Minimax Rates for Robust Community Detection

Allen Liu  
Department of EECS  
MIT  
Cambridge, USA  
cliu568@mit.edu

Ankur Moitra  
Department of Mathematics  
MIT  
Cambridge, USA  
moitra@mit.edu

Abstract—In this work, we study the problem of community detection in the stochastic block model with adversarial node corruptions. Our main result is an efficient algorithm that can tolerate an $\epsilon$-fraction of corruptions and achieves error $O(\epsilon) + e^{-\Omega((1-\epsilon)^3)}$, where $C' = (\sqrt{\pi} - \sqrt{\theta})^2$ is the signal-to-noise ratio and $a/n$ and $b/n$ are the inter-community and intra-community connection probabilities respectively. These bounds essentially match the minimax rates for the SBM without corruptions. We also give robust algorithms for $\mathbb{Z}_2$-synchronization. At the heart of our algorithm is a new semidefinite program that uses global information to robustly boost the accuracy of a rough clustering. Moreover, we show that our algorithms are doubly-robust in the sense that they work in an even more challenging robust setting in which the adversarial corruptions can be unbounded. This stands in contrast to the semi-random model.

Index Terms—Community detection, stochastic block model, minimax rates, robustness, semi-random model

I. INTRODUCTION

The stochastic block model (SBM) was introduced by Holland, Laskey and Leinhardt [19] in 1983. It generates a random graph with a planted community structure. For now, we focus on the case of two equal-sized communities. The model works as follows: There is an unknown bisection of the $n$ nodes in the graph and each of the two groups is called a community. Pairs of nodes are connected by an edge independently according to the following rules: If the nodes belong to the same community, the connection probability is $a/n$. And otherwise, if the nodes belong to different communities, the connection probability is $b/n$. Here $a$ and $b$ are parameters and the goal is to understand how well we can approximate the planted bisection for different choices of $a$ and $b$.

The stochastic block model has been extensively studied over the years. It is known that the model exhibits various sharp statistical phase transitions. For example, in the weak recovery problem, the goal is to get nontrivial agreement with the planted bisection. Decelle et al. [9] conjectured that weak recovery is possible iff $(a - b)^2 > 2(a + b)$. This threshold is also called the Kesten-Stigum bound. Their conjecture was based on non-rigorous arguments from statistical physics. Mossel et al. [31] and Massoulie [27] proved the conjecture. Moreover they gave efficient algorithms that solve weak recovery down to the Kesten-Stigum bound. In the exact recovery problem, the goal is to recover the planted bisection exactly with high probability. Abbe et al. [2] showed that exact recovery is possible iff $a = p \log n$ and $b = q \log n$ and $(\sqrt{p} - \sqrt{q})^2 > 2$. Hajek et al. [18] gave an efficient algorithm matching this bound based on semidefinite programming. Note that for exact recovery we need logarithmic average degree to preclude having isolated nodes. In contrast, weak recovery is possible with constant average degree. The results in this paper will work in both regimes.

Recent works have focused on achieving the minimax rates for accuracy. In particular consider the quantity

$$\inf_{\hat{\pi}} \sup_{\Theta} \text{err}(\hat{\pi}, \pi)$$

Here $\pi$ is an estimator for the planted partition and $\Theta$ is a space of parameters for the stochastic block model. For example, we can consider the worst-case error over all stochastic block models on $n$ nodes with imbalance at most $\alpha$ and where the inter-community and intra-community connection probabilities are at least $a/n$ and at most $b/n$ respectively. Finally $\text{err}(\pi, \pi)$ denotes the fraction of misclassified nodes. It is impossible to determine which is the first community and which is the second community, so the error is only measured up to a global swap between the two communities. In particular $0 \leq \text{err} \leq 1/2$. In this notation, weak recovery is possible iff $\text{err} < 1/2$ independently of $n$ and exact recovery is possible iff $\text{err} = 0$ with high probability. And yet, studying the minimax rates allows us to ask sharper questions about the behavior of the optimal accuracy as a function of $a$ and $b$.

Belief propagation is believed to obtain the optimal accuracy in a wide range of parameters. We can think about some of the conjectures, which are now theorems, as being pieces of this puzzle. For example, the trivial fixed points of belief propagation correspond to solutions that achieve $\text{err} = 1/2$. Decelle et al. [9] showed that when $(a - b)^2 > 2(a + b)$ the trivial fixed point is unstable and thus it is natural to expect belief propagation to converge to another solution, which, presumably solves the weak recovery problem. Indeed the algorithms of Mossel et al. [31] and Massoulie [27] can be thought of as low-temperature limits of belief propagation. In a remarkable work, Mossel, Neeman and Sly [30] gave an algorithm that achieves the optimal error when the signal-to-
noise ratio, defined as \( C = (\sqrt{n} - \sqrt{b})^2 \) is large enough. Again their algorithm, particularly their method for boosting the overall accuracy of a rough initial estimate, has important parallels with belief propagation. Zhang and Zhou [36] gave an approximate characterization of the minimax rates. They showed that 

\[
\inf_{\hat{C}} \sup_{C} \text{err}(\hat{C}, C) = e^{-\frac{1}{2}(1+o(1))}
\]

for the two-community, approximately balanced case. Here the \( o(1) \) term is a function of \( C \) that goes to zero as \( C \) increases. They also prove generalizations to the imbalanced and \( k \)-community case, but their results are only information-theoretic and do not give any efficient algorithms. Fei and Chen [15] showed that the natural semidefinite program achieves this error exponent in the two community, balanced case. For more than two communities, there is believed to be a computational vs. statistical tradeoff beneath the Kesten-Stigum bound [1], but nevertheless the natural conjecture is that belief propagation achieves optimal error among all computationally efficient estimators.

A. Our Results

In this work, we ask an ambitious question: Is it possible to compete with the minimax rates while being robust to adversarial corruptions? We work in the node corruption model, where an adversary is allowed to arbitrarily control all the edges incident to an arbitrary \( \epsilon \)-fraction of the nodes in the graph (see Definition II.3). This models realistic settings where nodes represent agents who might make or break ties in a potentially malicious way so as to affect the outcome of a community detection/graph partitioning algorithm. Our main result is a positive answer to this question, along with a computationally efficient estimator for doing so:

**Theorem 1.1.** [Informal, see Theorem III.3] There is a polynomial-time algorithm that given an \( \epsilon \)-corrupted SBM with \( n \) vertices, edge probabilities \( b/n < a/n \leq 1/2 \), and two communities of sizes between \( cn/2 \) and \( n/(2\alpha) \) for some constant \( \alpha \), outputs a labelling that has expected error at most 

\[
O(\epsilon) + e^{-\frac{1}{2}(1+o(1))} + o(1/n)
\]

where \( C = (\sqrt{a} - \sqrt{b})^2 \).

Makarychev et al. [26] gave an algorithm for almost exact recovery that works even when an adversary may corrupt \( o(n) \) edges. However, their accuracy and robustness guarantees are weaker. They require additional constraints on \( \epsilon \) compared to \( a, b \) – as we discuss in Section IV-A, such constraints are actually necessary in the edge corruption model. Stephan and Massoulié [35] studied node corruptions, but only allowed for \( O(n^3) \) nodes to be corrupted for some constant \( \delta > 0 \). Banks et al. [6] and Ding et al. [14] studied weak recovery in the edge corruption model. In particular Ding et al. [14] showed that if \( a \) and \( b \) are above the Kesten-Stigum bound, there is some \( \epsilon > 0 \) for which weak recovery is still possible even when as many as \( \epsilon n \) edges are adversarially added/deleted. Compared to our results, both the goal (competing with the minimax error vs. getting non-trivial error) and the corruption model (node corruptions vs. edge corruptions) are different. Moreover the order of quantifiers is different since in their work the fraction of corruptions is allowed to be an arbitrary function of \( a \) and \( b \). In contrast, our results give essentially tight bounds on the largest fraction of corruptions that can be tolerated while still competing with the minimax error in the stochastic model. See Section IV-A for further discussion of node vs. edge corruptions and its effect on the minimax rates. Finally Acharya et al. [3] studied the related problem of estimating the parameter \( p \) in an Erdos-Renyi random graph \( G(n, p) \) with node corruptions.

We also extend our results to the \( k \) community case:

**Theorem 1.2.** [Informal, see Theorem III.4] There is a polynomial-time algorithm that given an \( \epsilon \)-corrupted SBM with \( n \) vertices, edge probabilities \( b/n < a/n \leq 1/2 \), and \( k \) communities of sizes between \( cn/k \) and \( n/(k\alpha) \) for some constants \( k, \alpha \), outputs a labelling that has expected error at most 

\[
O(\epsilon) + e^{-\frac{1}{2}C(1+o(1))} + o(1/n)
\]

where \( C = (\sqrt{a} - \sqrt{b})^2 \).

**Remark.** Note that the exponent of \( -\alpha C/k \) is also optimal, even in the non-robust setting, as it matches the lower bound proven in [36].

We also study the \( \mathbb{Z}_2 \) synchronization problem: There is an unknown vector \( \ell \in \{ \pm 1 \}^n \) and we observe a spiked random matrix 

\[
\frac{\lambda I + \ell F}{\sqrt{n}} + \frac{W}{\sqrt{n}}
\]

where \( \lambda \) is a parameter and \( W \) is a Gaussian Wigner matrix with iid entries that are mean zero, variance one Gaussians. The goal is to compute an estimate \( \hat{\ell} \) that minimizes the disagreement with \( \ell \). Again we cannot determine the sign of \( \ell \) so we measure disagreement between \( \ell \) and \( \hat{\ell} \) with respect to a global sign flip.

The analogues of many of the key results in community detection are known for \( \mathbb{Z}_2 \) synchronization too. For example, in the weak recovery problem the goal is to get an estimate \( \hat{\ell} \) that achieves non-trivial error, which is possible iff \( \lambda > 1 \) [32], [34]. There are also sharp characterizations of the asymptotic mutual information [10] which can be used to pin down the minimax rates. See also [15]. Again we ask: is it possible to compete with the minimax rates in the presence of adversarial corruptions? In this setting we allow an adversary to arbitrarily corrupt the entries in an \( \epsilon \)-fraction of the rows/columns (see Definition II.6). Again, we show that this is possible, and give computationally efficient algorithms for doing so:

\[\text{In their paper, they write the signal-to-noise ratio as } (a - b)^2 / (2(a + b)) \text{ which is always within a factor of 2 of our definition. They do not actually write their accuracy as an explicit function of } C. \text{ They only prove that the accuracy is the same as that achieved in a broadcast tree reconstruction problem as long as } C \text{ is at least some universal constant.} \]
**Theorem I.3.** [Informal, see Theorem III.6] There is a polynomial-time algorithm that given an $e$-corrupted $\mathbb{Z}_2$-synchronization instance with parameter $\lambda$, outputs a labelling that has expected error at most
\[ O(e) + e^{-\lambda^2/(1+\omega(1))} + o(1/n). \]

Remark. Note that the exponent of $-\lambda^2/2$ is optimal, even in the non-robust setting, as it matches the lower bound proven in [15].

These results come somewhat as a surprise. As we discussed earlier, the minimax rates are closely related to belief propagation, and belief propagation is inherently brittle, particularly to adversarial corruptions. Moreover, the positive results in the non-robust setting essentially all come from finding a good initial estimate of the planted bisection, and then boosting using some local procedure like having each node taking the majority vote of its neighbors [30]. Even approaches based on semi-definite programming do this, within a primal-dual analysis [15]. The main issue is that this approach is doomed in the setting of adversarial node corruptions. In particular, an adversary can game the algorithm in such a way that performing “boosting” results in a new partition that only achieves trivial error (see Observation IV.1).

At the heart of our results is a way to perform robust boosting based on using global information about the entire graph. In particular, we give a semi-definite programming algorithm to discover and correct large sets of nodes that are unduly affecting the labels of many other nodes. More broadly, our work raises the exciting possibility that, maybe even beyond the stochastic block model, it is possible to compete with the sharp error rates achieved by belief propagation all while being provably robust. See Section IV for a more detailed technical overview.

**B. Doubly-Robust Community Detection**

Our work fits into the broader agenda of designing algorithms for inference and learning with strong provable robustness guarantees [4], [5], [7], [11]–[13], [20], [22]–[25]. Much of the literature operates in a setting where samples are generated from a “nice” distribution where the moments are regular and well-behaved and can be used to detect large groups of correlated outliers. So far, this is the case for our algorithms too, since we can rely on the predictable spectral properties of graphs generated from the stochastic block model.

Without corruptions, the important work of Feige and Kilian [16] considered augmenting the stochastic block model with a monotone adversary. This is called the semi-random model. After a graph is sampled from the stochastic block model, but before it is revealed to our algorithm, the monotone adversary is allowed to arbitrarily add edges between pairs of nodes belonging to the same community, and delete edges between pairs of nodes belonging to different communities. This seemingly only makes the problem easier. But in fact designing algorithms that continue to work in the semi-random model is challenging and subtle. In many ways, the semi-random model prevents algorithms from being tuned to the stochastic block model. For exact recovery, algorithms based on semidefinite programming continue to work in the semi-random model in the same range of parameters [18], [33]. For weak recovery, Moitra et al. [28] showed that it is no longer possible to get algorithms that work down to the Kesten-Stigum bound, and thus there is a strict information-theoretic separation between the stochastic and semi-random models. Finally Fei and Chen [15] gave an algorithm, also based on semidefinite programming, that competes with the minimax error in the stochastic setting, even in the presence of a monotone adversary.

Given that being robust is not just about tolerating adversarial corruptions, or any one single goal, it is natural to ask: Are there doubly-robust algorithms for community detection? In particular we want algorithms that work with both adversarial node corruptions and also an unbounded number of monotone changes. Indeed our main algorithms all extend to this challenging setting:

**Theorem I.4** (Informal). There is a polynomial time algorithm that given an $e$-corrupted semi-random SBM outputs a labelling that has the same expected error as in Theorems I.1 and I.2 subject to the same assumptions on the parameters.

A major difficulty of working with node corruptions is that an adversary can corrupt much more than a linear number of edges, because he has control over all the edges incident to corrupted nodes. In the stochastic block model, there is a natural limit to how much an adversary would actually use this power because if some nodes are too high-degree they become easy to identify. But in a semi-random model, a monotone adversary can create many high-degree nodes that we would not actually want to delete. Thus the two types of adversaries can compound our difficulties, and create situations where an overwhelming majority of the edges have been corrupted but nevertheless we cannot easily remove them by deleting high-degree nodes. Our algorithms for $\mathbb{Z}_2$-synchronization are also doubly-robust as Theorem I.3 also continues to hold in an $e$-corrupted and semi-random model.

**C. Broader Context**

We briefly discuss some previous approaches to community detection and why they fail to obtain the robustness guarantees that we aim for here. Some methods are based on computing statistics over non-backtracking walks, e.g. [31] or self-avoiding walks, e.g. [21]. When an adversary can control a constant fraction of the nodes or edges, he can force the expectation of these statistics to be incorrect so that they no longer are correlated with whether a pair of nodes is on the same side of the community or not. Other, related methods are based on spectral properties of the non-backtracking walk operator or a matrix counting all self-avoiding walks of a certain length, e.g. [27]. The spectral properties of these matrices break down, and look fairly arbitrary, with corruptions. Moreover these techniques are for weak recovery, and one
would need to boost to achieve optimal accuracy for larger signal-to-noise ratio. There are approaches for boosting based on belief belief propagation, e.g. [30]. However they are based on approximating the posterior distribution of the community labeling, and when there are corruptions there is no reason for the posterior to achieve optimal, or even non-trivial accuracy. There are also approaches for community detection that are based on semidefinite programming, e.g. [2], [16]–[18], [29]. However, they are all based on an SDP relaxation for the minimum bisection problem, and when an adversary can control high-degree nodes, he can alter the minimum bisection so that it becomes essentially uncorrelated with the planted community structure. While there are modifications such as in [15], [26] that can deal with corruptions, these modifications are not able to achieve the types of strong robustness and accuracy guarantees that we obtain here.

II. Problem Setup

We now formally define the models and problems that we study.

A. Community Detection

We begin with a standard definition of a stochastic block model.

**Definition II.1** (Stochastic Block Model (SBM)). An SBM is a graph on $n$ nodes generated as follows. There is some unknown partition of $[n]$ into $k$ sets $S_1, \ldots, S_k$. Given parameters $a, b$ with $a > b$, a graph is then generated where nodes in the same community are connected with probability $a/n$ and nodes in different communities are connected with probability $b/n$ (all edges are sampled independently).

Next, we introduce the notion of semi-random noise, where an adversary can make arbitrarily many “helpful” changes.

**Definition II.2** (Semi-Random SBM). A Semi-random SBM is a graph on $n$ nodes generated as follows. There is some unknown partition of $[n]$ into $k$ sets $S_1, \ldots, S_k$. Given parameters $a, b$ with $a > b$, a graph is then generated where nodes in the same community are connected with probability $a/n$ and nodes in different communities are connected with probability $b/n$ (all edges are sampled independently). An adversary may then arbitrarily add additional edges within communities and remove edges between different communities.

Finally, we introduce a corruption model where the adversary may completely corrupt an $\epsilon$-fraction of nodes and make semi-random changes on the rest of the graph. Algorithms that work in this setting need to be doubly-robust to both the small fraction of corrupted nodes and the semi-random noise.

**Definition II.3.** [\textit{$\epsilon$-Corrupted Semi-Random SBM}] A \textit{$\epsilon$-corrupted} Semi-Random SBM is a graph on $n$ nodes generated as follows. There is some unknown partition of $[n]$ into $k$ sets $S_1, \ldots, S_k$. Given parameters $a, b$ with $a > b$, a graph is then generated where nodes in the same community are connected with probability $a/n$ and nodes in different communities are connected with probability $b/n$ (all edges are sampled independently). An adversary may then

- Arbitrarily add additional edges within communities and remove edges between different communities
- Pick up to $\epsilon n$ nodes and modify their incident edges arbitrarily

**Remark.** In later sections, we will often just say \textit{$\epsilon$-corrupted SBM} instead of \textit{$\epsilon$-Corrupted Semi-Random SBM} but it will always refer to an SBM with both adversarial corruptions and semi-random noise.

The goal of the learner is to observe a graph generated from an \textit{$\epsilon$-Corrupted} Semi-random SBM and output a partition of $[n]$ that is close to the unknown partition. Formally, if the learner outputs $\hat{S}_1, \ldots, \hat{S}_k$ and the true partition is $S_1, \ldots, S_k$ then the error is the minimum number of errors over all permutations of the sets i.e.

$$err = \min_{\pi:|k|\rightarrow|k|} \left( \sum_{i=1}^{k} |S_i \Delta \hat{S}_\pi(i)| \right).$$

We will use the term accuracy for $1 - err$.

We will assume that the learner is given the parameters $a, b, k, \epsilon$ and also a parameter $\alpha$ such that $\alpha n/k \leq |S_i| \leq n/(\alpha k)$ for all $i$ (so $\alpha$ bounds the imbalance in the community sizes). Note that even without corruptions (but with semi-randomness), there are known obstacles to recovering the planted partition without knowledge of the parameters [33]. Of course, if the parameters are unknown, we can simply guess them using a grid and output a list of candidate partitions at least one of which must have high accuracy.

B. $\mathbb{Z}_2$-Synchronization

Next, we define the problem of $\mathbb{Z}_2$-Synchronization.

**Definition II.4 (\textit{$\mathbb{Z}_2$-Synchronization}).** We are given an $n \times n$ matrix $A$ generated as follows. There is an unknown sign vector $\ell \in \{-1, 1\}^n$. We then observe $\lambda \ell^T / \sqrt{n} + E$ where $\lambda$ is some parameter and $E$ has entries drawn i.i.d from $N(0, 1)$.

Similar to before for SBMs, we can define a natural extension of the above model that allows for semi-random noise.

**Definition II.5 (\textit{Semi-random $\mathbb{Z}_2$-Synchronization}).** We are given an $n \times n$ matrix $A$ generated as follows. There is an unknown sign vector $\ell \in \{-1, 1\}^n$. We then observe $\lambda \ell^T / \sqrt{n} + E$ where $\lambda$ is some parameter. $E$ has entries drawn i.i.d from $N(0, 1)$ and $F$ has the same signs as $\ell \ell^T$ (entrywise).

Finally, we define a model that allows for an $\epsilon$-fraction of adversarial corruptions as well as semi-random noise.

**Definition II.6.** [\textit{$\epsilon$-Corrupted Semi-random $\mathbb{Z}_2$-Synchronization}] We are given an $n \times n$ matrix $A$ generated as follows. There is an unknown sign vector $\ell \in \{-1, 1\}^n$. Let $E$ be an $n \times n$ matrix whose entries are drawn i.i.d from $N(0, 1)$. Let $A_0 = \lambda \ell^T / \sqrt{n} + E$. Now an adversary may modify $A_0$ by
• Adding a matrix $F$ whose entries have the same signs are $\ell \mathbb{F}$
• Picking up to $\ell n$ elements of $[n]$ and modifying the corresponding rows and columns arbitrarily

We observe the matrix $A$ after the adversary makes the above modifications to $A_0$.

As usual, the goal of the learner is to observe $A$ generated as above and output a partition of $[n]$ that is close to the unknown partition given by the signs of $\ell$ where error is defined as the minimum disagreement up to flipping the components of the partition.

III. WHAT IS THE RIGHT ACCURACY?

Our goal is to give algorithms that achieve nearly optimal error in the presence of corruptions and semi-random noise. We first discuss prior work that characterizes the optimal error in a non-robust setting i.e. without corruptions or semi-random noise.

A. Community Detection

In [36], the authors characterize the optimal accuracy achievable information-theoretically in a pure SBM (with no semi-random noise or corruptions) as the signal-to-noise ratio goes to infinity.

**Theorem III.1** ([36]). Consider a (pure) SBM on $n$ nodes with $k$ communities with edge probabilities $a/n$ and $b/n$. Also assume that all communities have sizes between $an/k$ and $n/(ak)$. Assume that $a, b = o(n)$ and define $C = (\sqrt{a - \sqrt{b}})^2$. Then as $C/(k \log k)$ → $\infty$ any algorithm must incur expected error at least

$$e^{-(1+o(1))\frac{C}{k}}$$ if $k = 2$

$$e^{-(1+o(1))\frac{C}{k}}$$ if $k \geq 3$

where the $o(1)$ is some quantity that goes to 0 as $C/(k \log k) \rightarrow \infty$.

Note that the factor of $\alpha$ shows up only when $k \geq 3$ because then there can be two small communities of size $an/k$ but this cannot happen for $k = 2$. In [36], the authors also prove that the above is tight when $\alpha \geq \sqrt{3}/5$ in the sense that it is indeed possible to achieve the accuracy specified in Theorem III.1. However, their proof is only information-theoretic and they do not give a polynomial time algorithm that achieves this.

In [15], the authors give a polynomial time algorithm for matching the accuracy in Theorem III.1 for $k = 2$ and balanced communities. Their algorithm is based on the max-cut SDP and also works in the semi-random model.

**Theorem III.2** (Informal [15]). Consider a (pure or semi-random) SBM on $n$ nodes with two balanced communities with edge probabilities $a/n$ and $b/n$. Assume that $a, b = o(n)$ and define $C = (\sqrt{a - \sqrt{b}})^2$. There is an algorithm that achieves error $e^{-C/2+O(\sqrt{C})}$ with $1 - o(1)$ probability.

There are a few additional technical conditions in their theorem such as the fact that the implicit constant in the $O(\sqrt{C})$ may depend on the ratio $a/b$ (but for say fixed $a, b$, it is a universal constant). Nevertheless, as far as we are aware, this is the best known explicit bound on the classification accuracy in an SBM in terms of the signal-to-noise ratio.

Our main theorems, stated below, essentially match this guarantee but are significantly more general – they work for imbalanced and more than two communities and in the presence of a $\epsilon$-fraction of adversarial corruptions.

**Theorem III.3** (Robust Community Detection with $k = 2$). There is a polynomial-time algorithm that when run on an $\epsilon$-corrupted semi-random SBM with $n$ vertices, edge probabilities $b/n < a/n \leq 1/2$ and $k = 2$ communities of sizes between $an/2$ and $n/(2\alpha)$, outputs a labelling that has expected error at most

$$O(\epsilon \alpha^{-3}) = e^{-C/2} + O(\alpha^{-3} \sqrt{C} \log C) + \frac{e^{-\log n}}{n}$$

where $C = (\sqrt{a} - \sqrt{b})^2$.

**Theorem III.4** (Robust Community Detection with $k \geq 3$). There is a polynomial-time algorithm that when run on an $\epsilon$-corrupted semi-random SBM with $n$ vertices, edge probabilities $b/n < a/n \leq 1/2$ and $k$ communities of sizes between $an/k$ and $n/(k\alpha)$, outputs a labelling that has expected error at most

$$O(\epsilon k / \alpha^3) + e^{-\alpha C/k + poly(k/\alpha) \sqrt{C} \log C} + \frac{e^{-\log n}}{n}$$

where $C = (\sqrt{a} - \sqrt{b})^2$.

In Theorems III.3 and III.4, the hidden constants in the $O(\cdot)$ and $\alpha^{-3}$ are all universal constants. We imagine that $\alpha, k$ are fixed constants and that $C$ is sufficiently large as a function of $\alpha, k$. We take $n \rightarrow \infty$. $C$ may be held constant as $n$ grows or it may grow with $n$. Our error can be decomposed as follows. The $O(\epsilon)$ term comes from the corruptions. The exponential term comes from the error that must be incurred, even without any corruptions. Note that the leading terms in the exponents in our error guarantees $-C/2$ for $k = 2$ and $-\alpha C/k$ for $k \geq 3$ – are sharp in that they exactly match those in Theorem III.1. The last term is $o(1/n)$ so it contributes no additional errors with high probability.

B. $\mathbb{Z}_2$-Synchronization

The paper [15] also gives recovery guarantees for $\mathbb{Z}_2$-synchronization based on the max-cut SDP. They also prove a lower-bound that nearly matches their recovery guarantee.

**Theorem III.5** (Informal [15]). Consider a (pure or semi-random) $\mathbb{Z}_2$-synchronization instance on $n$ variables with parameter $\lambda$. Then there is an algorithm that achieves error $e^{-\lambda \sqrt{2}} + O(\lambda)$ with $1 - o(1)$ probability. Furthermore, no algorithm can achieve error better than $e^{-\lambda (1 + o(1)) \sqrt{2}}$.

Again, for $\mathbb{Z}_2$-synchronization, we are able to essentially match this guarantee, but robustly, in the presence of an $\epsilon$-fraction of adversarial corruptions.
Theorem III.6 (Robust $\mathbb{Z}_2$-Synchronization). There is a polynomial-time algorithm that when run on an $\epsilon$-corrupted semi-random $\mathbb{Z}_2$-synchronization instance with parameter $\lambda$, outputs a labelling that has expected error at most

$$O(\epsilon) + e^{-\lambda^2/2 + O(\lambda)} + \frac{e^{-\sqrt{\log n}}}{n}.$$ \[
\]

As before, all of the hidden constants in the $O(\cdot)$ are universal constants. We assume that $\lambda$ is at least some sufficiently large universal constant. We imagine taking $n \rightarrow \infty$ and $\lambda$ may be held constant or it may grow with $n$. As before, our error guarantee can be decomposed into the $O(\epsilon)$ term for the corruptions and the exponential term for the error that must be incurred even without any corruptions. The leading term of $-\lambda^2/2$ in the exponent is sharp as it matches the lower bound in Theorem III.5.

IV. TECHNICAL OVERVIEW

We now give an overview of our techniques. For simplicity, consider the case $k = 2$ and assume that the communities are balanced. Let the inter-community and intra-community connection probabilities be $a/n$ and $b/n$ respectively and define $C = (\sqrt{a} - \sqrt{b})^2$. A standard approach to achieving strong accuracy guarantees in community detection is to first obtain a rough labelling and then boost it. In our case, for the rough labelling, there is some additional work required to achieve robustness to node corruptions (see the full version for more details). However the boosting step is the key component and we will focus on that here. A natural first attempt would be to simply boost the accuracy with majority voting i.e. we set each node’s label to agree with the majority of its neighbors. Without corruptions, it turns out that if we have a rough clustering with $1/\sqrt{C}$ error (which our rough clustering algorithm does obtain) then one step of boosting will already get to $e^{-C/2 + O(\sqrt{C})}$ error. However, corruptions make the problem significantly more difficult. In particular it turns out that the adversary can make it so that majority voting gets most of the labels wrong, even while keeping the degrees of the nodes fixed:

Observation IV.1 (Informal). Consider an SBM with edge probabilities $a/n, b/n$ and assume that we are given the true labelling. An adversary can choose $cn$ nodes and corrupt them while preserving their degrees so that majority voting gets most labels wrong as long as $\epsilon \geq \frac{2(a-b)}{a+b}$.

The adversary can accomplish this by picking $cn$ nodes and reconnecting all of their edges to random nodes in the opposite community. Now, on average, nodes have $(1-\epsilon)a/2$ neighbors within their community and $(1-\epsilon)b/2 + \epsilon(a+b)/2$ neighbors in the opposite community and the latter quantity is larger by the definition of $\epsilon$.

This precludes obstacle classes of “local” algorithms that attempt to label each node based solely on information within its neighborhood. In particular, such algorithms cannot get guarantees that depend only on the signal-to-noise ratio $C$ because, in the argument above, by increasing the degrees $a,b$ while holding $C$ fixed, the fraction of corruptions that can be tolerated goes to 0. Thus we will need to exploit global information in order to get a stronger and more robust boosting algorithm.

At a high-level we will design a suitable “stability” property with respect to a graph (given by its adjacency matrix $A$) and a labelling of its vertices given by $\ell \in \{-1,1\}^n$. It will have the key properties that

- The subgraph of uncorrupted nodes with correct labels will satisfy this stability property.
- A subgraph with too many mislabeled nodes will violate this stability property.

Our algorithm will proceed as follows: we find the largest subgraph for which the stability property is satisfied. We argue that this must give us essentially only the correctly labeled nodes. We then flip the labels on all of the nodes outside this subgraph and argue that this must boost the accuracy because almost all of those nodes must have been mislabeled to begin with.

a) Stability: Now we discuss the stability property that we use in more detail. First, consider an SBM (i.e. with no corruptions) and let $A$ be its adjacency matrix. Let $\hat{A}$

$$\hat{A} = A - \frac{a+b}{2} J.$$ \[
\]

Let the true labelling of the vertices be given by a vector $\ell \in \{-1,1\}^n$ and define the matrix $\hat{L} = \tilde{L}^T$. The first key observation is that the entries of $\hat{A} \circ \hat{L}$ have positive expectation, where $\circ$ denotes the Hadamard product. It can be shown that for any $\gamma n \times (1-10\gamma)n$ combinatorial rectangle with $\gamma \geq e^{-C+O(\sqrt{C})}$, the sum of the entries of $\hat{A} \circ \hat{L}$ is positive with high probability. In particular the lower bound on $\gamma$ comes from tail bounds on the binomial distribution i.e. $\sim e^{-C+O(\sqrt{C})}$ fraction of the rows may actually have negative sum. This lower bound on $\gamma$ is also exactly what shows up in our accuracy guarantee. For the purposes of this overview, we will say that a matrix $\hat{A}$ is $\gamma$-stable with respect to a labelling $\hat{L}$ if the sum of the entries of $\hat{A} \circ \hat{L}$ over any $\gamma n \times (1-10\gamma)n$ combinatorial rectangle is positive.

Now we have a global notion of stability, involving subsets of $\gamma n$ rows, that we know $\hat{A}$ must satisfy. This notion of stability is also robust to a small fraction of corruptions. In particular, imagine that some matrix $\hat{A}$ is $\gamma$-stable with respect to the labelling $\hat{L}$ where some of the nodes in $\hat{A}$ may be corrupted. Because we enforce that the sum over any $\gamma n \times (1-10\gamma)n$ combinatorial rectangle of $\hat{A} \circ \hat{L}$ is positive, we can consider rectangles whose rows and columns correspond only to the uncorrupted nodes. Thus, the submatrix of uncorrupted nodes must be stable as well (with a slightly worse parameter $\gamma$). In other words, a small fraction of corruptions cannot hide an unstable submatrix – this is the key for dealing with adversarial corruptions.

\footnote{For technical reasons, in the full proof we will use a slightly different adjustment to demean $\hat{A}$ but the definition here suffices for the purposes of the overview}
b) Boosting Program: We can now describe our boosting procedure. We take as inputs an adjacency matrix \( \hat{A} \) and a rough labelling \( \ell \in \{-1, 1\}^n \). Let \( \theta \) be the error of \( \ell \) with respect to the true labelling. We may assume that \( \theta \) is significantly larger than \( \gamma \) and \( \epsilon \) (the fraction of corruptions), since otherwise we can just terminate and output \( \ell \). Consider the following program:

**Definition IV.2 (Boosting Program (Informal)).** Assume that we have a rough labelling of the vertices given by a vector \( \ell \in \{-1, 1\}^n \) and define the matrix \( L = \ell \ell^T \). We solve for a weight vector \( w \in \mathbb{R}^n \) with entries between 0 and 1 such that

- In the matrix \( \hat{A} \odot L \odot ((1 - w)(1 - w)^T) \), the sum of the entries over any \( \theta n \times (1 - 10\theta) n \) combinatorial rectangle is positive

Our objective is to minimize \( \sum_i w_i \).

The above program is nonconvex, and ultimately we will need to work with a convex relaxation. The key step in the relaxation is to replace \( (1 - w)(1 - w)^T \) with a matrix

\[
W = J - [w_1 \mathbf{1} \ldots w_n \mathbf{1}] - \begin{bmatrix} w_1 \mathbf{1}^T \\ \vdots \\ w_n \mathbf{1}^T \end{bmatrix} + N \tag{2}
\]

with a constraint on trace norm \( \|N\|_1 \). We also relax the notion of a combinatorial rectangle, replacing it with a matrix \( N \) and a constraint on \( \|N\|_1 \). Note that trace norm constraints make intuitive sense for these relaxations because combinatorial rectangles are rank-1 and the trace norm is a useful convex relaxation for rank constraints. There are several additional technical details that we need to deal with and constraints that we need to enforce but we will not discuss those here. The precise relaxed constraints and full semidefinite programming relaxation can be found in the full version of this paper.

The intended solution to the boosting program is \( w = w_{\text{base}} \) where \( w_{\text{base}} \) is the vector with 1s in entries corresponding to corrupted or mislabeled nodes and 0s in entries corresponding to uncorrupted and correctly labeled nodes. Once we obtain a solution \( w \), we simply flip the labels on all of the vertices where \( w \) has large weight. If \( w \) is close to \( w_{\text{correct}} \), then this procedure will boost the accuracy because we are correcting essentially all of the mistakes in the original labelling.

**c) Analysis:** It remains to argue that the solution that we obtain must be close to \( w_{\text{base}} \). We do this in two steps. We first prove that the intended solution \( w_{\text{base}} \) is indeed feasible. Informally:

**Lemma IV.3 (Informal).** \( w_{\text{base}} \) is a feasible solution to the boosting program

For the non-convex boosting program in Definition IV.2, to argue that \( w_{\text{base}} \) is feasible, it suffices to consider \( \hat{A} \), constructed from (1) for an SBM (since \( (1 - w_{\text{base}})(1 - w_{\text{base}})^T \) zeros out all corrupted nodes), and prove that it has nonnegative sums over all \( \theta n \times (1 - 10\theta) n \) combinatorial rectangles. Arguing about the relaxation requires more work. The high level idea is to combine the trace constraints (for relaxed combinatorial rectangles) with spectral bounds on \( A \) to bound their inner product. In the constant-degree regime, \( A \) does not actually satisfy the necessary spectral bounds (because nodes may have logarithmic degree). Fortunately, we are able to appeal to results in [8] showing there is a large submatrix of \( A \), corresponding to all nodes whose degree is not too high, that does satisfy the necessary spectral bounds. This spectral bound on a large submatrix of \( A \) turns out to suffice for our argument.

The second step in the analysis involves characterizing the structure of the optimal solution. Informally:

**Lemma IV.4 (Informal).** Any feasible solution to the boosting program with objective value at most \( 20 \theta \) must satisfy that there are at most \( 0.1\theta \) nodes \( i \in [n] \) that are mislabeled by \( \ell \) and have \( w_i \leq 0.9 \).

The key intuition is that any solution must place essentially no weight on the mislabeled nodes because otherwise the combinatorial rectangle sum constraints of the program will be violated. In particular, let \( S \) be the set of mislabeled, uncorrupted nodes and \( R \) be the set of correctly labeled, uncorrupted nodes. Then the matrix \( \hat{A} \odot L \) has entries with negative mean on the set indexed by \( S \times R \). Thus, if \( w \) is not close to 1 on the mislabeled nodes, then the sum of the entries of

\[
\hat{A} \odot L \odot ((1 - w)(1 - w)^T)
\]

will be negative, violating the constraint of the program. The full proof requires significantly more care because we replace \( (1 - w)(1 - w)^T \) with a matrix \( W \) with certain linear and trace norm constraints (recall (2)). Nevertheless, we use the decomposition of \( W \) in (2) and show that, combined with the spectral properties of \( A \), an analog of the above argument can be pushed through.

Finally, it remains to combine Lemma IV.3 and Lemma IV.4 to complete the analysis of our boosting procedure. The key point is that the two lemmas together imply that the optimal solution (where recall we are minimizing the total weight) must be close to \( w_{\text{base}} \) because

- \( w_{\text{base}} \) is feasible and places weight only on mislabeled and corrupted nodes
- Our solution \( w \) must place essentially full weight on all of the mislabeled nodes

Thus, almost all of the nodes with low weight in the optimal solution must be mislabeled and so flipping their labels will actually reduce the error by a constant factor. Ideally, we might hope that the boosting procedure achieves the optimal error in one shot. However, this is not the case because there can be a constant fraction (e.g. 0.1) difference between \( w \) and \( w_{\text{base}} \). Still, we can then simply iterate this boosting step, re-solving the program again with the new labelling. We show that we can keep iterating until we get down to the optimal error of

\[
\sim \gamma + \epsilon.
\]

d) Semi-Random Noise: Working in the semi-random model can foil many natural algorithms. In particular, it is no longer possible to enforce the same degree constraints or
satisfy spectral constraints on the adjacency matrix. For our analysis, however, dealing with semi-random noise follows almost immediately. The crucial property about the way we formulated the boosting program is that we know the signs of the entries of all of the matrices that appear, namely \((1-w)(1-w^T)\) and \(L\). Thus, we can explicitly reason about locations where the semi-random noise is positive or negative.

More specifically, recall that for Lemma IV.3, we consider \(w = w_{\text{base}}\) which is the indicator vector of the corrupted and mislabeled nodes. Thus, if we let \(F\) denote the matrix of semi-random changes i.e. entries of \(F = 1\) if the corresponding edge was added, \(-1\) if the edge was removed, and \(0\) otherwise, then

\[
F \odot L \odot ((1-w)(1-w)^T)
\]

is entry-wise nonnegative. The monotone changes only strengthen the inequalities that our program enforces. For Lemma IV.4, we restrict to a rectangle \(S \times R\) where the nodes in \(S\) are incorrectly labeled and the nodes in \(R\) are correctly labeled. Thus,

\[
F \odot L
\]

is entrywise nonpositive on \(S \times R\). Since \((1-w)(1-w^T)\) is entrywise nonnegative, if a constraint involving the rectangle \(S \times R\) were violated before, the monotone changes only make the violated constraints worse, and consequently the same argument still works.

### A. What Happens with Edge Corruptions?

Several other papers [6], [14], [26], [35] have considered a model where instead of node corruptions, we allow for edge corruptions, where an \(\epsilon\)-fraction of the edges may be corrupted. However, there are several information-theoretic barriers to obtaining high accuracy with edge corruptions and thus the results in these papers necessarily have weaker guarantees. First note that

**Observation IV.5.** For \(\epsilon \geq \frac{2(a-b)}{ab}\), an SBM with edge probabilities \(a/n, b/n\) where we an adversary may corrupt an \(\epsilon\)-fraction of the edges is statistically indistinguishable from \(G(n, (a + b)/(2n))\).

The above holds simply because the adversary may delete edges within communities and add edges between different communities so that the effective edge probabilities are both \((a + b)/(2n)\). Thus, we run into the same barrier as in Observation IV.1 for local algorithms – it is impossible to obtain accuracy guarantees that depend only on the signal-to-noise ratio \(C\) because the fraction of corruptions that can be tolerated goes to 0 as \(a, b\) increase while holding \(C\) fixed.

It is also worth noting that the edge-corruption model does not combine nicely with semi-random noise. In particular, the fraction of edge corruptions can only scale with the number of edges in the pure SBM and not the total number of edges. This is because the semi-random noise may simply add cliques on a small set of vertices. In particular, in the sparse regime where \(a, b\) are constant, the total number of edges is \(\Theta(n)\). If the semi-random noise adds a clique on \(O(\sqrt{n})\) vertices, then the adversary is now allowed enough corruptions to erase the entire graph on the remaining vertices.

**Observation IV.6.** With semi-random noise added to an SBM with edge probabilities \(a/n, b/n\), it is impossible to achieve accuracy better than 0.5 if an adversary may corrupt an \(\epsilon\)-fraction of the total number of edges.

### B. Full Paper

Due to space constraints, the full proofs are deferred to the full version which can be found at https://arxiv.org/abs/2207.11903.

### References

[1] Emmanuel Abbe. Community detection and stochastic block models: recent developments. *The Journal of Machine Learning Research*, 18(1):6446–6531, 2017.

[2] Emmanuel Abbe, Afonso S Bandeira, and Georgina Hall. Exact recovery in the stochastic block model. *IEEE Transactions on information theory*, 62(1):471–487, 2015.

[3] Jayadev Acharya, Ayush Jain, Gautam Kamath, Ananda Theertha Suresh, and Huanyu Zhang. Robust estimation for random graphs. *arXiv preprint arXiv:2111.05320*, 2021.

[4] Aimesh Bakshi, Ilias Diakonikolas, He Jia, Daniel M. Kane, Pravesh K. Kothari, and Santosh S. Vempala. Robustly learning mixtures of \(k\) arbitrary gaussians, 2020.

[5] Aimesh Bakshi and Adarsh Prasad. Robust linear regression: Optimal rates in polynomial time. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 102–115, 2021.

[6] Jess Banks, Sidhanth Mohanty, and Prasad Raghavendra. Local statistics, semidefinite programming, and community detection. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1298–1316. SIAM, 2021.

[7] Sitin Chen, Frederic Koehler, Ankur Moitra, and Morris Yau. Online and distribution-free robustness: Regression and contextual bandits with huber contamination. *arXiv preprint arXiv:2010.04157*, 2020.

[8] Peter Chin, Anup Rao, and Van Vu. Stochastic block model and community detection in sparse graphs: A spectral algorithm with optimal rate of recovery. In *Conference on Learning Theory*, pages 391–423. PMLR, 2015.

[9] Aurelien Decelle, Florent Krzakala, Cristopher Moore, and Lenka Zdeborová. Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. *Physical Review E*, 84(6):066106, 2011.

[10] Yash Deshpande, Emmanuel Abbe, and Andrea Montanari. Asymptotic mutual information for the two-groups stochastic block model. *arXiv preprint arXiv:1307.08685*, 2015.

[11] Ilias Diakonikolas, Gautam Kamath, Daniel Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Sever: A robust meta-algorithm for stochastic optimization. In *International Conference on Machine Learning*, pages 1596–1606. 2019.

[12] Ilias Diakonikolas, Gautam Kamath, Daniel Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Recent developments.

[13] Ilias Diakonikolas, Gautam Kamath, Daniel M Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Being robust (in high dimensions) can be practical. In *International Conference on Machine Learning*, pages 999–1008. PMLR, 2017.

[14] Jingtong Ding, Tommaso d’Orsi, Rajai Nasser, and David Steurer. Robust recovery for stochastic block models. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 387–394. IEEE, 2022.

[15] Yingjie Fei and Yudong Chen. Achieving the bayes error rate in synchronization and block models by sdp, robustly. *IEEE Transactions on Information Theory*, 66(6):3929–3953, 2020.

[16] Uriel Feige and Joe Kilian. Heuristics for semirandom graph problems. *Journal of Computer and System Sciences*, 63(4):639–671, 2001.

[17] Olivier Guédon and Roman Vershynin. Community detection in sparse networks via grothendieck’s inequality. 2014.

830
[18] Bruce Hajek, Yihong Wu, and Jiaming Xu. Achieving exact cluster recovery threshold via semidefinite programming. *IEEE Transactions on Information Theory*, 62(5):2788–2797, 2016.

[19] Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137, 1983.

[20] Samuel B Hopkins and Jerry Li. Mixture models, robustness, and sum of squares proofs. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1021–1034. ACM, 2018.

[21] Samuel B. Hopkins and David Steurer. Bayesian estimation from few samples: community detection and related problems. *CoRR*, abs/1710.00264, 2017.

[22] Adam Klivans, Pravesh K Kothari, and Raghu Meka. Efficient algorithms for outlier-robust regression. In *Conference On Learning Theory*, pages 1420–1430, 2018.

[23] Kevin A Lai, Anup B Rao, and Santosh Vempala. Agnostic estimation of mean and covariance. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 665–674. IEEE, 2016.

[24] Allen Liu and Ankur Moitra. Learning gmms with nearly optimal robustness guarantees. *CoRR*, abs/2104.09665, 2021.

[25] Allen Liu and Alexander S Wein. How robust are reconstruction thresholds for community detection? In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 828–841, 2016.

[26] Laurent Massoulié. Community detection thresholds and the weak ramanujan property. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 694–703, 2014.

[27] Elchanan Mossel, Joe Neeman, and Allan Sly. A proof of the block model threshold conjecture. *Combinatorica*, 38(3):665–708, 2018.

[28] Alexei Onatski, Marcelo J Moreira, and Marc Hallin. Asymptotic power of sphericity tests for high-dimensional data. *The Annals of Statistics*, 41(3):1204–1231, 2013.

[29] Amelia Perry and Alexander S Wein. A semidefinite program for unbalanced multisection in the stochastic block model. In *2017 International Conference on Sampling Theory and Applications (SampTA)*, pages 64–67. IEEE, 2017.

[30] Elchanan Mossel, Afonso S Bandeira, and Ankur Moitra. Optimality and sub-optimality of pca i: Spiked random matrix models. *The Annals of Statistics*, 46(5):2416–2451, 2018.

[31] Liu, Xiaodi Zhao, and Wenchao Zhang. Robustness of spectral methods for community detection. In *Conference on Learning Theory*, pages 2831–2860. PMLR, 2019.

[32] Anderson Y Zhang and Harrison H Zhou. Minimax rates of community detection in stochastic block models. *The Annals of Statistics*, 44(5):2252–2280, 2016.