Robust Recovery for Stochastic Block Models, Simplified and Generalized

Sidhanth Mohanty
sidhanth@csail.mit.edu
MIT
United States

Prasad Raghavendra
raghavendra@berkeley.edu
UC Berkeley
United States

David X. Wu
david_wu@berkeley.edu
UC Berkeley
United States

ABSTRACT

We study the problem of *robust community recovery*: efficiently recovering communities in sparse stochastic block models in the presence of adversarial corruptions. In the absence of adversarial corruptions, there are efficient algorithms when the *signal-to-noise ratio* exceeds the *Kesten–Stigum (KS) threshold*, widely believed to be the computational threshold for this problem. The question we study is: *does the computational threshold for robust community recovery also lie at the KS threshold?* We answer this question affirmatively, providing an algorithm for robust community recovery for arbitrary stochastic block models on any constant number of communities, generalizing the work of Ding, d'Orsi, Nasser & Steurer on an efficient algorithm above the KS threshold in the case of 2-community block models.

There are three main ingredients to our work:

(1) The Bethe Hessian of the graph is defined as $H_G(t) \triangleq (D_G - I)t^2 - A_G t + I$ where D_G is the diagonal matrix of degrees and A_G is the adjacency matrix. Empirical work suggested that the Bethe Hessian for the stochastic block model has outlier eigenvectors corresponding to the communities right above the Kesten-Stigum threshold. We formally confirm the existence of outlier eigenvalues for the Bethe Hessian, by explicitly constructing outlier eigenvectors from the community vectors.

(2) We develop an algorithm for a variant of robust PCA on sparse matrices. Specifically, an algorithm to partially recover top eigenspaces from adversarially corrupted sparse matrices under mild delocalization constraints.

(3) A rounding algorithm to turn vector assignments of vertices into a community assignment, inspired by the algorithm of Charikar & Wirth for 2XOR.

CCS CONCEPTS

• **Theory of computation → Random network models**.

KEYWORDS

robust inference, stochastic block model, community detection, spectral algorithms

STOC '24, June 24–28, 2024, Vancouver, BC, Canada

ACM ISBN 979-8-4007-0383-6/24/06
https://doi.org/10.1145/3618260.3649761

ACM Reference Format:
Sidhanth Mohanty, Prasad Raghavendra, and David X. Wu. 2024. Robust Recovery for Stochastic Block Models, Simplified and Generalized. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing (STOC '24), June 24–28, 2024, Vancouver, BC, Canada.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3618260.3649761

1 INTRODUCTION

The stochastic block model (SBM) has provided an enlightening lens into understanding a wide range of computational phenomena in Bayesian inference problems, such as computational phase transitions & information-computation gaps [11, 20, 26, 31], spectral methods for sparse matrices [6, 24, 32], local message-passing algorithms [17, 30, 34], and robustness [3, 13, 25, 28].

The SBM is a model of random graphs where the vertices are partitioned into *communities*, denoted by $\boldsymbol{x}$, and the probability of an edge existing is contingent on the communities that the two endpoints are part of. The algorithmic task is the *community recovery* problem: given an input graph $\boldsymbol{G}$, estimate the posterior $\boldsymbol{x}|\boldsymbol{G}$ with an efficient algorithm.

Definition 1.1 (Informal)**.** In the *stochastic block model*, we are given a $k \times k$ matrix M, a distribution π over $[k]$, and $d > 0$, and $\mathrm{SBM}_n(\mathrm{M}, \pi, d)$ denotes the distribution where an n-vertex graph $\boldsymbol{G}$ is sampled by:

(1) drawing a *color* $\boldsymbol{x}(u) \sim \pi$ for every $u \in [n]$,
(2) for each pair of vertices u, v, the edge $\{u, v\}$ is chosen with probability $\mathrm{M}_{\boldsymbol{x}(u),\boldsymbol{x}(v)} \cdot \frac{d}{n}$.

In the *community recovery* problem, the goal is to give an efficient algorithm that takes $\boldsymbol{G}$ as input and outputs a community assignment $\widehat{x}$ approximating $\boldsymbol{x}|\boldsymbol{G}$ (see Definition 4.4 for a formal definition).

Computational Thresholds. For a given M and π, increasing d can only possibly make the problem easier. The main question is to understand the *computational threshold* for community recovery — i.e. the minimum value of d where the problem goes from being intractable to admitting efficient algorithms.

The first predictions for this computational threshold came from the *cavity method* in statistical physics in the work of Decelle, Krzakala, Moore & Zdeborova [11]. They posited that the location of this transition is at the *Kesten–Stigum threshold* (henceforth KS threshold), a threshold for broadcast processes on trees studied in the works of Kesten & Stigum [22, 23]. The algorithmic side of these predictions was confirmed in the case of the 2-community block model in the works of Mossel, Neeman & Sly [31] and Massoulié [26], and then for block models in increasing levels of generality by Bordenave, Lelarge & Massoulié [6], Abbe & Sandon [2], and Hopkins & Steurer [20].

Robust Algorithms. All of these algorithms utilize the knowledge of the distribution the input is sampled from quite strongly — they are based on $\Omega(\log n)$-length walk statistics in the stochastic block model. However, the full generative process in inference is not always known precisely. Thus, we would like algorithms that utilize but do not overfit to the distributional assumptions.

Demanding that our algorithm be *robust*, i.e. resilient to adversarial corruptions to the input, is often a useful way to design algorithms that are less sensitive to distributional assumptions. This leads one to wonder: can algorithms that don't strongly exploit the prior distribution approach the KS threshold?

Optimization vs. Inference. Earlier approaches to robust recovery in 2-community block models were based on *optimization*: semidefinite programming relaxations of the minimum bisection problem, as in the work of Guedon & Vershynin [18]. These approaches have the advantage of being naturally robust, since the algorithms are approximately Lipschitz around random inputs, but the minimum bisection relaxation is not known to achieve statistical optimality and only succeeds well above the KS threshold.

The following two results point to the suboptimality of optimization-based strategies. Moitra, Perry & Wein [28] considered the *monotone* adversary in the 2-community setting, where the adversary is allowed to make an *unbounded* number of edge insertions within communities and edge deletions across communities. At an intuitive level, this is supposed to only make the problem easier and indeed does so for the minimum bisection approach, but to the contrary [28] proves that the threshold for recovery *increases*. Dembo, Montanari & Sen [12] exactly nailed the size of the minimum bisection in Erdős–Rényi graphs, which are complete noise and have no signal in the form of a planted bisection — and strikingly, it is actually *smaller* than the size of the planted bisection in the detectable regime! Thus, it is conceivable that there are bisections completely orthogonal to the planted bisection in a stochastic block model graph that nevertheless have the same size.

The problem of recovering communities is more related to the task of *Bayesian inference*, i.e., applying Bayes' rule and approximating $\boldsymbol{x}|\boldsymbol{G}$. Optimizing for the minimum bisection is akin to computing the *maximum likelihood estimate*, which does not necessarily produce samples representative of the posterior distribution of $\boldsymbol{x}|\boldsymbol{G}$.

SDPs for Inference. The work of Banks, Mohanty & Raghavendra [3] proposed a semidefinite programming-based algorithm for inference tasks that incorporates the prior distribution in the formulation, and illustrated that this algorithm can distinguish between $\boldsymbol{G}$ sampled from the stochastic block model from an Erdős–Rényi graph of equal average degree anywhere above the KS threshold while being resilient to $\Omega(n)$ arbitrary edge insertions and deletions.

A similar SDP formulation was later studied by Ding, d'Orsi, Nasser & Steurer [13] in the 2-community setting, and was used to give an algorithm to recover the communities with a constant advantage over random guessing in the presence of $\Omega(n)$ edge corruptions for all degrees above the KS threshold. They analyze the spectra of matrices associated with random graphs after deleting vertices with large neighborhoods, which introduces unfriendly correlations, and causes their analysis to be highly technical.

The main contribution of our work is an algorithm for robust recovery, which is amenable to a significantly simpler analysis. Our algorithm also succeeds at the recovery task for arbitrary block models with a constant number of communities.

Theorem 1.2 (Informal statement of main theorem). *Let (M, π, d) be SBM parameters such that d is above the KS threshold, and let $\boldsymbol{G}, \boldsymbol{x} \sim \mathrm{SBM}_n(\mathrm{M}, \pi, d)$. There exists $\delta = \delta(\mathrm{M}, \pi, d) > 0$ such that the following holds. There is a polynomial time algorithm that takes as input any graph $\widetilde{\boldsymbol{G}}$ that can be obtained by performing arbitrary δn edge insertions and deletions to $\boldsymbol{G}$ and outputs a coloring $\widehat{\boldsymbol{x}}$ that has "constant correlation" with $\boldsymbol{x}$, with high probability over the randomness of $\boldsymbol{G}$ and $\boldsymbol{x}$.*

Many of the ingredients in the above result are of independent interest. First, we exhibit a symmetric matrix closely related to the Bethe Hessian of the graph, such that its bottom eigenspace is correlated with the communities. Next, we design an efficient algorithm to robustly recover the bottom-r eigenspace of a *sparse* matrix in the presence of adversarial corruptions. Finally, we demonstrate a general rounding scheme to obtain community assignments from this eigenspace.

Remark 1.3 (Robustness against node corruptions)**.** The node corruption model, introduced by Liu & Moitra [25], is a harsher generalization of the edge corruption model. In recent work, Ding, d'Orsi, Hua & Steurer [14] proved that in the setting of sparse SBM, any algorithm that is robust to edge corruptions can be turned into one robust to node corruptions in a blackbox manner. Hence, our results apply in this harsher setting too.

1.1 Related Work

We refer the reader to the survey of Abbe [1] for a detailed treatment of the rich history and literature on community detection in block models, its study in other disciplines, and the many information-theoretic and computational results in various parameter regimes.

Introducing an adversary into the picture provides a beacon towards algorithms that utilize but do not *overfit* to distributional assumptions. Over the years, a variety of adversarial models have been considered, some of which we survey below.

Corruption Models for Stochastic Block Model. Prior to the works of [3, 13], Stefan & Massoulié [33] considered the robust recovery problem, and gave a robust spectral algorithm to recover communities under $O(n^{\varepsilon})$ adversarial edge corruptions for some small enough $\varepsilon > 0$.

Liu & Moitra [25] introduced the *node corruption* model where an adversary gets to perform arbitrary edge corruptions incident to a constant fraction of corrupted vertices, and gave algorithms that achieved optimal accuracy in the presence of node corruptions and the monotone adversary sufficiently above the KS threshold. Soon after, Ding, d'Orsi, Hua & Steurer [14] gave algorithms achieving the Kesten–Stigum threshold using algorithms for the edge corruption model in the low-degree setting [13], and results on the optimization SDP in the high-degree setting [29] in a blackbox manner.

Semirandom & Smoothed Models. Some works have considered algorithm design under harsher adversarial models, where an adversarially chosen input undergoes some random perturbations.

Remarkably, at this point, the best algorithms for several graph and hypergraph problems match the performance of our best algorithms for their completely random counterparts. For example, at this point, the semirandom planted coloring and clique problems were introduced by Blum & Spencer [5], and Feige & Kilian [16], and a line of work [9, 27] culminating in the work of Buhai, Kothari & Steurer [7] showed that the size of the planted clique/coloring recoverable in the semirandom setting matches the famed $\sqrt{n}$ in the fully random setting.

Another example where algorithms for a semirandom version of a block model-like problem have been considered is semirandom CSPs with planted solutions, where the work of Guruswami, Hsieh, Kothari & Manohar [19] gives algorithms matching the guarantees of solving fully random planted CSPs.

1.2 Organization

In Section 2, we give an overview of our algorithm and proof. In Section 3, we give some technical preliminaries. In Section 4, we describe our algorithm and show how to analyze it.

2 TECHNICAL OVERVIEW

An n-vertex graph $\boldsymbol{G}$ is drawn from a stochastic block model and undergoes δn adversarial edge corruptions, and then the corrupted graph $\widetilde{G}$ is given to us as input. For simplicity of discussion, we restrict our attention to *assortative* symmetric k-community block models above the KS threshold, i.e. the connection probability between two vertices i and j only depends on whether they belong to the same community or different communities, and the intra-community probability is higher. Nevertheless, our approach generalizes to any arbitrarily specified k-community block model above the KS threshold.

Let us first informally outline the algorithm; see Section 4 for formal details.

(1) First, we preprocess the corrupted graph $\widetilde{\boldsymbol{G}}$ by truncating high degree vertices, which removes corruptions localized on small sets of vertices in the graph.
(2) We then construct an appropriately defined graph-aware symmetric matrix $M_{\boldsymbol{G}} \in \mathbb{R}^{n\times n}$ whose negative eigenvalues contains information about the true communities for the *uncorrupted* graph. We motivate this construction in Section 2.1.
(3) We recursively trim the rows and columns of $M_{\widetilde{\boldsymbol{G}}}$ to remove small negative eigenvalues in its spectrum. Then we use a spectral algorithm to robustly recover a subspace U which contains information about the communities. Both points are described in Section 2.2.
(4) Finally, we round the subspace U into a community assignment, using a vertex embedding provided by U. This is detailed in Section 2.3.

2.1 Outlier Eigenvectors for the Bethe Hessian

Bordenave, Lelarge & Massoulié [6] analyzed the spectrum of the *nonbacktracking matrix* and rigorously established its connection to community detection. The asymmetric nonbacktracking matrix $B_G \in \{0,1\}^{2|E(G)|\times 2|E(G)|}$ is indexed by directed edges, with

$$(B_G)_{(u_1\to v_1),(u_2\to v_2)} \triangleq \mathbf{1}[v_1 = u_2]\mathbf{1}[v_2 \neq u_1].$$

[6] showed that above the KS threshold, the k outlier eigenvalues for $B_{\boldsymbol{G}}$ correspond to the k community vectors. More precisely, in the case of symmetric k-community stochastic block models above the KS threshold, [6] proved that for the randomly drawn graph $\boldsymbol{G}$, there is a small $\varepsilon > 0$ for which its nonbacktracking matrix $B_{\boldsymbol{G}}$ has exactly k eigenvalues larger than $(1+\varepsilon)\sqrt{d}$ in magnitude.

The *Bethe Hessian* matrix is a symmetric matrix associated with a graph, whose early appearances can be traced to the works of Ihara [21] and Bass [4]. The Bethe Hessian of a graph with parameter $t \in \mathbb{R}$ is defined as

$$H_G(t) \triangleq (D_G - I)t^2 - A_G t + I,$$

where D_G and A_G are the diagonal degree matrix and adjacency matrix of G, respectively. For t in the interval $[0,1]$, it can be interpreted as a regularized version of the standard graph Laplacian. The Bethe Hessian for the stochastic block model was considered in the empirical works [24, 32], where they observed that for some choice of t, the Bethe Hessian and the nonbacktracking matrix has outlier eigenvectors which can be used for finding communities in block models. Concretely, in [32] they observed that for $\boldsymbol{G}$ drawn from stochastic block models above the KS threshold, there is a choice of t such that $H_{\boldsymbol{G}}(t)$ only has a small number of negative eigenvectors, all of which correlate with the hidden community assignment.

We confirm this empirical observation in the following proposition.

Proposition 2.1 (Bethe Hessian spectrum)**.** *Let (M,π,d) be k-community SBM parameters such that d is above the KS threshold, and let $\boldsymbol{G},\boldsymbol{x} \sim \mathrm{SBM}_n(\mathrm{M},\pi,d)$. Then there exists $\varepsilon > 0$ such that for $t^* = \frac{1}{(1+\varepsilon)\sqrt{d}}$, the Bethe Hessian $H_{\boldsymbol{G}}(t^*)$ has at most k negative eigenvalues and at least $k-1$ negative eigenvalues.*

Constructing the outlier eigenspace. There are two assertions in Proposition 2.1. To show that $H_{\boldsymbol{G}}(t^*)$ has at most k negative eigenvalues, one can relate these negative eigenvalues to the k outlier eigenvalues of $B_{\boldsymbol{G}}$ using an Ihara–Bass argument and use a continuity argument as outlined in Fan and Montanari [15, Theorem 5.1].

The more interesting direction is to exhibit at least $k-1$ negative eigenvalues; we will explicitly construct a $k-1$ dimensional subspace starting with the community vectors to witness the negative eigenvalues for $H_G(t^*)$.

Let $\mathbf{1}_c$ denote the indicator vector for the vertices belonging to community c and $\mathbf{1}$ the all-ones vector. We show that every vector in the span of $\{A^{(\ell)}(\mathbf{1}_c - \frac{1}{k}\mathbf{1})\}_{c\in[k]}$ achieves a negative quadratic form against $H_{\boldsymbol{G}}(t^*)$, where $A^{(\ell)}$ is the $n \times n$ matrix where the (i,j)-th entry encodes the number of length-ℓ nonbacktracking walks between i and j. This demonstrates a $(k-1)$-dimensional subspace on which the quadratic form is negative. Formally, we show the following:.

Proposition 2.2. *Under the same setting and notations as Proposition 2.1, for $\ell \geqslant 0$ define*

$$M_{\boldsymbol{G},\ell} \triangleq A^{(\ell)} H_{\boldsymbol{G}}(t^*) A^{(\ell)}.$$

For $\ell = \Theta\left(\frac{\log(1/\varepsilon)}{\varepsilon}\right)$ and every $c \in [k]$, we have

$$\left\langle \mathbf{1}_c - \tfrac{1}{k}\mathbf{1}, M_{\boldsymbol{G},\ell}(\mathbf{1}_c - \tfrac{1}{k}\mathbf{1})\right\rangle \leqslant -\Omega(n).$$

Hence, $M_{G,\ell}$ has at most k negative eigenvalues and at least $k-1$ negative eigenvalues.

Nonbacktracking powers and related constructions were previously studied in [26, 31], but there they take $\ell = \Theta(\log n)$, whereas we only consider constant ℓ. Besides simplifying the analysis of the quadratic form, using constant ℓ is also critical for tolerating up to $\Omega(n)$ corruptions.

As a consequence of Proposition 2.2, the negative eigenvectors of $M_{G,\ell}$ are correlated with the centered community indicators $\{\mathbf{1}_c - \frac{1}{k}\mathbf{1}\}_{c\in[k]}$, while the negative eigenvectors of $H_G(t^*)$ are correlated with $\{A^{(\ell)}(\mathbf{1}_c - \frac{1}{k}\mathbf{1})\}_{c\in[k]}$. The upshot is that we can directly use the negative eigenvectors of $M_{G,\ell}$ to recover the true communities in the absence of corruptions.

Remark 2.3. Based on the empirical observations in [24, 32], a natural hope is to directly use the Bethe Hessian for recovery. However, it turns out that the quadratic form of the centered true community indicators $\left\langle (\mathbf{1}_c - \frac{1}{k}\mathbf{1}), H_G(t^*)(\mathbf{1}_c - \frac{1}{k}\mathbf{1})\right\rangle$ are actually *positive* close to the KS threshold, so the same approach does not establish that the negative eigenvectors of $H_G(t^*)$ correlate with the communities.

We will now discuss how to recover the outlier eigenspace in the presence of adversarial corruptions.

2.2 Robust PCA for Sparse Matrices

The discussion above naturally leads to the following algorithmic problem of robust recovery: Given as input a corrupted version $\widetilde{M}$ of a symmetric matrix M, can we recover the bottom/top r-dimensional eigenspace of M? Since the true communities are constantly correlated with the outlier eigenspace of $M = M_{G,\ell}$, recovering the outlier eigenspace of M from its corrupted version $\widetilde{M} = \widetilde{M}_{G,\ell}$ is a major step towards robustly recovering communities.

The problem of robustly recovering the top eigenspace, a.k.a. robust PCA has been extensively studied, and algorithms with provable guarantees have been designed (see [8]). However, the robust PCA problem in our work is distinct from those considered in the literature in a couple of ways. For us, the uncorrupted matrix M is sparse and both the magnitude and location of the noisy entries are adversarial. Furthermore, for our purposes, we need not recover the actual outlier eigenspace of M. Indeed, as we discuss below, it suffices to robustly recover a constant dimensional subspace which is constantly correlated with the true communities.

We design an efficient algorithm to robustly recover such a subspace under a natural set of sufficient conditions on M. Before we describe these conditions, let us fix some notation. We will call a vector $x \in \mathbb{R}^n$ to be C-delocalized if no coordinate is large relative to others, i.e., $|x_i|^2 \leqslant \frac{C}{n}\|x\|^2$ for all $i \in [n]$. Delocalization has previously been used in the robust PCA literature under the name "incoherence" [8].

Let M be a $n \times n$ matrix with at most r negative eigenvalues. In particular, the r-dimensional negative eigenspace V_M of M is the object of interest. Let $\widetilde{M}$ be a corrupted version of M, differing from M in δn coordinates.

Given the corrupted version $\widetilde{M}$, a natural goal would be to recover the r-dimensional negative eigenspace V_M. It is easy to see that it could be impossible to recover the space V_M. Instead, we will settle for a relaxed goal, namely, recover a slightly larger dimensional subspace U that non-trivially correlates with delocalized vectors in the true eigenspace V_M. More formally, we will solve the following problem.

Problem 2.4. Given the corrupted matrix $\widetilde{M}$ as input, give an efficient algorithm to output a subspace U with the following properties:

(1) **Low dimensional.** The dimension of U is $O(r)$.
(2) **Delocalized.** The diagonal entries of its projection matrix Π_U are bounded by $O\left(\frac{r}{n}\right)$.
(3) **Preserves delocalized part of negative eigenspace.** For any C-delocalized unit vector y such that $\langle y, My\rangle < -\Omega(1)$, we have $\langle y, \Pi_U y\rangle \geqslant \Omega(1)$.

In fact, our algorithm will recover a principal submatrix of $\widetilde{M}$ whose eigenspace V for eigenvalues less than $-\eta$ is $O(r)$-dimensional. Moreover, the eigenspace V can be processed to another delocalized, $O(r)$-dimensional subspace U that satisfies the conditions outlined above.

Although the matrix M has a constant number of negative eigenvalues, its corruption $\widetilde{M}$ can have up to $\Omega(n)$ many. At first glance, it may be unclear how a constant dimensional subspace U can be extracted from $\widetilde{M}$. The crucial observation is that the large negative eigenvalues introduced by the corruptions are highly localized. Thus, we will design an iterative trimming algorithm that aims to delete rows and columns to clean up these localized corruptions. When the algorithm terminates, it yields the $O(r)$-dimensional subspace V.

Recovering a Principal Submatrix. We now describe the trimming algorithm informally and refer the reader to the full version of the paper for the formal details.

We first fix some small parameter $\eta > 0$ and execute the following procedure, which produces a series of principal submatrices $\widetilde{M}^{(t)}$ for $t \geqslant 0$, starting with $\widetilde{M}^{(0)} \triangleq \widetilde{M}$.

(1) At step t, if the eigenspace V of eigenvalues of $\widetilde{M}^{(t)}$ less than $-\eta$ is $O(r)$-dimensional, we terminate the algorithm and output V.
(2) Otherwise, compute the projection $\Pi^{(t)}$ corresponding to the $\leqslant -\eta$ eigenspace of $\widetilde{M}^{(t)}$.
(3) Sample an index $i \in [n]$ of $\widetilde{M}^{(t)}$ with probability proportional to $\Pi^{(t)}_{i,i}$.
(4) Zero out row and column i, and set this new principal submatrix as $\widetilde{M}^{(t+1)}$.

We now discuss the intuition behind the procedure and formally prove its correctness in the full version. The main idea of step 3 is that one should prefer to delete highly localized eigenvectors which have relatively large negative eigenvalues. This is reasonable because the size of the diagonal entries of $\widetilde{M}^{(t)}$ serve as a rough proxy for the level of delocalization.

As a concrete illustration of this intuition, suppose that $\widetilde{M} = \Pi^{(0)} = -uu^\top - vv^\top$, where u, v are orthogonal unit vectors. Moreover, suppose u is C-delocalized whereas $v = e_1$. Then $\Pi^{(0)}_{1,1} = 1$ whereas $|\Pi^{(0)}_{i,i}| \leqslant C^2/n$ for $i > 1$. Hence, deleting the first row and column of $\widetilde{M}$ also deletes the localized eigenvector v. In general, whenever one of the eigenvectors of $\widetilde{M}^{(t)}$ is heavily localized on a subset of coordinates S, the diagonal entries in $\Pi^{(t)}_{S,S}$ are disproportionately large. This leads to a win-win scenario: either we reach the termination condition, or we are likely to mitigate the troublesome large localized eigenvectors.

We now discuss how we achieve the second and third guarantees in Problem 2.4.

Trimming the Subspace. The final postprocessing step is simple. Let V denote the eigenspace with eigenvalues less than $-\eta$ for the matrix $\widetilde{M}^{(T)}$ obtained at end of iterative procedure.

To ensure delocalization (condition 2 in Problem 2.4), the idea is to take its projector Π_V and trim away the rows and columns with diagonal entry exceeding $\frac{\tau}{n}$ for some large parameter $\tau > 0$. The desired delocalized subspace U is obtained by taking the eigenspace of the trimmed Π_V corresponding to the eigenvalues exceeding a threshold that is $O(\eta)$. Since V is $O(r)$-dimensional, so too is U.

The more delicate part is condition 3 in Problem 2.4. Namely, we must show that despite corruptions and the repeated trimming steps, x remains a delocalized witness vector for Π_U, and thus has constant correlation with the subspace U. The key intuition for this is that delocalized witnesses are naturally robust to adversarial corruptions, so long as the adversarial corruptions have bounded row-wise ℓ_1 norm. In particular, since delocalization is an ℓ_∞ constraint, Hölder's inequality bounds the difference in value of the quadratic form using M and $\widetilde{M}$. In the full version of the paper, we prove that for sufficiently small constant levels of corruption, x is also a delocalized witness for $\widetilde{M}$ and Π_U.

Finally, we discuss how to round the recovered subspace U into a community assignment.

2.3 Rounding to Communities

At this stage, we are presented with a constant-dimensional subspace U with the key feature that it is correlated with the community assignment vectors $\{\mathbf{1}_c\}_{c\in[k]}$. Our goal is to round U to a community assignment that is "well-correlated" with the ground truth. In order to discuss how we achieve this goal, we must make precise what it means to be "well-correlated" with the ground truth. Notice that a community assignment is just as plausible as the same assignment with the names of communities permuted, and thus counting the number of correctly labeled vertices is not a meaningful metric.

A more meaningful metric is the number of pairwise mistakes, i.e. the number of pairs of vertices in the same community assigned to different communities or in different communities assigned to the same community. A convenient way to express this metric is via the inner product of positive semidefinite matrices encoding whether pairs of vertices belong to the same community or not. Given a community assignment x, we assign it the matrix X, defined as

$$X[i,j] = \begin{cases} 1 & \text{if } x(i) = x(j) \\ -\frac{1}{k-1} & \text{if } x(i) \neq x(j). \end{cases}$$

For the ground truth assignment $\boldsymbol{x}$ and the output of our algorithm $\widehat{x}$, we measure the correlation with $\langle \boldsymbol{X}, \widehat{X}\rangle$. Observe that for any guess $\widehat{X}$ that is oblivious to the input (for example, classifying all vertices to the same community, or blindly guessing), the value of $\langle \boldsymbol{X}, \widehat{X}\rangle$ is concentrated below $\widetilde{O}(n^{3/2})$. On the other hand, if $\widehat{X} = \boldsymbol{X}$, then this correlation is $\Omega(n^2)$. See Definition 4.4 for how this notion generalizes to arbitrary block models, and subsumes other notions of weak-recovery defined in literature.

The projection matrix Π_U satisfies

$$\langle \Pi_U, \boldsymbol{X}\rangle \geqslant \Omega(\|\Pi_U\|_F \cdot \|\boldsymbol{X}\|_F) = \Omega(n).$$

We give a randomized rounding strategy according to which $\mathbf{E}\,\widehat{X} \succeq c \cdot n \cdot \Pi_U$ for some constant $c > 0$. Consequently, $\mathbf{E}\langle \boldsymbol{X}, \widehat{X}\rangle = cn \cdot \langle \Pi_U, \boldsymbol{X}\rangle \geqslant \Omega(n^2)$.

Observe that for any community assignment x, its matrix representation X is rank-$(k-1)$, which lets us write it as $VV^\top$ for some $n \times (k-1)$ matrix V. Here, the i-th row of V is some vector $v_{x(i)}$ that only depends only on the community $x(i)$ where vertex i is assigned.

Our rounding scheme uses Π_U to produce an embedding of the n vertices as rows of a $n \times (k-1)$ matrix W whose rows are in $\{v_1, \dots, v_k\}$. In the community assignment $\widehat{x}$ outputted by the algorithm, the i-th vertex is assigned to community j if the i-th row of W is equal to v_j. We then show that $\mathbf{E}\, WW^\top \succeq c \cdot n \cdot \Pi_U$. Since $\widehat{X} = \mathbf{E}\, WW^\top$, we can conclude $\mathbf{E}\langle \boldsymbol{X}, \widehat{X}\rangle \geqslant \Omega(n^2)$.

Rounding Scheme. Our first step is to obtain an embedding of the n vertices into $\mathbb{R}^{k-1}$ by choosing a $(k-1)$-dimensional random subspace U' of U, then writing its projector as $M'M'^\top$, and choosing the embedding as the rows of M': $u_1', \dots, u_n'$. Suppose this embedding has the property that for some $c' > 0$, the rows of $\sqrt{c'n}U'$ lie inside the convex hull of $v_1, \dots, v_k$, then we can express each u_i' as a convex combination $\sum_{j=1}^k p_j^{(i)} v_j$ and then independently sample w_i from $\{v_1, \dots, v_k\}$ according to the probability distribution $(p_j^{(i)})_{j\in[k]}$. The resulting embedding W would satisfy the property that $\mathbf{E}\, WW^\top \succeq c' \cdot \frac{k-1}{\dim(U)} \cdot n \cdot \Pi_U$, where this inequality holds since the off-diagonal entries are equal, and the diagonal of $WW^\top$ is larger.

The reason an appropriate scaling c' exists follows from the facts that the convex hull of $v_1, \dots, v_k$ is full-dimensional and contains the origin, which we prove in the full version of the paper.

3 PRELIMINARIES

Stochastic Block Model Notation. We write $\mathbf{1}$ to denote the all-ones vector and e_i to denote the ith standard basis vector, with the dimensions implicit. For a k-community block model, let $\pi \in \mathbb{R}^k$ denote the prior community probabilities, and $\Pi = \operatorname{diag}(\pi)$, so that $\pi = \Pi\mathbf{1}$. The edge probabilities are parameterized by a symmetric matrix $\mathrm{M} \in \mathbb{R}^{k\times k}$, the block probability matrix. A true community assignment $\boldsymbol{x} : [n] \to [k]$ is sampled i.i.d. from π. Conditioned on $\boldsymbol{x}$, an edge between i and j is sampled with probability $\frac{\mathrm{M}_{\boldsymbol{x}(i),\boldsymbol{x}(j)} d}{n}$. To ensure that the average degree is d, we stipulate that $\mathrm{M}\pi = \mathbf{1}$.

We will also use $\boldsymbol{X} \in \mathbb{R}^{n\times k}$ to denote the *one-hot encoding* of $\boldsymbol{x}$, i.e., the matrix where the t-th row is equal to $e_{\boldsymbol{x}(t)}$. We will sometimes find it convenient to access the columns of $\boldsymbol{X}$, which are the indicator vectors for the k different communities; we denote these by $\mathbf{1}_c$ for any community $c \in [k]$. For any $f : [k] \to \mathbb{R}$, define the lift of f with respect to the true community assignment by $\boldsymbol{f}^{(n)} \triangleq \sum_{c\in[k]} f(c) \cdot \mathbf{1}_c$.

Another natural matrix that appears throughout the analysis is the Markov transition matrix $T \triangleq M\Pi$, which by detailed balance evidently has stationary distribution π. This is an asymmetric matrix, but since T defines a time-reversible Markov chain with respect to π, T is self-adjoint with respect to the inner product $\langle \cdot, \cdot \rangle_\pi$ in $\mathbb{R}^k$ induced by π. Hence T is diagonalizable with real eigenvalues and its eigenvalues are $1 = \lambda_1 > |\lambda_2| \geqslant \cdots \geqslant |\lambda_k|$, with ties broken by placing positive eigenvalues before the negative ones. Note that the normalization condition $\mathrm{M}\pi = \mathbf{1}$ translates into $T\mathbf{1} = \mathbf{1}$.

Matrix Notation. We use $\preceq$ and $\succeq$ to denote inequalities on matrices in the Loewner order. For any $n \times n$ matrix X, we use $\Pi_{\leqslant a}(X)$ and $\Pi_{\geqslant a}(X)$ to denote the projectors onto the spaces spanned by eigenvectors of X with eigenvalue at most and at least a respectively. We also define $X_{\leqslant a} \triangleq \Pi_{\leqslant a}(X) X \Pi_{\leqslant a}(X)$ and $X_{\geqslant a} \triangleq \Pi_{\geqslant a}(X) X \Pi_{\geqslant a}(X)$, the corresponding truncations of the eigendecomposition of X.

For $S \subseteq [n]$, we use $X_{S,S}$ to denote the matrix obtained by taking X and zeroing out all rows and columns with indices outside S.

Nonbacktracking Matrix and Bethe Hessian. For a graph G, let B_G be its nonbacktracking matrix, A_G be its adjacency matrix, D_G be its diagonal matrix of degrees, $A_G^{(\ell)}$ be its ℓ-th nonbacktracking power of A_G, and $H_G(t) \triangleq (D_G - I)t^2 - A_G t + I$ be its Bethe Hessian matrix. The matrix we use for our algorithm is $M_{G,\ell}(t) \triangleq A_G^{(\ell)} H_G(t) A_G^{(\ell)}$. We will drop the G from the subscript when the graph G is clear from context.

Determinants. Below, we collect some standard linear algebraic facts that will prove useful.

Fact 3.1. *Suppose a matrix X has a kernel of dimension k, then every $(n-j) \times (n-j)$ submatrix of X for $j < k$ is singular.*

Fact 3.2 (Jacobi's formula). *For any differentiable function $X : \mathbb{R} \to \mathbb{R}^{n \times n}$,*

$$\frac{d}{du} \det(X(u)) = \sum_{i=1}^{n} \sum_{j=1}^{n} \det\Big(X(u)_{[n]\setminus\{i\},[n]\setminus\{j\}}\Big) \cdot (-1)^{i+j} \cdot \frac{d}{du}(X(u))_{i,j}.$$

Lemma 3.3. *Let $X : \mathbb{R} \to \mathbb{R}^{n \times n}$ and $f : \mathbb{R} \to \mathbb{R}$ be any pair of smooth functions. For any $j \geqslant 0$, there exist functions $(q_{S,T} : \mathbb{R} \to \mathbb{R})_{S,T \subseteq [n],\ |S|=|T| \geqslant n-j}$ such that:*

$$\left(\frac{d}{du}\right)^j [\det(X(u)) \cdot f(u)] = \sum_{\substack{S,T \subseteq [n] \\ |S|=|T| \geqslant n-j}} \det\big(X(u)_{S,T}\big) q_{S,T}(u).$$

Proof. We prove this by induction. This is clearly true when $j = 0$, and the induction step is a consequence of Jacobi's formula. □

Kesten-Stigum Threshold. We say that a stochastic block model is above the Kesten–Stigum (KS) threshold if $\lambda_2(T)^2 d > 1$, where recall that λ_2 is the second largest eigenvalue in absolute value. We use r to denote the number of eigenvalues of T equal to $\lambda_2(T)$.

4 RECOVERY ALGORITHM

Let $\boldsymbol{G}$ be the graph drawn from $\mathrm{SBM}_n(\mathrm{M}, \pi, d)$, and let $\widetilde{\boldsymbol{G}}$ denote the input graph which is $\boldsymbol{G}$ along with an arbitrary δn adversarial edge corruptions. Our algorithm for clustering the vertices into communities proceeds in multiple phases, described formally below.

The first phase preprocesses the graph by making it bounded degree and constructs an appropriate matrix M associated to the graph. The second phase cleans up M and uses a spectral algorithm to robustly recover a subspace containing nontrivial information about the true communities. Finally, the third phase rounds the subspace to an actual community assignment.

Algorithm 4.1. $\widetilde{\boldsymbol{G}}$ is given as input, and a community assignment to the vertices is produced as output.

Phase 1: Deletion of High-degree Vertices. For some large constant $B > 0$ to be specified later, we perform the following truncation step: delete all edges incident on vertices with degree larger than B in $\widetilde{\boldsymbol{G}}$. This forms a graph $\widetilde{\boldsymbol{G}}_B$, with corresponding adjacency matrix $A_{\widetilde{\boldsymbol{G}}_B} \in \mathbb{R}^{|V(\boldsymbol{G})| \times |V(\boldsymbol{G})|}$. To avoid confusion, we preserve the vertex set $V(\boldsymbol{G})$, but it should be understood that the truncated vertices do not contribute to the graph.

For technical considerations, we also define a (nonstandard) truncated diagonal matrix

$$\overline{D}_{\widetilde{\boldsymbol{G}}_B} \triangleq \operatorname{diag}\left(\deg(v) \mathbf{1}[\deg(v) \leqslant B]\right)_{v \in V(\boldsymbol{G})} \tag{1}$$

With this, we can then define the truncated Bethe Hessian matrix

$$\overline{H}_{\widetilde{\boldsymbol{G}}_B}(t) \triangleq I - tA_{\widetilde{\boldsymbol{G}}_B} + t^2(\overline{D}_{\widetilde{\boldsymbol{G}}_B} - I). \tag{2}$$

Finally, the input matrix to the next phase is

$$\overline{M}_{\widetilde{\boldsymbol{G}}_B,\ell}(t) \triangleq A^{(\ell)}_{\widetilde{\boldsymbol{G}}_B} \overline{H}_{\widetilde{\boldsymbol{G}}_B}(t) A^{(\ell)}_{\widetilde{\boldsymbol{G}}_B}, \tag{3}$$

where we also choose the value of t later.

Remark 4.2. To reduce any chance of confusion with the notation, we reiterate our conventions for distinguishing between different versions of various matrices. If a graph is truncated at level B, then we add a subscript B. We use tilde to denote that we are working with a corrupted graph. Finally, we use overline to denote that we are working with the nonstandard version of the Bethe Hessian after truncation.

For example, the matrix $\overline{D}_{\boldsymbol{G}_B}$ no longer corresponds to the degree matrix of $\boldsymbol{G}_B$, since as stated it still counts edges from truncated vertices. This is done to simplify the analysis of the spectrum of $\overline{M}_{\boldsymbol{G}_B,\ell}(t)$ but we do not believe it to be essential.

Phase 2: Recovering a Subspace with Planted Signal. Define $M \triangleq \overline{M}_{\widetilde{\boldsymbol{G}}_B,\ell}$. We give an iterative procedure to "clean up" M by deleting a few rows and columns. We then run a spectral algorithm on the cleaned up version of M.

Let $\eta > 0$ be a small constant we choose later, and let $K \triangleq B^{2\ell+3}$.

(1) Define $M^{(0)}$ as M. Let t be a counter initialized at 0, and $\Phi(X)$ as the number of eigenvalues of X smaller than $-\eta$.
(2) While $\Phi(M^{(t)}) > \frac{2K}{\eta} r$: compute the projection matrix $\Pi^{(t)} \triangleq \Pi_{\leqslant -\eta}(M^{(t)})$, choose a random $i \in [n]$ with probability

$\frac{\Pi^{(t)}_{i,i}}{\mathrm{Tr}(\Pi^{(t)})}$, and define $M^{(t+1)}$ as the matrix obtained by zeroing out the i-th row and column of $M^{(t)}$. Then increment t.

Let T be the time of termination and $\tau > 0$ be a large enough constant we choose later. We compute $\Pi^{(T)}$, and then compute as the set S of all indices i where $\Pi^{(T)}_{i,i} \leqslant \frac{\tau}{n}$. Define $\widetilde{\Pi}$ as $\left(\Pi^{(T)}_{S,S}\right)_{\geqslant \eta/K}$, and compute its span U, where we recall that $(X)_{\geqslant a}$ denotes the truncation of the eigendecomposition of X for eigenvalues at least a. This subspace U is passed to the next phase.

Phase 3: Rounding to a Community Assignment. Define r' as $r-1$ when $\lambda_2(T) > 0$ and as r when $\lambda_2(T) < 0$. We first obtain an r'-dimensional embedding of the vertices into $\mathbb{R}^{r'}$. Compute a random r'-dimensional subspace U' of U, and take an orthogonal basis $u'_1, \ldots, u'_{r'}$. Place these vectors as a column of a matrix M' in $\mathbb{R}^{n\times r'}$. The rows of M' gives us the desired embedding.

On the other hand, we use the natural embedding of the k communities into $\mathbb{R}^{r'}$ induced by the r' nontrivial right eigenvectors corresponding to the eigenvalue $\lambda_2(T)$: $(\psi_i)_{1\leqslant i\leqslant r'}$ of T. In particular, let $\Psi_{r'} \triangleq \begin{bmatrix}\psi_1 & \cdots & \psi_{r'}\end{bmatrix} \in \mathbb{R}^{k\times r'}$ be the matrix of these r' nontrivial eigenvectors of T. Then the row vectors $\phi_1, \ldots, \phi_k \in \mathbb{R}^{r'}$ form the desired embedding of communities.

In the rounding algorithm, we first find the largest c such that all the rows of $c \cdot M'$ lie in the convex hull of $\phi_1, \ldots, \phi_k$. We can find such a value of c if it exists by solving a linear program, and we prove that this $c > 0$ is guaranteed to exist in the full version of this paper. Then, for each $i \in [n]$ we express each row of $c \cdot M'$ as a convex combination $\sum_{j=1}^{k} w_i^{(j)} \phi_j$ for nonnegative $w_i^{(j)}$ such that $\sum_{j=1}^{k} w_i^{(j)} = 1$. Finally, we assign vertex i to community j with probability $w_i^{(j)}$, and output the resulting community assignment $\widehat{\boldsymbol{x}}$.

Remark 4.3. Scaling the rows of M' so as to lie in the convex hull of $\{\phi_j\}_{j\in[k]}$, is reminiscent of the rounding algorithm of Charikar & Wirth [10] to find a cut of size $\frac{1}{2} + \Omega(\frac{\varepsilon}{\log(1/\varepsilon)})$ in a graph with maximum cut of size $\frac{1}{2} + \varepsilon$: in their algorithm, they scale n scalars to lie in the interval $[-1, 1]$.

Analysis of Algorithm. Our goal is to prove that the output $\widehat{\boldsymbol{x}}$ of our algorithm is well-correlated with the true community assignment $\boldsymbol{x}$. We begin by defining a notion of weak recovery for k-community stochastic block models.

Definition 4.4 (Weak recovery). Let $\Psi \triangleq \begin{bmatrix}\psi_2 & \cdots & \psi_k\end{bmatrix} \in \mathbb{R}^{k\times(k-1)}$ be the matrix of the top-$(k-1)$ nontrivial eigenvectors of the transition matrix T of a stochastic block model.

For $\rho > 0$, we say that a (randomized) algorithm for producing community assignments $\widehat{\boldsymbol{X}} \in \mathbb{R}^{n\times k}$ achieves ρ-weak recovery if

$$\left\langle \mathbf{E}\,\widehat{\boldsymbol{X}}_\Psi, \boldsymbol{X}_\Psi \right\rangle \geqslant \rho \|\mathbf{E}\,\widehat{\boldsymbol{X}}_\Psi\|_F \|\boldsymbol{X}_\Psi\|_F,$$

where $B_\Psi \triangleq (B\Psi)(B\Psi)^\top$ for a matrix $B \in \mathbb{R}^{n\times k}$.

Remark 4.5. Intuitively, this notion is capturing the "advantage" of the algorithm over random guessing, or simply outputting the most likely community. See the full version for a more detailed discussion of this notion, how it recovers other previously considered measures of correlation in the case of the symmetric block model, and why it is meaningful. In particular, it implies the notion of weak recovery used in [13].

Our main guarantee is stated below.

Theorem 4.6. *For any SBM parameters (M, π, d) above the KS threshold, there is a constant $\rho(\mathrm{M}, \pi, d) > 0$ such that the above algorithm takes in the corrupted graph $\widetilde{\boldsymbol{G}}$ as input and outputs $\widehat{\boldsymbol{x}}$ achieving $\rho(\mathrm{M}, \pi, d)$-weak recovery with probability $1 - o_n(1)$ over the randomness of $\boldsymbol{G} \sim \mathrm{SBM}_n(\mathrm{M}, \pi, d)$.*

To prove the above theorem it suffices to analyze $\left\langle (\mathbf{E}\,\widehat{X})_\Psi, \boldsymbol{X}_\Psi \right\rangle$. To see why, let us first set up some notation. For each vertex i, we obtain a simplex vector $w_i \in \mathbb{R}^k$, which we can stack as rows into a weight matrix $W \in \mathbb{R}^{n\times k}$. We then independently round each vertex so that $\mathbf{E}\,\widehat{\boldsymbol{X}} = W$.

To analyze our rounding scheme, first note that $\mathbf{E}[\widehat{\boldsymbol{X}}_\Psi]$ is equal to W_Ψ off of the diagonal and is larger than W_Ψ on the diagonal, and thus $\mathbf{E}[\widehat{\boldsymbol{X}}_\Psi] \succeq W_\Psi$. Since $\boldsymbol{X}_\Psi$ is positive semidefinite, $\left\langle \mathbf{E}[\widehat{\boldsymbol{X}}_\Psi], \boldsymbol{X}_\Psi \right\rangle \geqslant \langle W_\Psi, \boldsymbol{X}_\Psi\rangle$. Thus, it suffices to lower bound $\langle W_\Psi, \boldsymbol{X}_\Psi\rangle$. By construction, W_Ψ is equal to $c^2 \cdot \Pi_{U'}$, where recall that U' was a random r'-dimensional subspace of U, the output of Phase 2 of the algorithm. Thus,

$$\mathop{\mathbf{E}}_{U'} W_\Psi = c^2 \cdot \mathop{\mathbf{E}}_{U'} \Pi_{U'} = c^2 \cdot \frac{r'}{\dim(U)} \Pi_U.$$

In the full version of this manuscript, we prove that when (M, π, d) are above the KS threshold, $\langle \Pi_U, \boldsymbol{X}_\Psi\rangle \geqslant \Omega(1) \cdot \|\Pi_U\|_F \cdot \|\boldsymbol{X}_\Psi\|_F$ and $\dim(U) = O(1)$. Furthermore, we show that when $\mathrm{diag}(\Pi_U) = O(1/n)$, we can take $c = \Omega(\sqrt{n})$; this delocalization condition is guaranteed by phase 2 of the algorithm. Combined with the fact that $\|\widehat{\boldsymbol{X}}_\Psi\|_F = O(n)$, it follows that $\left\langle \mathbf{E}\,\widehat{\boldsymbol{X}}_\Psi, \boldsymbol{X}_\Psi \right\rangle \geqslant \Omega(1) \cdot \|\widehat{\boldsymbol{X}}_\Psi\|_F \cdot \|\boldsymbol{X}_\Psi\|_F$, which establishes Theorem 4.6.

ACKNOWLEDGMENTS

We would like to thank Omar Alrabiah and Kiril Bangachev for diligent feedback on an earlier draft of this paper. DW acknowledges support from NSF Graduate Research Fellowship DGE-2146752.

REFERENCES

[1] Emmanuel Abbe. 2017. Community detection and stochastic block models: recent developments. *The Journal of Machine Learning Research* 18, 1 (2017), 6446–6531. https://doi.org/10.1561/0100000067
[2] Emmanuel Abbe and Colin Sandon. 2015. Community detection in general stochastic block models: Fundamental limits and efficient algorithms for recovery. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*. IEEE, 670–688. https://doi.org/10.1109/focs.2015.47
[3] Jess Banks, Sidhanth Mohanty, and Prasad Raghavendra. 2021. Local Statistics, Semidefinite Programming, and Community Detection. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 1298–1316. https://doi.org/10.1137/1.9781611976465.79
[4] Hyman Bass. 1992. The Ihara-Selberg zeta function of a tree lattice. *International Journal of Mathematics* 3, 06 (1992), 717–797.
[5] Avrim Blum and Joel Spencer. 1995. Coloring random and semi-random k-colorable graphs. *Journal of Algorithms* 19, 2 (1995), 204–234. https://doi.org/10.1006/jagm.1995.1034
[6] Charles Bordenave, Marc Lelarge, and Laurent Massoulié. 2015. Non-backtracking spectrum of random graphs: community detection and non-regular Ramanujan graphs. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*. IEEE, 1347–1357. https://doi.org/10.1109/focs.2015.86

[7] Rares-Darius Buhai, Pravesh K Kothari, and David Steurer. 2023. Algorithms approaching the threshold for semi-random planted clique. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*. 1918–1926. https://doi.org/10.1145/3564246.3585184

[8] Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright. 2011. Robust principal component analysis? *Journal of the ACM (JACM)* 58, 3 (2011), 1–37. https://doi.org/10.1145/1970392.1970395

[9] Moses Charikar, Jacob Steinhardt, and Gregory Valiant. 2017. Learning from untrusted data. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*. 47–60. https://doi.org/10.1145/3055399.3055491

[10] Moses Charikar and Anthony Wirth. 2004. Maximizing quadratic programs: Extending Grothendieck's inequality. In *45th Annual IEEE Symposium on Foundations of Computer Science*. IEEE, 54–60. https://doi.org/10.1109/focs.2004.39

[11] Aurelien Decelle, Florent Krzakala, Cristopher Moore, and Lenka Zdeborová. 2011. Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. *Physical review E* 84, 6 (2011), 066106. https://doi.org/10.1103/physreve.84.066106

[12] Amit Dembo, Andrea Montanari, and Subhabrata Sen. 2017. Extremal cuts of sparse random graphs. *The Annals of Probability* 45, 2 (2017), 1190–1217. https://doi.org/10.1214/15-aop1084

[13] Jingqiu Ding, Tommaso d'Orsi, Rajai Nasser, and David Steurer. 2022. Robust recovery for stochastic block models. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 387–394. https://doi.org/10.1109/focs52979.2021.00046

[14] Jingqiu Ding, Tommaso d'Orsi, Yiding Hua, and David Steurer. 2023. Reaching Kesten-Stigum Threshold in the Stochastic Block Model under Node Corruptions. In *The Thirty Sixth Annual Conference on Learning Theory*. PMLR, 4044–4071. https://doi.org/10.48550/arxiv.2305.10227

[15] Zhou Fan and Andrea Montanari. 2017. How well do local algorithms solve semidefinite programs?. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*. 604–614. https://doi.org/10.1145/3055399.3055451

[16] Uriel Feige and Joe Kilian. 2001. Heuristics for semirandom graph problems. *J. Comput. System Sci.* 63, 4 (2001), 639–671. https://doi.org/10.1006/jcss.2001.1773

[17] Yuzhou Gu and Yury Polyanskiy. 2023. Uniqueness of BP fixed point for the Potts model and applications to community detection. *arXiv preprint arXiv:2303.14688* (2023). https://doi.org/10.48550/arxiv.2303.14688

[18] Olivier Guédon and Roman Vershynin. 2016. Community detection in sparse networks via Grothendieck's inequality. *Probability Theory and Related Fields* 165, 3-4 (2016), 1025–1049. https://doi.org/10.1007/s00440-015-0659-z

[19] Venkatesan Guruswami, Jun-Ting Hsieh, Pravesh K Kothari, and Peter Manohar. 2023. Efficient Algorithms for Semirandom Planted CSPs at the Refutation Threshold. *arXiv preprint arXiv:2309.16897* (2023). https://doi.org/10.1109/focs57990.2023.00026

[20] Samuel B Hopkins and David Steurer. 2017. Efficient Bayesian estimation from few samples: community detection and related problems. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 379–390. https://doi.org/10.1109/focs.2017.42

[21] Yasutaka Ihara. 1966. On discrete subgroups of the two by two projective linear group over p-adic fields. *Journal of the Mathematical Society of Japan* 18, 3 (1966), 219–235. https://doi.org/10.2969/jmsj/01830219

[22] Harry Kesten and Bernt P Stigum. 1966. Additional limit theorems for indecomposable multidimensional Galton-Watson processes. *The Annals of Mathematical Statistics* 37, 6 (1966), 1463–1481. https://doi.org/10.1214/aoms/1177699139

[23] Harry Kesten and Bernt P Stigum. 1967. Limit theorems for decomposable multidimensional Galton-Watson processes. *J. Math. Anal. Appl.* 17, 2 (1967), 309–338. https://doi.org/10.1016/0022-247x(67)90155-2

[24] Florent Krzakala, Cristopher Moore, Elchanan Mossel, Joe Neeman, Allan Sly, Lenka Zdeborová, and Pan Zhang. 2013. Spectral redemption in clustering sparse networks. *Proceedings of the National Academy of Sciences* 110, 52 (2013), 20935–20940. https://doi.org/10.48550/arXiv.1306.5550

[25] Allen Liu and Ankur Moitra. 2022. Minimax rates for robust community detection. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 823–831. https://doi.org/10.1109/focs54457.2022.00083

[26] Laurent Massoulié. 2014. Community detection thresholds and the weak Ramanujan property. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*. 694–703. https://doi.org/10.1145/2591796.2591857

[27] Theo McKenzie, Hermish Mehta, and Luca Trevisan. 2020. A new algorithm for the robust semi-random independent set problem. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 738–746. https://doi.org/10.48550/arXiv.1808.03633

[28] Ankur Moitra, William Perry, and Alexander S Wein. 2016. How robust are reconstruction thresholds for community detection?. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*. 828–841. https://doi.org/10.1145/2897518.2897573

[29] Andrea Montanari and Subhabrata Sen. 2016. Semidefinite programs on sparse random graphs and their application to community detection. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*. 814–827. https://doi.org/10.1145/2897518.2897548

[30] Elchanan Mossel, Joe Neeman, and Allan Sly. 2014. Belief propagation, robust reconstruction and optimal recovery of block models. In *Conference on Learning Theory*. PMLR, 356–370. https://doi.org/10.48550/arXiv.1309.1380

[31] Elchanan Mossel, Joe Neeman, and Allan Sly. 2018. A proof of the block model threshold conjecture. *Combinatorica* 38, 3 (2018), 665–708. https://doi.org/10.1007/s00493-016-3238-8

[32] Alaa Saade, Florent Krzakala, and Lenka Zdeborová. 2014. Spectral clustering of graphs with the Bethe Hessian. *Advances in neural information processing systems* 27 (2014). https://doi.org/10.48550/arXiv.1406.1880

[33] Ludovic Stephan and Laurent Massoulié. 2019. Robustness of spectral methods for community detection. In *Conference on Learning Theory*. PMLR, 2831–2860. https://doi.org/10.48550/arXiv.1811.05808

[34] Qian Yu and Yury Polyanskiy. 2023. Ising model on locally tree-like graphs: Uniqueness of solutions to cavity equations. *IEEE Transactions on Information Theory* (2023). https://doi.org/10.1109/tit.2023.3316795

Received 13-NOV-2023; accepted 2024-02-11