness caused by the $1/\epsilon$ scale.

It is necessary to solve linear system of the form

$$(1 + c\,dt)U^{n+1} + LU^{n+1}W = F^n \tag{20}$$

many times. In a continuum setting, one can use a fast Fourier transform, but we do not know of a graph analog with comparable computational efficiency. Instead, we find the $2\hat{n}$ eigenvectors that correspond to the smallest eigenvalues[12] of L and the entire spectrum of W. Consequently, $L \approx V_L D_L V_L^T$ and $W = V_W D_W V_W^T$, and the system (20) is appromately equivalent to

$$(1 + c\,dt)V_L^T U^{n+1} V_W + D_L V_L^T U^{n+1} V_W D_W = V_L^T F^n V_W\,.$$

Letting $\hat{U}^n = V_L^T U^n V_W$ and $\hat{F}^n = V_L^T F^n V_W$, we write

$$(1 + c\,dt)\hat{U}^{n+1} + D_L \hat{U}^{n+1} D_W = \hat{F}^n\,, \tag{21}$$

which is easy to solve for $\hat{U}^{n+1}$. We convert $\hat{U}^{n+1}$ to a solution using $U^{n+1} = V_L \hat{U}^{N+1} V_W^T$. (See [6] for a discussion of this method of recovering U^{n+1} from $\hat{U}^{n+1}$.)

One final detail that we wish to note is that we want the evolution of U to be restricted to have a row sum of 1, so that we can interpret it in terms of probabilities. To do this, we use a vectorized version of the projection algorithm from [16] at each time step.

The MBO solver uses a very similar pseudospectral scheme, although it does not include convex splitting. Unlike in the AC scheme, we need to estimate two time steps automatically in our code, instead of tuning them by hand. The first is the inner-loop step, which we determined using a restriction (which one can

[12] The number $2\hat{n}$ is somewhat arbitrary; we choose it to exceed $\hat{n}$, but for computational convenience, we do not want it to be too large.

show is necessary for stability[13]) that the time step should not exceed twice the reciprocal of the largest eigenvalue of the linear operator that maps U to $\frac{1}{m}kk^TUe^{-W}$. The time step between thresholdings of U is given by the reciprocal of the geometric mean of the largest and smallest eigenvalues of the operator that maps $U \to LUW$. The associated intuition is that linear diffusion should have enough time to evolve (to avoid getting stuck) but not enough time to evolve to equilibrium (because the equilibrium does not depend on the initial condition, so it carries no information about it). The reciprocal of the smallest eigenvalue gives an estimate of the time that it takes to reach equilibrium, and the reciprocal of the largest eigenvalue gives an estimate of the fastest evolution of the system. We choose the geometric mean between these values to produce a number between these two extremes.[14] References [10] and [87] proved bounds (although in a simpler setting) that support these time-step choices for MBO schemes.

REFERENCES

[1] A. A. Amini, A. Chen, P. J. Bickel, and E. Levina, *Pseudo-likelihood methods for community detection in large sparse networks*, Ann. Statist., 41 (2013), pp. 2097–2122.

[2] A. Arenas, A. Fernández, and S. Gómez, *Analysis of the structure of complex networks at different resolution levels*, New J. Phys., 10 (2008), 053039.

[3] N. W. Ashcroft and N. D. Mermin, *Solid State Physics*, Brooks Cole, Pacific Grove, CA, 1st ed., 1976.

[4] M. Ayati, S. Erten, M. R. Chance, and M. Koyuturk, *Mobas: Identification of disease-associated protien subnets using modularity-based scoring*, EURASIP J. Bioinf. Sys. Bio., 1 (2015), pp. 1–14.

[5] D. S. Bassett, E. T. Owens, M. A. Porter, M. L. Manning, and K. E. Daniels, *Extraction of force-chain network architecture in granular materials using community detection*, Soft Matter, 11 (2015), pp. 2731–2744.

[6] A. L. Bertozzi and A. Flenner, *Diffuse interface models on graphs for classification of high dimensional data*, Multiscale Model. Simul., 10 (2012), pp. 1090–1118.

[7] A. L. Bertozzi and A. Flenner, *Diffuse interface models on graphs for classification of high dimensional data*, SIAM Rev., 58 (2016), pp. 293–328.

[8] R. F. Betzel and D. S. Bassett, *Multi-scale brain networks*, NeuroImage, 160 (2017), pp. 73–83.

[9] W. J. Boettinger, J. A. Warren, C. Beckermann, and A. Karma, *Phase-field simulation of solidification*, Ann. Rev. Mater. Res., 32 (2002), pp. 163–194.

[10] Z. Boyd, E. Bae, X.-C. Tai, and A. L. Bertozzi, *Simplified energy landscape for modularity using total variation*, arXiv:1707.09285, (2017).

[11] Y. Boykov and V. Kolmogorov, *Computing geodesics and minimal surfaces via graph cuts*, in Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2, ICCV '03, Washington, DC, USA, 2003, IEEE Computer Society, pp. 26–33.

[12] Y. Boykov, V. Kolmogorov, D. Cremers, and A. Delong, *An integral solution to surface evolution PDEs via geo-cuts*, in Computer Vision — ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7–13, 2006, Proceedings, Part III, A. Leonardis, H. Bischof, and A. Pinz, eds., Springer-Verlag, Berlin, Germany, 2006, pp. 409–422.

[13] A. Buades, B. Coll, and J. M. Morel, *A non-local algorithm for image denoising*, in Computer Vision and Pattern Recognition, vol. 2, June 2005, pp. 60–65.

[14] E. J. Candès, J. Romberg, and T. Tao, *Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information*, IEEE Trans. Inform. Theory, 52 (2006), pp. 489–509.

[15] Cenna, *Gr3gr.gif*. Wikimedia Commons https://commons.wikimedia.org/wiki/File:Grgr3d_small.gif.

[16] Y. Chen and X. Ye, *Projection onto a simplex*, arXiv:1101.6081, (2011).

[17] F. Cleri, S. R. Phillpot, and D. Wolf, *Atomistic simulations of intergranular fracture in symmetric-tilt grain boundaries*, Interface Sci., 7 (1999), pp. 45–55.

[18] A. Condon and R. M. Karp, *Algorithms for graph partitioning on the planted partition model*, Random Structures & Algorithms, 18 (2001), pp. 116–140.

[19] P. Csermely, A. London, L.-Y. Wu, and B. Uzzi, *Structure and dynamics of core–periphery networks*, J. Complex Netw., 1 (2013), pp. 93–123.

[20] G. Dal Maso, *An Introduction to Gamma-Convergence*, Birkhauser, Boston, 1993.

[21] S. Esedoglu and F. Otto, *Threshold dynamics for networks with arbitrary surface tensions*, Comm. Pure Appl. Math., 68 (2015), pp. 808–864.

[22] D. J. Eyre, *An unconditionally stable one-step scheme for gradient systems*. https://www.math.utah.edu/~eyre/research/methods/stable.ps, 1998.

[23] S. E. Fienberg and S. S. Wasserman, *Categorical data analysis of single sociometric relations*, Sociol. Meth., 12 (1981), pp. 156–192.

[24] S. Fortunato and M. Barthélemy, *Resolution limit in community detection*, Proc. Natl. Acad. Sci. USA, 104 (2007), pp. 36–41.

[13] See, e.g., [49] for the necessary techniques, which are standard in the numerical analysis of ordinary differential equations.

[14] From experimentation, we concluded that it is better to multiply this time step by 8 to avoid getting stuck too early. We started with the geometric mean, because eigenvalues can have very different orders of magnitudes.

[25] S. Fortunato and D. Hric, *Community detection in networks: A user guide*, Phys. Rep., 659 (2016), pp. 1–44.
[26] B. K. Fosdick, D. B. Larremore, J. Nishimura, and J. Ugander, *Configuring random graph models with fixed degree sequences*, arXiv:1608.00607 (SIAM Review, in press), (2016).
[27] O. Frank and F. Harary, *Cluster inference by using transitivity indices in empirical graphs*, J. Am. Stat. Soc., 77 (1982), pp. 835–840.
[28] H. J. Frost, C. V. Thompson, and D. T. Walton, *Simulation of thin film grain structures:I Grain growth stagnation*, Acta Metall. Mater., 38 (1990), pp. 1455–1462.
[29] C. Garcia-Cardona, E. Merkurjev, A. L. Bertozzi, A. Percus, and A. Flenner, *Multiclass segmentation using the Ginzburg-Landau functional and the MBO scheme*, IEEE Trans. Pattern Anal. Mach. Intell., 36 (2014), pp. 1600–1614.
[30] T. Gerhart, J. Sunu, L. Lieu, E. Merkurjev, J.-M. Chang, J. Gilles, and A. L. Bertozzi, *Detection and tracking of gas plumes in LWIR hyperspectral video sequence data*, in SPIE Defense, Security, and Sensing, International Society for Optics and Photonics, 2013, pp. 87430J–87430J.
[31] G. Gilboa and S. Osher, *Nonlocal operators with applications to image processing*, Multiscale Model. Simul., 7 (2008), pp. 1005–1028.
[32] T. Goldstein and S. Osher, *The split Bregman method for L1-regularized problems*, SIAM J. Imaging Sci., 2 (2009), pp. 323–343.
[33] B. H. Good, Y.-A. de Montjoye, and A. Clauset, *Performance of modularity maximization in practical contexts*, Phys. Rev. E, 81 (2010), 046106.
[34] R. A. Hegemann, L. M. Smith, A. B. Barbaro, A. L. Bertozzi, S. E. Reid, and G. E. Tita, *Geographical influences of an emerging network of gang rivalries*, Physica A, 390 (2011), pp. 3894–3914.
[35] P. W. Holland, K. B. Laskey, and S. Leinhardt, *Stochastic blockmodels: First steps*, Social Netw., 5 (1983), pp. 109–137.
[36] P. Holme, *Modern temporal network theory: A colloquium*, Eur. Phys. J. B, 88 (2015), 234.
[37] P. Holme and J. Saramäki, *Temporal networks*, Phys. Rep., 519 (2012), pp. 97–125.
[38] H. Hu, T. Laurent, M. A. Porter, and A. L. Bertozzi, *A method based on total variation for network modularity optimization using the MBO scheme*, SIAM J. Appl. Math., 73 (2013), pp. 2224–2246.
[39] M. Jacobs, *Algorithms for multiphase partitioning*, PhD thesis, University of Michigan, Ann Arbor, 2017.
[40] M. Jacobs, E. Merkurjev, and S. Esedoglu, *Auction dynamics: A volume-constrained MBO scheme*, J. Comp. Phys., 354 (2018), pp. 288–310.
[41] L. G. S. Jeub, P. Balachandran, M. A. Porter, P. J. Mucha, and M. W. Mahoney, *Think locally, act locally: The detection of small, medium-sized, and large communities in large networks*, Phys. Rev. E, 91 (2015), 012821.
[42] B. Karrer and M. E. J. Newman, *Stochastic blockmodels and community structure in networks*, Phys. Rev. E, 83 (2011), 016107.
[43] B. W. Kernighan and S. Lin, *An efficient heuristic procedure for partitioning graphs*, Bell Syst. Tech. J., 49 (1970), pp. 291–307.
[44] M. Kim and J. Leskovec, *Inferring missing nodes and edges in networks*, in Proceedings of the 2011 SIAM International Conference on Data Mining, N. Chawla and W. Wang, eds., 2011, pp. 47–58.
[45] D. Kinderlehrer, I. Livshits, and S. Ta'asan, *A variational approach to modeling and simulation of grain growth*, SIAM J. Sci. Comput., 28 (2006), pp. 1694–1715.
[46] M. Kivelä, A. Arenas, M. Barthélemy, J. P. Gleeson, Y. Moreno, and M. A. Porter, *Multilayer networks*, J. Complex Netw., 2 (2014), pp. 203–271.
[47] I. M. Kloumann, J. Ugander, and J. Kleinberg, *Block models and personalized PageRank*, Proc. Nat. Acad. Sci. USA, 114 (2017), pp. 33–38.
[48] A. Lancichinetti, S. Fortunato, and F. Radicchi, *Benchmark graphs for testing community detection algorithms*, Phys. Rev. E., 78 (2008), 056117.
[49] R. J. LeVeque, *Finite difference methods for differential equations*. Available at https://pdfs.semanticscholar.org/8ec6/d5e07121fb25213657d89c3bfb523e1e4721.pdf.
[50] X. Luo and A. L. Bertozzi, *Convergence of the graph Allen–Cahn scheme*, J. Stat. Phys., 167 (2017), pp. 934–958.
[51] D. Manolakis, C. Siracusa, and G. Shaw, *Adaptive matched subspace detectors for hyperspectral imaging applications*, in 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 5, 2001, pp. 3153–3156.
[52] C. Mantegazza, *Lecture Notes on Mean Curvature Flow*, Springer-Verlag, Berlin, Germany, 2011.
[53] Z. Meng, E. Merkurjev, A. Koniges, and A. L. Bertozzi, *Hyperspectral image classification using graph clustering methods*, IPOL J. Image Process. Online, 7 (2017), pp. 218–245.
[54] E. Merkurjev, E. Bae, A. L. Bertozzi, and X.-C. Tai, *Global binary optimization on graphs for data segmentation*, J. Math. Imaging Vision, 52 (2015), pp. 414–435.
[55] E. Merkurjev, J. Sunu, and A. L. Bertozzi, *Graph MBO method for multiclass segmentation of hyperspectral stand-off detection video*, in IEEE International Conference on Image Processing, 2014.
[56] B. Merriman, J. Bence, and S. Osher, *Diffusion generated motion by mean curvature*, Proc. Comput. Crystal Growers Workshop, (1992), pp. 73–83.
[57] L. Modica, *The gradient theory of phase transitions and the minimal interface criterion*, Archive for Rational Mechanics and Analysis, 98 (1987), pp. 123–142.
[58] C. Moore, *The computer science and physics of community detection: Landscapes, phase transitions, and hardness*, arXiv:1702.00467, (2017).
[59] W. W. Mullins, *Two-dimensional motion of idealized grain boundaries*, J. Appl. Phys., 27 (1956), pp. 900–904.
[60] M. E. J. Newman, *Finding community structure in networks using the eigenvectors of matrices*, Phys. Rev. E, 74 (2006), p. 036104.
[61] M. E. J. Newman, *Networks: an Introduction*, Oxford University Press, 2010.
[62] M. E. J. Newman, *Equivalence between modularity optimization and maximum likelihood methods for community detec-*

tion, Phys. Rev. E, 94 (2016), 052315.
[63] M. E. J. Newman and A. Clauset, *Structure and inference in annotated networks*, Nature Commun., 7 (2016), 11863.
[64] M. E. J. Newman and M. Girvan, *Finding and evaluating community structure in networks*, Phys. Rev. E, 69 (2004).
[65] M. E. J. Newman and G. Reinert, *Estimating the number of communities in a network*, Phys. Rev. Lett., 117 (2016), 078301.
[66] S. Osher and J. A. Sethian, *Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton–Jacobi formulations*, J. Comput. Phys., 79 (1988), pp. 12–49.
[67] N. Otter, M. A. Porter, U. Tillmann, P. Grindrod, and H. A. Harrington, *A roadmap for the computation of persistent homology*, EPJ Data Sci., 6 (2017), pp. 1–38.
[68] L. Papadopoulos, M. A. Porter, K. E. Daniels, and D. S. Bassett, *Network analysis of particles and grains*, J. Complex Netw., in press (arXiv:1708.08080) (2018).
[69] R. Pastor-Satorras, C. Castellano, P. Van Mieghem, and A. Vespignani, *Epidemic processes in complex networks*, Rev. Mod. Phys., 87 (2015), pp. 925–979.
[70] T. P. Peixoto, *Hierarchical block structures and high-resolution model selection in large networks*, Phys. Rev. X, 4 (2014), 011047.
[71] T. P. Peixoto, *Inferring the mesoscale structure of layered, edge-valued, and time-varying networks*, Phys. Rev. E, 92 (2015), 042807.
[72] T. P. Peixoto, *Model selection and hypothesis testing for large-scale network models with overlapping groups*, Phys. Rev. X, 5 (2015), 011033.
[73] T. P. Peixoto, *Bayesian stochastic blockmodeling*, arXiv:1705.10225, (2018). Chapter in "Advances in Network Clustering and Blockmodeling", edited by P. Doreian, V. Batagelj, A. Ferligoj, (John Wiley & Sons, New York City, USA [forthcoming]).
[74] M. A. Porter, P. J. Mucha, M. E. J. Newman, and C. M. Warmbrand, *A network analysis of committees in the U.S. House of Representatives*, Proc. Nat. Acad. Sci. U.S.A., 102 (2005), pp. 7057–7062.
[75] M. A. Porter, J.-P. Onnela, and P. J. Mucha, *Communities in networks*, Notices Amer. Math. Soc., 56 (2009), pp. 1082–1097, 1164–1166.
[76] M. A. Riolo, G. T. Cantwell, G. Reinert, and M. E. J. Newman, *Efficient method for estimating the number of communities in a network*, Phys. Rev. E, 96 (2017), 032310.
[77] P. Rombach, M. A. Porter, J. H. Fowler, and P. J. Mucha, *Core-periphery structure in networks (revisited)*, SIAM Rev., 59 (2017), pp. 619–646.
[78] R. A. Rossi and N. K. Ahmed, *Role discovery in networks*, IEEE Trans. Knowl. Data Eng., 27 (2015), pp. 1112–1131.
[79] L. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation noise removal algorithm*, Phys. D, 60 (1992), pp. 259–268.
[80] C. S. Smith, *Metal Interfaces*, American Society for Metals, Cleveland, 1952, ch. Grain shapes and other metallurgical applications of topology, pp. 65–113.
[81] T. A. B. Snijders and K. Nowicki, *Estimation and prediction for stochastic blockmodels for graphs with latent block structure*, J. Classif., 14 (1997), pp. 75–100.
[82] A. L. Traud, E. D. Kelsic, P. J. Mucha, and M. A. Porter, *Comparing community structure to characteristics in online collegiate social networks*, SIAM Rev., 53 (2011), pp. 526–543.
[83] A. L. Traud, P. J. Mucha, and M. A. Porter, *Social structure of Facebook networks*, Phy. A, 391 (2012), pp. 4165–4180.
[84] N. G. Trillos, D. Slepcev, J. Von Brecht, T. Laurent, and X. Bresson, *Consistency of Cheeger and ratio graph cuts*, J. Mach. Learn. Res., 17 (2016), pp. 1–46.
[85] F. Tudisco, P. Mercado, and M. Hein, *Community detection in networks via nonlinear modularity eigenvectors*, arXiv:1708.05569, (2017).
[86] Y. van Gennip and A. L. Bertozzi, *Gamma-convergence of graph Ginzburg-Landau functionals*, Adv. Differential Equations, 17 (2012), pp. 1115–1180.
[87] 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.
[88] A. Vedaldi and B. Fulkerson, *VLFeat: An open and portable library of computer vision algorithms*. http://www.vlfeat.org, 2008.
[89] N. Veldt, D. F. Gleich, and A. Wirth, *Unifying sparsest cut, cluster deletion, and modularity clustering objectives with correlation clustering*, arXiv:1712.05825, (2017).
[90] U. von Luxborg, *A tutorial on spectral clustering*, Statist. Comput., 17 (2007), pp. 395–416.
[91] D. Weaire and J. P. Kermode, *Computer simulation of a two-dimensional soap froth I: Method and motivation*, Phil. Mag. B, 48 (1983), pp. 245–259.
[92] W. W. Zachary, *An information flow model for conflict and fission in small groups*, Journal of Anthropological Research, 33 (1977), pp. 452–473.
[93] W. Zhu, V. Chayes, A. Tiard, S. Sanchez, D. Dahlberg, A. L. Bertozzi, S. Osher, D. Zosso, and D. Kuang, *Unsupervised classification in hyperspectral imagery with nonlocal total variation and primal-dual hybrid gradient algorithm*, IEEE Trans. Geosci. Remote Sensing, 55 (2017), pp. 2786–2798.