How Robust are Reconstruction Thresholds for Community Detection?

Ankur Moitra[*1,2], William Perry[†1], and Alexander S. Wein[‡1]

[1]Massachusetts Institute of Technology, Department of Mathematics
[2]Massachusetts Institute of Technology, Computer Science and Artificial Intelligence Lab

March 23, 2016

Abstract

The stochastic block model is one of the oldest and most ubiquitous models for studying clustering and community detection. In an exciting sequence of developments, motivated by deep but non-rigorous ideas from statistical physics, Decelle et al. [DKMZ11] conjectured a sharp threshold for when community detection is possible in the sparse regime. Mossel, Neeman and Sly [MNS14b] and Massoulié [Mas14] proved the conjecture and gave matching algorithms and lower bounds.

Here we revisit the stochastic block model from the perspective of semirandom models where we allow an adversary to make 'helpful' changes that strengthen ties within each community and break ties between them. We show a surprising result that these 'helpful' changes can shift the information-theoretic threshold, making the community detection problem strictly harder. We complement this by showing that an algorithm based on semidefinite programming (which was known to get close to the threshold) continues to work in the semirandom model (even for partial recovery). This suggests that algorithms based on semidefinite programming are robust in ways that *any* algorithm meeting the information-theoretic threshold cannot be.

These results point to an interesting new direction: Can we find robust, semirandom analogues to some of the classical, average-case thresholds in statistics? We also explore this question in the broadcast tree model, and we show that the viewpoint of semirandom models can help explain why some algorithms are preferred to others in practice, in spite of the gaps in their statistical performance on random models.

1 Introduction

1.1 Background

The stochastic block model is one of the oldest and most ubiquitous models for studying clustering and community detection. It was first introduced by Holland et al. [HLL83] more than thirty years ago and since then it has received considerable attention within statistics, computer science, statistical physics and information theory. The model defines a procedure to generate a random graph: first, each of the n nodes is independently assigned to one of r communities, where p_i is the probability of being assigned to community i. Next, edges are sampled independently based on the community assignments: if nodes u and v belong to communities i and j respectively, the edge (u, v) occurs with probability Q_{ij} independent of all other edges, where Q is an $r \times r$ symmetric matrix. The goal is to recover the community structure — either exactly or approximately — from the graph.

Since its introduction, the stochastic block model has served as a testbed for the diverse range of algorithms that have been developed for clustering and community detection, including combinatorial methods [BCLS87], spectral methods [McS01], MCMC [JS98], semidefinite programs [Bop87] and belief propagation [MNS14a]. Recently, the stochastic block model has been thrust back into the spotlight, the goal being to establish tight thresholds for when community detection is possible, and to find algorithms that achieve them. Towards this end, Decelle et al. [DKMZ11] made some fascinating conjectures (which have since been resolved) in the case of two equal-sized communities with constant average degree, that we describe next.

Throughout this paper, we will focus on the two-community case and use a/n and b/n to denote the within-community and between-community connection probabilities respectively. We will be interested in the sparse setting where $a, b = \Theta(1)$. Moreover we set $p_1 = p_2 = 1/2$, so that the two communities are roughly equal-sized, and we assume $a > b$ (although we discuss the case $a < b$ in Appendix D). We will use $G(n, a/n, b/n)$ to denote the corresponding random graph model. The setting of parameters above is particularly well-motivated in practice, where a wide variety of networks have been observed [LLDM08] to have average degree that is bounded by a small constant. It is important to point out that when the average degree is constant, it is impossible to recover the community structure exactly in the stochastic block model because a constant fraction of nodes will be isolated. Instead, our goal is to recover a partition of the nodes into two communities that has non-trivial agreement (better than random guessing) with the true communities as $n \to \infty$, and we refer to this as *partial recovery*.

Conjecture 1.1. *[DKMZ11] If $(a-b)^2 > 2(a+b)$ then partial recovery in $G(n, a/n, b/n)$ is possible, and if $(a-b)^2 < 2(a+b)$ then partial recovery is information-theoretically impossible.*

This conjecture was based on deep but non-rigorous ideas originating from statistical physics, and was first derived heuristically as a stability criterion for belief propagation. This threshold also bears a close connection to known thresholds in the broadcast tree model, which is another setting in which to study partial recovery. We formally define the broadcast tree model in Section 2.1, but it is a stochastic process described by two parameters a and b. Kesten and Stigum [KS66] showed that partial recovery is possible if $(a-b)^2 > 2(a+b)$, and much later Evans et al. [EKPS00] showed that it is impossible if $(a - b)^2 \leq 2(a + b)$. The connection between the two models is that in the stochastic block model, the local neighborhood of a node resembles the broadcast tree model, and this was another compelling motivation for the conjecture.

In an exciting sequence of developments, Mossel, Neeman and Sly [MNS14b] proved a lower bound that even distinguishing $G(n, a/n, b/n)$ from an Erdős–Rényi graph $G(n, (a + b)/2n)$ is

information-theoretically impossible if $(a - b)^2 \leq 2(a + b)$, by a careful coupling to the broadcast tree model. Subsequently Mossel, Neeman and Sly [MNS13] and Massoulié [Mas14] independently gave matching algorithms that achieve partial recovery up to this threshold, thus resolving the conjecture! Mossel, Neeman and Sly [MNS14a] later showed that for some constant $C > 0$, if $(a - b)^2 > C(a + b)$ then belief propagation works and moreover the agreement of the clustering it finds is the best possible.

In fact, many other sorts of threshold phenomena have been found in different parameter regimes. Abbe, Bandeira and Hall [ABH15] studied exact recovery in the logarithmic degree setting $G(n, a \log n/n, b \log n/n)$ and showed that it is efficiently possible to recover the two communities exactly if $(\sqrt{a} - \sqrt{b})^2 > 2$ and information-theoretically impossible if $(\sqrt{a} - \sqrt{b})^2 < 2$. Abbe and Sandon [AS15] gave a precise characterization of when exact recovery is possible in the general stochastic block model for more than two communities with arbitrary relative sizes and arbitrary connection probabilities.

1.2 Semirandom Models

This abundance of sharp thresholds begs a natural question: how robust are these reconstruction thresholds? It is clear that if one substantially changes the distributional model, the thresholds themselves are likely to change. However there is a subtler issue. The algorithms that achieve these thresholds may in principle be over-fitting to a particular distributional model. Random graphs are well-known [AS00] to have rigid properties, such as sharp laws for the distribution of subgraph counts and a predictable distribution of eigenvalues. Real-world graphs do not have such properties.

In a remarkable paper, Blum and Spencer [BS95] introduced the semirandom model as an intermediary between average-case and worst-case analysis, to address such issues. The details of the model vary depending on the particular optimization problem, and since we will focus on clustering we will be most interested in the variant used by Feige and Kilian [FK01].

Semirandom Model for Community Detection: [FK01]

- Sample a 'precursor' graph G_{pre} from $G(n, a/n, b/n)$.

- A monotone 'adversary' observes G_{pre} along with the hidden community structure, and can delete any number of edges crossing between the two communities, and add any number of edges with each community.

- Output the resulting graph G.

The adversary above is called 'monotone' because it is restricted to making changes that seem to be helpful. It can only strengthen ties within each community, and break ties between them.[1] The key is that a monotone adversary can break the sorts of rigid structures that arise in random graphs, such as predictable degree distributions and subgraph counts. An algorithm that works in a semirandom model can no longer rely on these properties. Instead of a graph containing *only* random edges, all we are assured of is that it contains *some* random edges.

In this paper, we use semirandom models as our notion of 'robustness.' Many forms of robustness exist in the literature – for example, the independent work [MMV15b] gives algorithms that are robust to $o(n)$ non-monotone changes in addition to monotone changes. With most robustness

---

[1]This is the reason we require $a > b$. If we had $b > a$ we could define the adversary in the opposite way (which we analyze in Appendix D).

models, any algorithm will break down after enough errors, and one can compare algorithms based on how many errors they can tolerate. In contrast, semirandom models distinguish between algorithms *qualitatively*: as we will see, entire classes of algorithms continue to work under any number of monotone changes, while others do not.

Feige and Kilian [FK01] showed that semidefinite programs for *exact recovery* in the stochastic block model continue to work in the semirandom model — in fact they succeed up to the threshold for the random model [HWX15]. Since then, there have been many further developments including algorithms that work in semirandom models for planted clique [FK00], unique games [KMM11], various graph partitioning problems [MMV12], correlation clustering [MS10, MMV15a] and the general planted partition model[2] [PW15] (and in some cases for even more powerful adversaries). A common theme is that if you have a semidefinite program that works in some stochastic setting, then it often extends almost automatically to the semirandom setting. So are there semidefinite programs that achieve the same sharp partial recovery results as Mossel, Neeman and Sly [MNS13] and Massoulié [Mas14], and that extend to the semirandom model too? Or is there a genuine gap between what is achievable in the random vs. the semirandom setting?

1.3   Our Results

Recall that in the semirandom block model, a monotone adversary observes a sample from the stochastic block model and is then allowed to add edges within each community and delete edges that cross between communities. We will design a particularly simple adversary to prevent an algorithm from utilizing the paths of length two that go from a '+' node to a '−' node and back to a '+' node, where the middle node has degree two. Our adversary will delete any such path it finds (with some additional technical conditions to locally coordinate these modifications), and our main result is that, surprisingly, this simple modification strictly changes the partial recovery threshold. We will state our bounds in terms of the average degree $k = \frac{a+b}{2}$ and 'noise' $\varepsilon = \frac{b}{a+b} \in [0, \frac{1}{2})$, in which case the threshold $(a - b)^2 > 2(a + b)$ becomes $k(1 - 2\varepsilon)^2 > 1$. Note that this threshold requires $k > 1$. Then:

Theorem 1.2 (main). *For any $k > 1$, there exists $0 < \varepsilon < \frac{1}{2}$ so that $k(1 - 2\varepsilon)^2 > 1$ and hence partial recovery in the stochastic block model $G(n, a/n, b/n)$ is possible, and yet there is a monotone adversary so that partial recovery in the semirandom model is information-theoretically impossible.*

A common tool in the algorithms of Mossel, Neeman and Sly [MNS13] and of Massoulié [Mas14] is the use of non-backtracking walks and spectral bounds for their transition matrices. Our adversary explicitly deletes a significant number of these walks. This simple modification not only defeats these particular algorithms, but we can show that *no* algorithm can achieve partial recovery up to the threshold $k(1 - 2\varepsilon)^2 > 1$. To the best of our knowledge, this is the first explicit separation between what is possible in the random model vs. the semirandom model, for *any* problem with a monotone adversary.

We show a complementary result, that no monotone adversary can make the problem too much harder. Various semidefinite programs have been designed for partial recovery. These algorithms work in the constant-degree regime, but are not known to work all the way down to the information-theoretic threshold. In particular, we follow Guédon and Vershynin [GV15] and show that their analysis (with a simple modification) works as is for the semirandom model. This shows that semidefinite programs not only give rise to algorithms that work for the semirandom model in the exact recovery setting, but also for partial recovery too under some fairly general conditions.

---

[2]In the general planted partition model, we have $Q_{ii} = a/n$ for all i and $Q_{ij} = b/n$ for all $i \neq j$, with $a > b$, but the number of communities and their relative sizes are arbitrary constants.

Theorem 1.3. *Let $k = \frac{a+b}{2}$ and $\varepsilon = \frac{b}{a+b} < \frac{1}{2}$. There is a constant $C > 1$ so that if $a > 20$ and $k(1-2\varepsilon)^2 > C$ then partial recovery is possible in the semirandom block model. Moreover it can be solved in polynomial time.*

Our robustness proof only applies to a particular form of SDP analysis. Given a different proof that the SDP succeeds in the random model for a larger range of parameters k, ε than above, there would be no guarantee that the SDP also succeeds in the semirandom model for that range of parameters. Hence we cannot formally conclude that it is impossible for the SDP to reach the information-theoretic threshold in the random model, though our results are suggestive in this direction.

We remark that each possible monotone adversary yields a new distribution on planted community detection problems. Hence, we can think of the algorithm in the theorem above as one that performs almost as well as information-theoretically possible across an entire family of distributions, simultaneously. This is a major advantage to algorithms based on semidefinite programming, and points to an interesting new direction in statistics. Can we move beyond average-case analysis? Can we find robust, semirandom analogues to some of the classical, average-case thresholds in statistics? The above two theorems establish upper and lower bounds for this semirandom threshold, that show that it is genuinely different from the average-case threshold. The usual notion of a threshold makes sense when we have exact knowledge of the process generating the instances we wish to solve. But when we lack this knowledge, semirandom models offer an avenue for exploration that can lead to new algorithmic and statistical questions.

Along the way to proving our main theorem, we show a random vs. semirandom separation for the broadcast tree model too. We define this model in Section 2.1. In short, it is a stochastic process in which each node is given one of two labels and gives birth to $\text{Pois}(a/2)$ nodes of the same label and $\text{Pois}(b/2)$ nodes of the different label, with the goal being to guess the label of the root given the labels of the leaves. There are many ways we could define a monotone adversary, and we focus on a particularly weak one that is only allowed to cut edges between nodes with different labels. We call this the cutting adversary, and we prove:

Theorem 1.4. *For any $k > 1$, there exists $0 < \varepsilon < \frac{1}{2}$ so that $k(1-2\varepsilon)^2 > 1$ and hence partial recovery in the broadcast tree model is possible, and yet there is a monotone cutting adversary for which partial recovery in the semirandom broadcast tree model is information-theoretically impossible.*

Furthermore we analyze the recursive majority algorithm and show that it is robust to an even more powerful class of monotone adversaries, which are allowed to entirely control the subtree at a node whose label is different than its parent. We call this a strong adversary, and we prove:

Theorem 1.5. *If $\frac{k}{\log k}(1-2\varepsilon)^2 > 1 + o(1)$ and $\varepsilon < \frac{1}{2}$ then partial recovery in the broadcast tree model is possible, even with respect to a strong monotone adversary, where 'o(1)' is taken as $k \to \infty$.*

These results highlight another well-studied model where the introduction of a monotone adversary strictly changes what is possible. Nevertheless there is an algorithm that succeeds across the entire range of distributions that arise from the action of a monotone adversary, simultaneously. Interestingly, our robustness results can also be seen as a justification for why practitioners use recursive majority at all. It has been known for some time that recursive majority does not achieve the Kesten–Stigum bound [Mos98] — the threshold for reconstruction in the broadcast tree model — although taking the majority vote of the leaves does. The advantage of recursive majority is that it is robust to very powerful adversaries while majority is not, and this only becomes clear when studying these algorithms through semirandom models!

2 Models and Adversaries

2.1 Preliminaries

Here we formally define the models we will be interested in, as well as the notion of partial recovery. Recall that $G(n, a/n, b/n)$ denotes the stochastic block model on two communities with $p_1 = p_2 = 1/2$ so that the communities are roughly equal sized. We will encode community membership as a *label* $\sigma_v \in \{+1, -1\}$ on each node v. We will also refer to this as a *spin*, following the convention in statistical physics. This numeric representation has the advantage that we can 'add' spins in order to take the majority vote, and 'multiply' them to compute the relative spin between a pair of nodes. We will be interested in the sparse regime $a, b = \Theta(1)$ where the graph has constant average degree, and we will assume $a > b$.

Next, we formally define partial recovery in the stochastic block model. Throughout this paper will will be interested in how our algorithms perform as n (number of nodes) goes to ∞. We say that an event holds a.a.s. (asymptotically almost surely) if it holds with probability $1 - o(1)$ as $n \to \infty$. Similarly, we say that an event happens for a.a.e. (asymptotically almost every) x if it holds with probability $1 - o(1)$ over a random choice of x.

Definition 2.1. We say that an assignment of $\{+1, -1\}$ spins to the nodes achieves η-**partial recovery** if at least ηn of these spins match the true spins, or at least ηn match after a global flip of all the spins. Moreover, an algorithm that outputs a vector of spins $\widehat{\sigma} \in \{+1, -1\}^n$ (indexed by nodes) is said to *achieve partial recovery* and there exists $\eta > \frac{1}{2}$ such that $\widehat{\sigma}$ achieves η-partial recovery a.a.s. in the limit $n \to \infty$.

Next, we define the broadcast tree model (which we introduced informally earlier). The broadcast tree model is a stochastic process that starts with a single root node ρ at level 0 whose spin $\sigma_\rho \in \{+1, -1\}$ is chosen uniformly at random. Each node in turn gives birth to $\text{Pois}(a/2)$ same-spin children and $\text{Pois}(b/2)$ opposite-spin children, where $\text{Pois}(c)$ is the Poisson distribution with expectation c. This process continues until level R at which point it stops, and the nodes at level R are called the *leaves*. (The nodes on level $\leq R - 1$ that by chance do not give birth to any children are not considered leaves, even though they are leaves in the graph-theoretic sense.) An algorithm observes the spins at the leaf nodes and the topology of the tree, and the goal is to recover the root spin:

Definition 2.2. An algorithm that outputs a spin $\widehat{\sigma}_\rho$ is said to *achieve partial recovery* on the tree if there exists $\eta > \frac{1}{2}$ such that $\widehat{\sigma}_\rho = \sigma_\rho$ with probability at least η, as $R \to \infty$.

The reparameterization in terms of (k, ε) becomes particularly convenient here: each node gives birth to $\text{Pois}(k)$ children, so $k = \frac{a+b}{2}$ is the average branching factor. Moreover, each child has probability $\varepsilon = \frac{b}{a+b}$ of having spin opposite to that of its parent.

It is known that taking the majority vote of the leaves is optimal in theory, in the sense that it achieves partial recovery for $k(1 - 2\varepsilon)^2 > 1$ [KS66] and that for $k(1 - 2\varepsilon)^2 \leq 1$, partial recovery is information-theoretically impossible [EKPS00]. This is called the Kesten–Stigum bound, and it can also be interpreted as a condition on the second-largest eigenvalue of an appropriately defined transition matrix. There are many other natural variants of the broadcast tree model, that are more general instances of multi-type branching processes. However, even for simple extensions, the precise information-theoretic threshold is still unknown. In the Potts model, where nodes are labeled with one of q labels, Sly [Sly11] showed that the Kesten–Stigum bound is not tight, as predicted by Mézard and Montanari [MM05]. And for an asymmetric extension of the binary model above, Borgs et al. [BCMR06] showed that the Kesten–Stigum bound is tight for some settings of

the parameters. In our setting, this historical context presents a substantial complication because if we apply a monotone adversary to a broadcast tree model and it results in a complex propagation rule, there may not be good tools to prove that partial recovery is impossible.

2.2 Discussion

There is a close connection between the stochastic block model and the broadcast tree model, since the local neighborhood of any vertex in the graph is locally tree-like. Hence, if our goal is to prove a random vs. semirandom gap for community detection, a natural starting point is to establish such a gap for the broadcast tree model. It turns out that there is more than one natural way to define a monotone adversary for the broadcast tree model, but it will not be too difficult to establish lower bounds for either of them. The more difficult task is in finding an adversary that can plausibly be coupled to a corresponding adversary in the stochastic block model, and this will require us to put many sorts of constraints on the type of adversary that we should use to obtain a separation for the broadcast tree model.

In the broadcast tree model, we will work with two notions of a monotone adversary. One is weak and will be used to show our separation results:

Cutting Semirandom Model for Broadcast Tree:

- Sample a 'precursor' broadcast tree T_{pre} from the broadcast tree model.

- The monotone cutting 'adversary' can delete any number of edges between nodes of different labels, thus removing subtrees.

- Output the resulting tree T. (The removed subtrees are not revealed.)

Our other adversary is stronger, and we will establish upper bounds against this adversary (with the recursive majority algorithm) to give a strong recoverability guarantee:

Strong Semirandom Model for Broadcast Tree:

- Sample a 'precursor' broadcast tree T_{pre} from the broadcast tree model.

- Whenever a child has the opposite label from its parent, the strong monotone 'adversary' can replace the entire subtree, starting from the child, with a different subtree (topology and labels) of its choice.

- Output the resulting tree T.

An upper bound against this latter model amounts to a recovery guarantee without any assumptions as to what happens after a 'mutation' of labels — for example, a genetic mutation might affect reproductive fitness and change the birth rule for the topology. Other variants of monotone adversaries could also be justified.

Majority Fails   It is helpful to first see how adversaries in these models might break existing algorithms. Recall that in the broadcast tree model, taking the majority vote of the leaves yields an algorithm that works up to the Kesten–Stigum bound, and this is optimal since reconstruction is impossible beyond this. In the language of k and ε, each node gives birth to $\text{Pois}(k(1-\varepsilon))$ nodes

6

of the same label and $\text{Pois}(k\varepsilon)$ nodes of the opposite label. Hence in a depth R tree we expect k^R leaves, but the total bias of the spins can be recursively computed as $k^R(1-2\varepsilon)^R$. The fact that majority works can be proven by applying the second moment method and comparing the bias to its variance.

However, an overwhelming number of the leaves are on some path that has a flip in the label at some point: we only expect $k^R(1-\varepsilon)^R$ nodes with all-root-spin ancestry, a vanishing proportion as $R \to \infty$. The strong monotone adversary has control over all the rest, and can easily break the majority vote. Even the monotone cutting adversary can control the majority vote, by cutting leaves whose spin matches the root but whose parents have the opposite label. This happens for a constant fraction of the leaf nodes, and this change overwhelms the majority vote. So majority vote fails against the semirandom model, for *all* nontrivial k and ε.

This is an instructive example, but we emphasize that breaking one algorithm does not yield a lower bound. For example, if the algorithm knew what the adversary were doing, it could potentially infer information about where the adversary has cut edges based on the degree profile, and could use this to guess the label of the root.

The Problem of Orientation Many first attempts at a separation in the broadcast tree model (which work!) rely on knowing the label of the root. However, such adversaries present a major difficulty in coupling them to a corresponding adversary in the graph. Each graph gives rise to many overlapping broadcast trees (the local neighborhood of each vertex) and a graph adversary needs to simultaneously make all of these tree reconstruction problems harder. This means a graph adversary cannot focus on trying to hide the spin of a specific tree root; rather, it should act in a local, orientation-free way that inhibits the propagation of information in all directions.

A promising approach is to look for nodes v whose neighbors in G_{pre} all have the opposite label, and cut all of these edges. Such nodes serve only to further inter-connect each community, and cutting their edges would seem to make community detection strictly harder. In the corresponding broadcast tree model, these nodes v represent flips in the label that are entirely corrected back, and deleting them penalizes any sort of over-reliance on distributional flukes in how errors propagate. For example, majority reconstruction in the tree fully relies on predictable tail events whereby nodes with label different from the root may lead to subtrees voting in the correct direction nonetheless.

The Problem of Dependence Now, however, a different sort of problem arises: if we were to naively apply the adversary described above to a broadcast tree, this would introduce complicated distance-2 dependencies in the distribution of observed spins, as certain diameter-2 spin patterns are banned in the observed tree (as they would have been cut). In particular, the resulting distribution is no longer a Markov random field. This is not inherently a problem, in that we could still hope to prove stronger lower bounds for such models beyond the Kesten–Stigum bound. However, the difficulty is that even for quite simple models on a tree (e.g. the Potts model [Sly11], asymmetric binary channels [BCMR06]) the threshold is not known, and the lower bound techniques that establish the Kesten–Stigum bound seem to break down.

An alternative is to specifically look for degree-2 nodes v whose neighbors in G_{pre} have the opposite label, and cut both incident edges. Although there are still issues about making this a Markov random field, we can alleviate them by adding a 3-clique potential on each degree-2 node and its two neighbors. Then after we marginalize out the label of the degree-2 node, the 3-clique potential becomes a 2-clique potential, and we return to a Markov random field over a tree! In other words, if we ignore the spin on a degree-2 node and treat its two incident edges like a single edge, we return to a well-behaved spin propagation rule.

2.3 Our Adversary

We are now ready to describe the adversary that we will use to prove Theorems 1.2 and 1.4. We only need two additional adjustments beyond the discussion above. Instead of making every possible degree-2 cutting move as described earlier, we will only cut with probability δ. We will tune δ to ensure that the monotone changes we make do not overshoot and accidentally reveal more information about the underlying communities by cutting in too predictable a fashion. Finally, our adversary adds local restrictions to where and how it cuts, to ensure that the changes it makes do not overlap or interfere with each other (e.g. chains of degree-2 nodes). These details will not be especially relevant until Section 5, where they simplify the combinatorics of guessing the original precursor graph from the observed graph.

Distribution 2.3. Let (a, b, n) be given. Write $k = \frac{a+b}{2}$ and $\varepsilon = \frac{b}{a+b}$. We sample a 'precursor' graph $G_{\mathrm{pre}} \sim G(n, a/n, b/n)$, and apply the following adversary:

Adversary:

- If at least 3 neighbors of a vertex v have degree not equal to 2, mark v good.

- If a degree-2 vertex has both of its neighbors marked good, mark it tagged.

- For each tagged node v whose two neighbors w_1, w_2 both have opposite label to v: with probability δ, delete both edges (v, w_1) and (v, w_2) (otherwise keep both edges) where

$$\delta \triangleq \begin{cases} 1 & \text{if } \varepsilon \leq \frac{1}{3}, \\ \frac{(1-2\varepsilon)^2}{\varepsilon^2} & \text{if } \varepsilon \geq \frac{1}{3}. \end{cases} \tag{1}$$

We can now outline the proof of our main theorem. In order to show that partial recovery is impossible, it suffices to show that it is impossible to reconstruct the relative spin (same or different) of two random nodes u and v, better than random guessing. Before applying the adversary, an $O(\log n)$-radius neighborhood U around u resembles the broadcast tree model rooted at u. After the adversary is applied to the graph, U resembles a broadcast tree model with a corresponding cutting adversary applied. This resemblance will be made precise by the coupling argument of Section 3; there will be some complications in this, and our tree will not be uniformly the same depth but will have a serrated boundary.

We show in Section 4 that when the cutting adversary is applied to the tree, the tree reconstruction problem (guess the spin of u from the spins on the boundary of U) becomes strictly harder: the average branching factor becomes lower due to sufficient cutting, while the new spin propagation rule resembles classical noisy propagation with at least as much noise (so long as we marginalize out the spins of tagged nodes). We then apply the proof technique of Evans et al. [EKPS00] to complete the tree lower bound.

The final step in the proof is to show that reconstructing the relative spin of u and v is at least as hard as reconstructing the spin of u given the spins on the boundary of U (which separates u and v with high probability). This is one of the most technically involved steps in the proof. In the lower bound for the random model [MNS14b], this step — called "no long-range correlations" — was already an involved calculation on a closed-form expression for $\mathbb{P}[G \mid \sigma]$. In our setting, in order to get a closed-form expression for the conditional probability, we will sum over all the possible precursor graphs G_{pre} that could have yielded the observed graph G. We characterize these

precursors in Lemma 5.7, and the main reason for the `good` and `tagged` nodes in our adversary construction is to simplify this process.

A natural open question is whether one can find the optimal monotone adversary. For instance, we have not even used the power to add edges within communities. Note however, that our current adversary is delicately constructed in order to make each step of the proof tractable. It is not too hard to propose alternative adversaries that seem stronger, but it is likely that one of the major steps in the proof (tree lower bound or no long-range correlations) will become prohibitively complicated. Recent predictions on the performance of SDP methods [JMR15] could be suggestive of the true semirandom threshold.

3   Coupling of Graph and Tree Models

It is well-known that sparse, random graphs are locally tree-like with very few short cycles. This is the basis of Mossel–Neeman–Sly's approach in coupling between a local neighborhood of the stochastic block model and the broadcast tree model [MNS14b]. Hence, our first order of business will be to couple a typical neighborhood from our graph distribution (Distribution 2.3) to the following tree model, against which we can hope to show lower bounds.

Distribution 3.1. Given the parameters (a, b, R), generate a tree T with spins σ as follows:

- Start with a root vertex ρ, with spin σ_ρ chosen uniformly at random from ± 1.

- Until a uniform depth $R + 3$ (where the root is considered depth 0), let each vertex v give birth to $\mathrm{Pois}(a/2)$ children of the same spin and $\mathrm{Pois}(b/2)$ children of opposite spin.

- Apply the graph adversary (from Distribution 2.3) to this tree; this involves assigning markings `good` and `tagged`, and cutting some `tagged` nodes. Keep only the connected component of the root.

- Remove all nodes of depth greater than R (the bottom 3 levels).

- Remove any `tagged` node at depth R along with its siblings, exposing the parent as a leaf.

The reason we trim 3 levels at the bottom is to ensure that the markings and cuttings in T match those in G, since these depend on the radius-3 neighborhood of each node and edge, respectively. Removing `tagged` nodes at level R will ensure that the more complicated spin interactions of `tagged` nodes and their neighbors do not span across the boundary of a tree neighborhood in the graph — we want to cleanly separate out a tree recovery problem. We use a slightly non-conventional definition of 'leaves':

Definition 3.2. When we refer to the **leaves** of a tree sampled from Distribution 3.1 we mean the nodes at depth R plus any nodes at depth $R - 1$ that are revealed during the last step. Nodes at depth $\leq R - 1$ that happen to give birth to no children are not considered leaves; if the root is `tagged` and gets cut by the adversary so that the tree is a single node, this single node is not considered a leaf.

We can couple the above tree model to neighborhoods of the graph:

Proposition 3.3. *Let* (a, b, n) *be given, and let* ρ_G *be any vertex of the graph. Let*

$$R = \left\lfloor \frac{\log n}{10 \log(2(a + b))} \right\rfloor - 3. \tag{2}$$

9

There exists a joint distribution $(G, \sigma_G, T, \sigma_T)$ *such that the marginals on* (G, σ_G) *and* (T, σ_T) *match the graph and tree models, respectively, while with probability* $1 - o(1)$:

- *there exists an isomorphism (preserving edges, spins, and the markings* good *and* tagged*) between the tree* T *and a vertex-subset* U *of* G,

- *the tree root corresponds to the vertex* ρ_G *in the graph, and the leaves correspond to a vertex set* B *that separates the interior* $U \smallsetminus B$ *from the rest of the graph* $G \smallsetminus U$.

- *we have* $|U| = O(n^{1/8})$.

Proof. As proven in [MNS14b], there exists a coupling between the precursor graph $(G_{\mathrm{pre}}, \sigma_G)$ and the precursor tree $(T_{\mathrm{pre}}, \sigma_T)$, such that T_{pre} matches the radius $R + 3$ neighborhood U of ρ_G in G_{pre}, which has size $O(n^{1/8})$; this fails with probability $o(1)$.

Next the adversary assigns vertex markings (good and tagged) to both models. The marking of each vertex is deterministic and only depends on the topology of the radius-3 neighborhood of the vertex. Thus the markings in G_{pre} and T_{pre} match up to radius R. Some tagged nodes are *cuttable*, i.e. both their neighbors have opposite spin; the cuttable nodes in G_{pre} and T_{pre} also match up to radius R. The adversary now cuts the edges incident to a random subset of cuttable vertices; we can trivially couple the random choices made on G_{pre} with those on T_{pre} up to radius R. We keep only the connected component of the root in T; likewise let us keep only the corresponding vertices in U, i.e. only those still connected to ρ_G by a path in U.

After removing nodes of depth greater than R, we have removed the subset of T for which the markings and the action of the adversary differ from those in G. Thus at this stage, the tree exactly matches the radius R neighborhood of ρ_G in G, along the isomorphism given by the coupling from [MNS14b]. Any boundary vertex of this neighborhood must have distance exactly R from ρ_G, thus its corresponding vertex in the tree has depth R and is a leaf. Passing to the final step of removing tagged leaves in T and their siblings, we remove their corresponding nodes from U; this does not change the boundary-to-leaves correspondence. $\qquad\square$

4   Random vs. Semirandom Separation in the Tree Model

In this section we show that our tree distribution (Distribution 3.1) evidences a random vs. semi-random separation in the broadcast tree model. Recall that the goal is to recover the spin of the root from the spins on the leaves, given full knowledge of the tree topology and the node markings (good and tagged). Recall the non-conventional definition of 'leaves' (Definition 3.2).

Let $\Delta = \Delta(a, b, R)$ be the advantage over random guessing, defined such that $\frac{1+\Delta}{2}$ is the probability that the optimal estimator (maximum likelihood) succeeds at the above task. We will show that our tree model is asymptotically infeasible ($\Delta \to 0$ as $R \to \infty$) for a strictly larger range of parameters (a, b) than that of the corresponding random model.

Proposition 4.1. *For every real number* $k > 1$, *there exists* $0 < \varepsilon < \frac{1}{2}$ *such that* $k(1 - 2\varepsilon)^2 > 1$, *yet for a tree sampled from Distribution 3.1 with parameters* $a = 2k(1 - \varepsilon)$, $b = 2k\varepsilon$ *and depth* R, *we have* $\Delta(a, b, R) \to 0$ *as* $R \to \infty$.

Recall that the condition $k(1 - 2\varepsilon)^2 > 1$ is the classical Kesten–Stigum bound, which is sufficient to beat random guessing in the random model [KS66]. Several decades later, this bound was found to be tight for the random model [EKPS00]. There remain many open questions regarding the hardness of tree models, and some care was required in crafting an adversary for this problem that keeps the proof of this lower bound tractable.

10

Broadly, the Kesten–Stigum bound asserts that recoverability depends on the average branching factor (a contribution from the tree topology) and the amount of noise (a contribution from the spin propagation rule). The first step of our proof is to distinguish these in our distribution: we can first generate a tree topology from the appropriate marginal distribution, and then sample spins from the conditional distribution given this tree. We will show how to view this distribution on spins within the lower bound framework of [EKPS00]. Moreover, the new spin propagation rule is at least as noisy as the original, while our cutting adversary has strictly decreased the average branching factor.

Proof of Proposition 4.1. We first re-state our tree model in terms of topology generation followed by spin propagation. Instead of letting each node give birth to $\text{Pois}(a/2)$ same-spin children and $\text{Pois}(b/2)$ opposite-spin children, we can equivalently let each node v give birth to $\text{Pois}(k)$ children and then independently choose the spin of each child to be the same as v with probability $1-\varepsilon$ and opposite to v otherwise. (The equivalence of these steps is often known as 'Poissonization'.) Here the correspondence between (a,b) and (k,ε) is as usual: $k = \frac{a+b}{2}$ and $\varepsilon = \frac{b}{a+b}$. This allows us to first sample the entire tree topology without spins. Then we can add markings (`good` and `tagged`), since these depend only on the topology. Next, we sample the spins as above, by generating an appropriate independent ± 1 value f_e on each edge, indicating whether or not a sign flip occurs across that edge. Finally, we cut edges according to the adversary's rule.

Distribution 4.2. Given the parameters (a, b, R), generate a tree T with spins σ as follows:

- Start with a root vertex ρ. Until a uniform depth $R+3$, let each vertex v give birth to $\text{Pois}(k)$ nodes.

- Mark nodes as `good` and `tagged` according to the rules of the graph adversary.

- For each edge e, generate an independent flip value f_e which is $+1$ with probability $1-\varepsilon$ and -1 otherwise.

- Choose the root spin σ_ρ uniformly from ± 1. Propagate spins down the tree, letting $\sigma_v = \sigma_u f_e$ where e is the edge connecting v to its parent u.

- Cut edges according to the adversary's rule, keeping only the connected component of the root.

- Trim the bottom of the tree (according to the last two steps of Distribution 3.1).

It is clear that this tree distribution (Distribution 4.2) is identical to the original tree distribution (Distribution 3.1). Our next step will be to re-state this model in yet another equivalent way. The goal now is to sample the final tree topology (including which edges get cut) before sampling the spins. Consider a `tagged` node v, its parent u, and its single child w. Instead of writing the spin propagation as independent flips $f_{(u,v)}$ and $f_{(v,w)}$, we will write it as random variables c_v and f_v. Here c_v is equal to 1 if the adversary decides to cut v (and 0 otherwise), and f_v is equal to 1 if $\sigma_u = \sigma_w$ (and -1 otherwise). This means if $f_{(u,v)} = f_{(v,w)} = 1$ then c_v is 1 with probability δ (and 0 otherwise); and if we do not have $f_{(u,v)} = f_{(v,w)} = 1$ then $c_v = 0$. Hence $c_v = 1$ with probability $\varepsilon^2 \delta$. If $c_v = 1$ then f_v is irrelevant because the adversary will cut v from the tree. Conditioned on $c_v = 0$, f_v takes the value $+1$ with probability $\frac{(1-\varepsilon)^2 + \varepsilon^2(1-\delta)}{1-\varepsilon^2\delta}$ and -1 otherwise. This means that for a `tagged` (but not cut) node v, the joint distribution of (σ_u, σ_w) obeys a propagation rule that

is equivalent to putting noise ε' (instead of ε) on edges (u, v) and (v, w), where (using the definition of δ)

$$\varepsilon' = \begin{cases} \frac{1}{2} - \frac{1}{2}\sqrt{\frac{1-3\varepsilon}{1+\varepsilon}} & \text{if } \varepsilon \leq \frac{1}{3}, \\ \frac{1}{2} & \text{if } \varepsilon \geq \frac{1}{3}. \end{cases}$$

One can verify that $\varepsilon < \varepsilon' \leq \frac{1}{2}$ for all $\varepsilon \in (0, \frac{1}{2})$. For most `tagged` nodes in the tree, we are simply going to replace ε by ε' on the incident edges, which gives the correct joint distribution of (σ_u, σ_w) but not the correct joint distribution of $(\sigma_u, \sigma_v, \sigma_w)$. This is acceptable because the distribution of leaf spins (given the root spin) is still correct; for this reason we have made sure that none of the leaves are `tagged` nodes. The only time when we actually care about the spin of a `tagged` node is in the case where the root ρ is `tagged`. In this case, the root might be cut by the adversary, yielding a 1-node tree (with no revealed leaves). Otherwise, if the root is `tagged` but not cut, our spin propagation rule needs to sample the spins of the root's two children (a `tagged` node that is not cut must have degree 2) from the appropriate joint distribution over $\{\pm 1\}^2$; let $\mathcal{D}^+$ denote this distribution, conditioned on $\sigma_\rho = +1$. It will not be important to compute $\mathcal{D}^+$ explicitly (although it is straightforward to do so). We are now ready to state the next tree model. This model is equivalent to the previous ones in that the joint distribution of the root spin, topology, markings, and leaf spins is the same. The spins on the `tagged` nodes (other than the root) do not have the same distribution as before, but this is irrelevant to the question of recovering the root from the leaves.

Distribution 4.3. Given the parameters (a, b, R), generate a tree T with spins σ as follows:

- Start with a root vertex ρ. Until a uniform depth $R+3$, let each vertex v give birth to $\text{Pois}(k)$ nodes.

- Mark nodes as `good` and `tagged` according to the rules of the graph adversary.

- Decide which nodes the adversary should cut: cut each `tagged` node independently with probability $\varepsilon^2 \delta$.

- Let $\varepsilon_e = \varepsilon'$ for edges e that are incident to a `tagged` node, and let $\varepsilon_e = \varepsilon$ for all other edges.

- For each edge e, generate an independent flip value f_e which is $+1$ with probability $1 - \varepsilon_e$ and -1 otherwise.

- Choose the root spin σ_ρ uniformly from ± 1. If the root is `tagged` but not isolated: let u_1, u_2 be its children, draw $(d_1, d_2) \sim \mathcal{D}^+$, and let $\sigma_{u_1} = d_1 \sigma_\rho$, $\sigma_{u_2} = d_2 \sigma_\rho$. For all other nodes, propagate spins down the tree as usual, letting $\sigma_v = \sigma_u f_e$ where e is the edge connecting v to its parent u.

- Trim the bottom of the tree (according to the last two steps of Distribution 3.1).

Our next step is to further modify this tree model in ways that only make it easier, in the sense that the advantage $\Delta(a, b, R)$ can only increase. First, we will address the issue of the complicated propagation rule $\mathcal{D}^+$ in the case that the root is `tagged`. Suppose the root is `tagged` (but not cut) and consider deterministically setting $\sigma_{u_1} = \sigma_{u_2} = \sigma_\rho$ where u_1, u_2 are the root's two children. From there, spin propagation continues as usual. We claim that this new model can only be easier than the original one. To see this, note that the new model can 'simulate' the original one: upon observing leaf spins drawn from the new model, drawn $(d_1, d_2) \sim \mathcal{D}^+$ and then, for each $i \in \{1, 2\}$ and for each leaf w descended from u_i, replace σ_w by $d_i \sigma_w$. For convenience we will also replace

the first level of the tree by deterministic zero-noise propagation in the case where the root is not `tagged`. We can similarly argue that the model only gets easier, since one can simulate the old model using the new model by sampling the noise on the first level.

Note that we now have a tree model such that once the topology is chosen, the spin-propagation rule is very simple: at each edge e, a sign flip occurs independently with some probability $\varepsilon_e \in \{\varepsilon, \varepsilon', 0\}$. Hardness results for such a tree model were studied by Evans et al., who established the following bound on the advantage Δ_T for a fixed tree topology T ([EKPS00] Theorem 1.3'):

$$\Delta_T^2 \leq 2 \sum_{\text{leaves } v} \Theta_v^2 \quad \text{where} \quad \Theta_v \triangleq \prod_{e \in \text{path}(v)} \theta_e,$$

and where $\theta_e \triangleq 1 - 2\varepsilon_e$, and path$(v)$ denotes the unique path from the root to leaf v. In our case, the tree T (including both the tree topology and markings) is random, so the advantage is $\Delta = \mathbb{E}_T[\Delta_T]$, by the law of total probability. By Jensen's inequality,

$$\Delta^2 = (\mathbb{E}_T[\Delta_T])^2 \leq \mathbb{E}_T[\Delta_T^2] \leq \mathbb{E}_T\left[2 \sum_{\text{leaves } v} \Theta_v^2\right] \triangleq W.$$

Our goal is to show $\Delta \to 0$ (as $R \to \infty$), so it is sufficient to show $W \to 0$. Any leaf in our tree is at height $\geq R-1$. Furthermore we have $\varepsilon_e = 0$ for edges incident to the root and $\varepsilon \leq \varepsilon_e \leq \frac{1}{2}$ elsewhere. Therefore, for any leaf v, we can bound $\Theta_v \leq (1 - 2\varepsilon)^{R-2}$. Note that we have not actually used the fact that ε' strictly exceeds ε; we will in fact be able to prove our result by leveraging only the decrease in branching factor and not the noise increase. Now we have

$$W = \mathbb{E}_T\left[2 \sum_{\text{leaves } v} \Theta_v^2\right] \leq 2\,\mathbb{E}_T[\# \text{ leaves}](1 - 2\varepsilon)^{2(R-2)}$$

and so we need only to bound the expected number of leaves. For $i = 0, \dots, R$, let *level i* denote the set of vertices at distance i from the root (so the root is on level 0). Call levels $1, 7, \dots, 6j + 1, \dots$ the *base* levels and call levels $4, 10, \dots, 6j - 2, \dots$ the *cutting* levels. Imagine growing the tree using Pois(k) births as usual, and marking nodes as `good` and `tagged` as usual. Now allow the adversary to cut vertices as usual, except refuse to cut any `tagged` node that is not on a cutting level. Note that this can only result in more leaves, and so this new process will give an upper bound on the expected number of leaves in our actual model.

To analyze the new process, suppose we have a node v on a base level. The number of descendants n_v of v on the next base level is a random variable whose distribution does not depend on the ancestors of v; this independence is the reason for defining base levels and cutting levels. Define k' by $(k')^6 \triangleq \mathbb{E}[n_v]$; then $(k')^6 < k^6$, since there is some nonzero probability that one of v's descendants will be cut at the subsequent cutting level. The expected number of leaves in the entire tree is now at most $k^6(k')^{R-6}$, where the factor k^6 accounts for the first level and the last ≤ 5 levels that are not followed by a base level. Now $W \leq 2k^6(k')^{R-6}(1 - 2\varepsilon)^{2(R-2)}$, which goes to 0 as $R \to \infty$ provided that

$$k'(1 - 2\varepsilon)^2 < 1.$$

Hence this inequality suffices for impossibility of recovery.

However, k' depends on the topology of the final tree model, which depends on k, ε, and δ, so this inequality is slightly more complicated than it looks. We write $k'(k, \varepsilon)$ to make this dependence explicit; recall that δ is a (continuous) function of ε. We argued above that $k'(k, \varepsilon) < k$ for all $0 \leq \varepsilon \leq \frac{1}{2}$. In particular, at the critical value $\varepsilon_{\text{crit}} = \frac{1}{2}(1 - \frac{1}{\sqrt{k}})$ for the *random* model, we have

$$k'(k, \varepsilon_{\text{crit}})(1 - 2\varepsilon_{\text{crit}})^2 < k(1 - 2\varepsilon_{\text{crit}})^2 = 1.$$

But $k'(k, \varepsilon)(1 - 2\varepsilon)^2$ is a continuous function of ε, so if we take ε to be slightly less than $\varepsilon_{\mathrm{crit}}$, we must still have $k'(k, \varepsilon)(1 - 2\varepsilon)^2 < 1$, while $k(1 - 2\varepsilon)^2 > 1$, as desired. $\qquad \square$

This completes the proof of Theorem 1.4 (semirandom vs. random separation on trees). In Appendix A, we explicitly compute a lower bound on the separation $k^6 - k'(k, \varepsilon)^6$.

5 No Long-Range Correlations

The lower bound of the previous section will form the core of a lower bound for the graph: we have already established that, for a large range of parameters, it is impossible to learn the spin of a node u purely from the spins at the boundary B of its tree-like local neighborhood. We will now see why this makes recovery in the graph impossible: there is almost nothing else to learn about σ_u from beyond its local neighborhood once we know the spins on B.

Lemma 5.1 (No long-range correlations). *Let a graph G (including markings* good *and* tagged*) and spins σ be drawn from Distribution 2.3. Let $A = A(G)$, $B = B(G)$, $C = C(G)$ be a vertex-partition of G such that*

- *B separates A from C in G (no edges between A and C),*

- *$|A \cup B| = O(n^{1/8})$,*

- *B contains no* tagged *nodes.*

Then $\mathbb{P}[\sigma_A \mid G, \sigma_{BC}] = (1 + o(1))\,\mathbb{P}[\sigma_A \mid G, \sigma_B]$, for asymptotically almost every (a.a.e.) G and σ.

Here σ_A denotes the spins on A and σ_{BC} denotes the spins on $B \cup C$.

To clarify, when we refer to good or tagged nodes in G we are referring to the original markings that they were assigned in G_{pre}. For instance, if a tagged node is cut by the adversary, it is still considered tagged in G even though it no longer has degree 2. Recall that the markings (good and tagged) in G are revealed to the reconstruction algorithm.

In Lemma 5.1, we do crucially use that B does not contain tagged nodes: if a node in B were tagged with spin $+1$, and we then revealed that its neighbor in C has spin -1, this would strengthen our belief that the neighbor in A has spin $+1$, as otherwise the tagged node would have some substantial probability of having been cut, which is observed not to be the case. So Lemma 5.1 would be false if we allowed tagged nodes in B.

5.1 Structure of Possible Precursors G_{pre}

The proof of Lemma 5.1 will require a thorough understanding of the distribution of spins given G, which we can only obtain by understanding the possible precursors of G under the adversary. The reason for the good and tagged nodes in our adversary construction is to make these precursors well-behaved. We start with some simple observations. Let G_{pre} be a graph that yields G (with nonzero probability) by application of the adversary.

Observation 5.2. *A* good *node has degree at least 3 in G.*

Proof. A good node has at least three non-degree-2 neighbors in G_{pre} and the adversary will not remove these. $\qquad \square$

Observation 5.3. *The adversary does not create any new degree-2 nodes. In other words, if a node has degree 2 in G then it has degree 2 in G_{pre} (with the same two neighbors).*

Proof. In order to create a degree-2 node v, the adversary must cut at least one edge incident to v. This can only happen if v is either `tagged` or `good`. It cannot be `tagged` because it does not have degree 2 in G_{pre}. But it also cannot be `good` or else it has degree at least 3 in G (by Observation 5.2). $\qquad\square$

The key property of the `good`/`tagged` construction is that the adversary cannot change whether or not a node has the following 'goodness' property.

Definition 5.4. Say that a node has the **goodness** property with respect to a graph if at least three of its neighbors do not have degree 2.

Lemma 5.5. *The `good` nodes are precisely the nodes that have the goodness property with respect to G.*

Note that this is not tautological because the `good` nodes are defined as the nodes that have the goodness property with respect to G_{pre}, not G.

Proof. We need to show that the goodness property is invariant under the adversary's action. First we assume v has goodness in G_{pre} and show that it also has goodness in G. Since v has the goodness property in G_{pre}, it has three neighbors u_1, u_2, u_3 in G_{pre} that do not have degree 2. Each u_i remains connected to v in G since the adversary only cuts edges that are incident to a degree-2 node. Furthermore, each u_i does not have degree 2 in G because degree-2 nodes cannot be created (Observation 5.3).

Now we show the converse: assume v does not have goodness in G_{pre} and show that it still does not have goodness in G. The only way that v could obtain goodness is if at least one of its degree-2 neighbors u becomes non-degree-2 (while remaining connected to v). But this is impossible because whenever the adversary cuts an edge incident to a degree-2 vertex u, it causes u to become isolated in G. $\qquad\square$

Next we state an easy fact about the structure of `tagged` nodes in G.

Lemma 5.6. *The `tagged` nodes in G are precisely the degree-2 nodes with two `good` neighbors, plus some isolated nodes.*

Proof. Every `tagged` node in G_{pre} has degree-2 with two `good` neighbors. If the node is cut, it becomes isolated in G; otherwise it remains degree-2 with two `good` neighbors. Conversely, let v be degree-2 in G with two `good` neighbors; we will show v is `tagged`. Since degree-2 nodes cannot be created (Observation 5.3), v has degree-2 in G_{pre} with the same two `good` neighbors, and is therefore `tagged`. $\qquad\square$

Now we are ready to characterize the possible precursors G_{pre} of a given G.

Lemma 5.7. *Suppose we have a graph G (including node markings) and spins σ, drawn from Distribution 2.3. The probability that a graph G_{pre} yields G under the action of the adversary is zero unless G_{pre} can be obtained from G by connecting each isolated `tagged` node v of G to exactly two `good` nodes of opposite spin to v. In this case, if G has $m = m(G)$ isolated `tagged` nodes and $w = w(\sigma, G)$ `tagged` nodes with two opposite-spin neighbors, then*

$$\mathbb{P}[G \mid G_{pre}, \sigma] = (1 - \delta)^w \delta^m$$

where we define $0^0 = 1$ in the case $\delta = 1, w = 0$.

Recall that δ is the probability with which the adversary cuts each `tagged` node that has two opposite-spin neighbors.

Proof. First suppose G_{pre} can yield G via the adversary. We will show that G_{pre} takes the desired form. The nodes cut by the adversary are precisely the isolated `tagged` nodes of G. Every such node was originally connected to two opposite-spin `good` nodes in G_{pre}. Therefore G_{pre} can be obtained from G by connecting each isolated `tagged` node to exactly two opposite-spin `good` nodes. Conversely, let G_{pre} be obtained from G by connecting each isolated `tagged` node to exactly two opposite-spin `good` nodes. The `good` nodes in G are (by Lemma 5.5) precisely the nodes that have the goodness property in G. These are also precisely the nodes that have the goodness property in G_{pre}, since the process of connecting each isolated `tagged` node to two `good` nodes does not change goodness. Consider running the adversary on G_{pre}. The nodes that it marks `good` will be precisely the `good` nodes in G. Also, the nodes that it marks `tagged` will be precisely the `tagged` nodes in G; this is clear from Lemma 5.6. This means the adversary will output G iff it chooses to cut the m `tagged` nodes that are isolated in G and chooses not to cut the w `tagged` nodes that have two opposite-spin neighbors in G. This happens with probability $(1-\delta)^w \delta^m$. $\square$

5.2 Proof of No Long-Range Correlations

Proof of Lemma 5.1. We proceed as follows. In the ordinary stochastic block model, given an observed graph, the probability of any set of spins σ factorizes as a product of pairwise interactions. In our model, by summing over all possible precursors G_{pre} that could have lead to G via the adversary, we find the same pairwise interactions together with a further global, combinatorial interaction. We show that the neighborhood A is too small to make a significant impact on this global interaction, while the pairwise interactions between A and C are weak, so that the only factors relevant to A are the pairwise interactions within $A \cup B$, which are independent of the spins in C.

Let G_{pre} denote the graph before the action of the adversary. Then $\mathbb{P}[G_{\mathrm{pre}} \mid \sigma]$ factors into the following potentials on unordered pairs: $\mathbb{P}[G_{\mathrm{pre}} \mid \sigma] = \Phi(\sigma, G_{\mathrm{pre}}) \triangleq \prod_{u,v} \phi(\sigma_u, \sigma_v, G_{\mathrm{pre}})$ with u, v ranging over unordered pairs of distinct vertices, where

$$\phi(\sigma_u, \sigma_v, H) = \begin{cases} a/n & \text{if } u \sim v \text{ and } \sigma_u = \sigma_v, \\ b/n & \text{if } u \sim v \text{ and } \sigma_u \neq \sigma_v, \\ 1 - a/n & \text{if } u \not\sim v \text{ and } \sigma_u = \sigma_v, \\ 1 - b/n & \text{if } u \not\sim v \text{ and } \sigma_u \neq \sigma_v \end{cases}$$

where $\sim$ denotes adjacency in the graph H.

To leverage this description, let us sum over all possible precursors G_{pre} of G under the adversary. Let $L(\sigma, G)$ denote the set of possible precursors G_{pre} of G, as described by Lemma 5.7: G_{pre} is obtained from G by connecting each of the m isolated `tagged` nodes to exactly two `good` nodes of the opposite spin.

$$\mathbb{P}[\sigma \mid G] = \sum_{G_{\mathrm{pre}} \in L(\sigma, G)} \mathbb{P}[G_{\mathrm{pre}} \mid \sigma]\, \mathbb{P}[\sigma]\, \mathbb{P}[G \mid \sigma, G_{\mathrm{pre}}] / \mathbb{P}[G]$$

$$\propto \sum_{G_{\mathrm{pre}} \in L(\sigma, G)} \Phi(\sigma, G_{\mathrm{pre}})\, \mathbb{P}[G \mid \sigma, G_{\mathrm{pre}}]$$

$$\propto \sum_{G_{\mathrm{pre}} \in L(\sigma, G)} \Phi(\sigma, G_{\mathrm{pre}})(1 - \delta)^w \delta^m$$

16

The proportionality constant hidden by '$\propto$' depends on G but not on σ. As G_{pre} is obtained from G by replacing $2m$ opposite-spin non-edges by opposite-spin edges, we have for every $G_{\mathrm{pre}} \in L(\sigma, G)$,

$$\Phi(\sigma, G_{\mathrm{pre}}) = \Phi(\sigma, G) \left(\frac{b/n}{1 - b/n} \right)^{2m}.$$

Thus none of the terms depend on the precise choice of G_{pre}:

$$\mathbb{P}[\sigma \mid G] \propto |L(\sigma, G)| \, \Phi(\sigma, G) \left(\frac{b/n}{1 - b/n} \right)^{2m} (1 - \delta)^w \delta^m$$

$$\propto |L(\sigma, G)| \, \Phi(\sigma, G)(1 - \delta)^w.$$

where we have dropped the constants that only depend on G (and not σ).

Now we compute $|L(\sigma, G)|$. Suppose there are $m/2 + \alpha$ isolated `tagged` nodes of positive spin and $m/2 - \alpha$ of negative spin. Suppose there are $g/2 + \beta$ `good` nodes of positive spin, and $g/2 - \beta$ of negative spin. Then the number of possible G_{pre} is

$$|L(\sigma, G)| = \binom{g/2 + \beta}{2}^{m/2 - \alpha} \binom{g/2 - \beta}{2}^{m/2 + \alpha}.$$

We can establish that this global L factor only barely depends on the spins of A:

Lemma 5.8. *For a.a.e. G, σ_B, σ_C, it holds for all σ_A, σ'_A that*

$$\frac{|L(\sigma_A, \sigma_B, \sigma_C, G)|}{|L(\sigma'_A, \sigma_B, \sigma_C, G)|} = 1 + o(1).$$

The proof proceeds via concentration of measure and Taylor expansion, and is deferred to Appendix B.

Paraphrasing this lemma, a.a.s. over (σ_B, G), there exists a 'good' subset of a.a.e. σ_C such that $|L(\sigma, G)|$ is independent of σ_A up to a $1 + o(1)$ factor. Let us also require of 'good' σ_C that the census (sum of spins) is $O(\sqrt{n} \log n)$; as C consists of all but $O(n^{1/8})$ nodes of G, it is equivalent to ask for the same concentration over all spins of G, and this occurs a.a.s. by Hoeffding. Let Ω be this set of 'good' σ_C values. Since $\sigma_C \in \Omega$ a.a.s. we have for any set $D = D(G)$ (which will be taken as either A or $A \cup B$ below),

$$\mathbb{P}[\sigma_D, G] = (1 + o(1)) \, \mathbb{P}[\sigma_D, \sigma_C \in \Omega, G] \text{ a.a.s. over } \sigma, G. \tag{3}$$

We include a rigorous proof of this statement in Appendix C.

It will be useful to factor the product Φ of classical pairwise interactions into subsets of these interactions as

$$\Phi(\sigma, G) = Q_{AB,AB}(\sigma_{AB}, G) Q_{BC,C}(\sigma_{BC}, G) Q_{A,C}(\sigma_{AC}, G).$$

Here, for instance, $Q_{BC,C}$ denotes the product of $\phi(\sigma_u, \sigma_v, G)$ over unordered pairs u, v consisting of one vertex from $B \cup C$ and one from C. The corresponding "no long-range correlations" proof in [MNS14b] established that for 'good' values $\sigma_C \in \Omega$, $Q_{A,C}(\sigma_{AC}, G) = (1 + o(1))K(G)$ for a quantity $K(G)$ depending only on $|A|$ and $|C|$. Their proof of this fact holds verbatim in our setting: it only requires that $|A| = o(n)$ and that the number of $+1$ spins in the graph is distributed as $\mathrm{Binom}(n, \frac{1}{2})$.

Similarly, we can factor the $(1 - \delta)^w$ term as

$$(1 - \delta)^{w(\sigma, G)} = (1 - \delta)^{w_A(\sigma_{AB}, G)}(1 - \delta)^{w_C(\sigma_{BC}, G)}$$

17

where, for instance, w_A counts the number of `tagged` nodes in A with two opposite-spin neighbors. This factorization holds as there are no A–C edges and no `tagged` nodes in B. We can now absorb these terms into the Q terms: define

$$Q'_{AB,AB}(\sigma_{AB}, G) = (1 - \delta)^{w_A(\sigma_{AB},G)} Q_{AB,AB}(\sigma_{AB}, G),$$

and similarly $Q'_{BC,C}$, and $\Phi' = Q'_{AB,AB} Q'_{BC,C} Q_{A,C}$, so that for $\sigma_C \in \Omega$,

$$\mathbb{P}[\sigma \mid G] \propto \Phi'(\sigma, G) |L(\sigma, G)|$$
$$\propto (1 + o(1)) \Phi'(\sigma, G) K(G) Q'_{AB,AB}(\sigma_{AB}, G) Q'_{BC,C}(\sigma_{BC}, G).$$

At this point we roughly adapt the proof of conditional independence from factorization in a Markov random field, following the corresponding proof in [MNS14b]. We compute that for a.a.e. (σ, G),

$$\mathbb{P}[\sigma_A \mid \sigma_B, G] = \frac{\mathbb{P}[\sigma_{AB}, G]}{\mathbb{P}[\sigma_B, G]}$$
$$= (1 + o(1)) \frac{\mathbb{P}[\sigma_{AB}, \sigma_C \in \Omega \mid G]}{\mathbb{P}[\sigma_B, \sigma_C \in \Omega \mid G]} \qquad \text{using (3)},$$
$$= (1 + o(1)) \frac{\sum_{\sigma'_C \in \Omega} \Phi'(\sigma_{AB}, \sigma'_C, G) |L(\sigma_{AB}, \sigma'_C, G)|}{\sum_{\sigma'_A, \sigma'_C \in \Omega} \Phi'(\sigma'_{AC}, \sigma_B, G) |L(\sigma'_{AC}, \sigma_B, G)|}$$
$$= (1 + o(1)) \frac{Q'_{AB,AB}(\sigma_{AB}, G) \sum_{\sigma'_C \in \Omega} Q'_{BC,C}(\sigma_B, \sigma'_C, G) |L(\sigma_{AB}, \sigma'_C, G)|}{\sum_{\sigma'_A} Q'_{AB,AB}(\sigma'_A, \sigma_B, G) \sum_{\sigma'_C \in \Omega} Q'_{BC,C}(\sigma_B, \sigma'_C, G) |L(\sigma'_{AC}, \sigma_B, G)|},$$
$$= (1 + o(1)) \frac{Q'_{AB,AB}(\sigma_{AB}, G)}{\sum_{\sigma'_A} Q'_{AB,AB}(\sigma'_A, \sigma_B, G)} \qquad \text{using Lemma 5.8.}$$

Multiplying the top and bottom by $K(G) Q'_{BC,C}(\sigma_{BC}, G) |L(\sigma, G)|$:

$$= (1 + o(1)) \frac{K(G) Q'_{AB,AB}(\sigma_{AB}, G) Q'_{BC,C}(\sigma_{BC}, G) |L(\sigma, G)|}{\sum_{\sigma'_A} K(G) Q'_{AB,AB}(\sigma'_A, \sigma_B, G) Q'_{BC,C}(\sigma_{BC}, G) |L(\sigma, G)|}$$
$$= (1 + o(1)) \frac{\Phi'(\sigma, G) |L(\sigma, G)|}{\sum_{\sigma'_A} \Phi'(\sigma'_A, \sigma_{BC}, G) |L(\sigma'_A, \sigma_{BC}, G)|} \qquad \text{using Lemma 5.8 again,}$$
$$= (1 + o(1)) \mathbb{P}[\sigma_A \mid \sigma_{BC}, G],$$

as desired. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad$ □

6 Random vs. Semirandom Separation in the Block Model

We can now assemble all of the pieces to prove our main result. We first prove that it is impossible to estimate the relative spin of any fixed pair of nodes, in a strictly larger parameter range than for the random model. The impossibility of partial recovery will then easily follow.

Proposition 6.1. *For all $k > 1$, there exists $0 < \varepsilon < \frac{1}{2}$ such that $k(1 - 2\varepsilon)^2 > 1$, yet given a graph G (including markings* `good`/`tagged`*) from Distribution 2.3 with parameters $a = 2k(1 - \varepsilon)$, $b = 2k\varepsilon$, we have that for any fixed vertices u, v,*

$$\mathbb{P}[\sigma_u = +1 \mid \sigma_v = +1, G] \to \frac{1}{2}$$

for a.a.e. G as $n \to \infty$.

By 'fixed' vertices u, v we mean that the vertices are fixed before σ and G are chosen; so by symmetry it doesn't matter which u, v pair we fix.

Proof. Given k, choose ε as in Proposition 4.1 (tree separation). With probability $1 - o(1)$, the tree coupling of Proposition 3.3 centered at vertex u succeeds. In this case, the neighborhood U of u coupled to the tree is of size $O(n^{1/8})$, and so with probability $1 - o(1)$, v lies outside this neighborhood. Let B be the boundary vertices of U, i.e. those vertices in U corresponding to the leaves of the tree. Let A be the interior $U \smallsetminus B$, and let C be the complement of U in G.

By the law of total variance,

$$\mathrm{Var}(\sigma_u \mid \sigma_v, G) \geq \mathbb{E}_{\sigma_{BC} \mid \sigma_v, G}[\mathrm{Var}(\sigma_u \mid \sigma_v, \sigma_{BC}, G)]. \tag{4}$$

With further probability $1 - o(1)$, Lemma 5.1 (no long-range correlations) succeeds, and we have

$$\mathbb{P}[\sigma_A \mid \sigma_{BC}, G] = (1 + o(1)) \, \mathbb{P}[\sigma_A \mid \sigma_B, G],$$

so that

$$\mathrm{Var}(\sigma_u \mid \sigma_{BC}, G) = (1 + o(1)) \, \mathrm{Var}(\sigma_u \mid \sigma_B, G).$$

But it follows from the coupling in Proposition 3.3 (which includes spins and markings) that

$$\mathrm{Var}(\sigma_u \mid \sigma_B, G) = (1 + o(1)) \, \mathrm{Var}(\sigma_\rho \mid \sigma_{\mathrm{leaves}}, T).$$

Since $R \to \infty$ as $n \to \infty$, non-reconstruction in the tree model (Proposition 4.1) implies that this latter variance converges to 1: the variance of a $\{\pm 1\}$-valued random variable with expectation $o(1)$ is $1 - o(1)$.

So we know that, with probability $1 - o(1)$, a $1 - o(1)$ proportion of σ_{BC} contribute a value $1 - o(1)$ to the expectation in (4). The remaining $o(1)$ proportion of σ_{BC} contribute a value bounded in magnitude by 1, as the variance of a $\{\pm 1\}$-valued variable must be. It follows that, a.a.s.,

$$\mathrm{Var}(\sigma_u \mid \sigma_v, G) \geq \mathbb{E}_{\sigma_{BC}}[\mathrm{Var}(\sigma_u \mid \sigma_{BC}, G)] = 1 - o(1),$$

and the only way that this is possible in the $\{\pm 1\}$-valued distribution $\sigma_u \mid \sigma_v, G$ is if

$$\mathbb{P}[\sigma_u = +1 \mid \sigma_v = +1, G] = \frac{1}{2} + o(1),$$

as desired. □

It will follow immediately from Proposition 6.1 that it is hard to find a partition that is correlated (better than random guessing) with the true one. For if it was possible to find such a partition then one could also guess the relative spins of u and v. We now complete the proof of our main theorem, restating it slightly more precisely.

Theorem 6.2 (restatement of Theorem 1.2). *For any $k > 1$, there exists $0 < \varepsilon < \frac{1}{2}$ so that $k(1 - 2\varepsilon)^2 > 1$ and hence partial recovery in the stochastic block model $G(n, a/n, b/n)$ is possible, and yet against the monotone adversary given in Section 2, for any $\eta > \frac{1}{2}$, no estimator of the spins achieves η-partial recovery with probability greater than $o(1)$ as $n \to \infty$.*

Proof. Let $\widehat{\sigma}$ be some assignment of spins to vertices that achieves η-partial recovery: $\widehat{\sigma}$ agrees with the true spins on at least ηn vertices, possibly after a global spin flip. Consider the relative spins $\widehat{\sigma}_u \widehat{\sigma}_v$, where u and v are distinct vertices; these match the true relative spins on at least

$$\eta n(\eta n - 1) + (1 - \eta)n((1 - \eta)n - 1) = (1 + o(1))(\eta^2 + (1 - \eta)^2)n^2$$

19

ordered pairs of distinct vertices. If we choose two distinct vertices at random, the chance of correctly estimating their relative spin from $\widehat{\sigma}$ is at least

$$(1 + o(1))(\eta^2 + (1 - \eta)^2). \tag{5}$$

Suppose for a contradiction that some estimator achieves η-partial recovery, for some $\eta > \frac{1}{2}$, with probability p not converging to 0 as $n \to \infty$. When η-partial recovery succeeds, the process above recovers the relative spin of two random vertices with probability at least $(1 + o(1))(\eta^2 + (1 - \eta)^2)$, and note $\eta^2 + (1 - \eta)^2 > \frac{1}{2}$. When partial recovery does not succeed, the process still recovers the relative spin of two random vertices with probability at least $(1 + o(1))\frac{1}{2}$, as can be seen by plugging in $\eta = \frac{1}{2}$ to (5). It follows that we can recover the relative spin of two random vertices with probability at least $(1 + o(1))[p(\eta^2 + (1 - \eta)^2) + (1 - p)\frac{1}{2}]$ which remains bounded above $\frac{1}{2}$ as $n \to \infty$, contradicting Proposition 6.1. (Proposition 6.1 assumes the markings on G are known, but clearly the problem is only harder when they are unknown, since an estimator can choose to ignore them.) $\qquad\square$

7 Robustness of SDPs for Partial Recovery

We now turn to giving algorithms for partial recovery in the semirandom setting. In the existing literature on *exact* recovery, it has been observed that algorithms obtained through semidefinite programming extend almost automatically to semirandom models [FK01, AL14, PW15], and moreover many of these results match the information-theoretic thresholds for exact recovery. In contrast, semidefinite programs for partial recovery, such as those of Guédon and Vershynin [GV15], come within a constant of the threshold but have been unable to close this gap.

In fact Guédon and Vershynin [GV15] developed a general framework for using SDPs to solve problems on sparse graphs, including partial recovery for the stochastic block model. We define a notion of semirandom model for any problem in this framework, generalizing the semirandom model for community detection. We show that any analysis that follows their framework carries over automatically to this semirandom model. In particular, after minor modifications, the SDP analyzed in [GV15] achieves partial recovery in the semirandom block model, up to within a constant factor of the classical threshold.

Semirandom vs. random gaps offer an explanation for why it seems hard to find semidefinite programs for partial recovery that reach the information-theoretic threshold: the analysis often extends equally well to the semirandom model, where we know the threshold is strictly harder!

7.1 The Guédon–Vershynin Framework and Partial Recovery

In the framework of [GV15], there are several key terms. The first is a matrix B which is usually computed in a simple manner from the adjacency matrix of the observed graph. In our case, we compute $B = A - \lambda J$, where J is the all-ones matrix, $\lambda = \frac{a+b}{2n}$ is a regularization constant, and A is the observed adjacency matrix (albeit with the non-standard convention that there are ones along the diagonal instead of zeros). The goal is to show that B — which is a random matrix — is near some fixed reference matrix R. In our case this will be $\frac{a-b}{2n}\sigma\sigma^\top$, which is nearly equal to $\mathbb{E}[B]$ except on the diagonal. Then, one solves an SDP to maximize $\langle B, Z \rangle$ over $Z \in \Omega$, where Ω is the space of symmetric PSD $n \times n$ matrices Z satisfying $\mathrm{diag}(Z) \leq I$ (i.e. the diagonal entries of Z are at most 1).[3] The goal of the analysis is to show that the SDP outputs some $\widehat{Z}$ that is close to

---

[3]The framework also allows Ω to have additional constraints, but we will not need this.

a "ground truth" Z_R, which in our case is the $\{\pm 1\}$-valued matrix $\sigma\sigma^\top$ of true relative spins. The following result outlines the steps of the analysis.

Proposition 7.1. *Suppose we have (B, R, Z_R) such that the following three conditions hold for some value α, some function $F(\beta)$, and some matrix norm $\|\cdot\|$:*

(1) Z_R is a maximizer of the reference objective $\langle R, -\rangle$ over Ω
(the reference SDP recovers the truth),

(2) $\|B - R\|_{\infty\rightarrow 1} \leq \alpha$
(the observed objective is close to the reference one in cut norm),

(3) if $Z \in \Omega$ and $\langle R, Z_R - Z\rangle \leq \beta$, then $\|Z_R - Z\|^2 \leq F(\beta)$
(good solutions to the reference SDP are close to the ground truth).

Then $\|Z_R - \widehat{Z}\|^2 \leq F(2K_G\alpha)$ where $\widehat{Z}$ is any maximizer of the empirical objective $\langle B, -\rangle$ over Ω, and K_G is the Grothendieck constant.

Here, the cut norm (or ∞-to-1 norm) of a matrix is defined as

$$\|M\|_{\infty\rightarrow 1} \triangleq \max_{\|x\|_\infty = 1} \|Mx\|_1 = \max_{x \in \{\pm 1\}^n} \|Mx\|_1.$$

The proof of Proposition 7.1 follows from Lemma 3.3 in [GV15], and uses Grothendieck's inequality.

The partial recovery results of [GV15] proceed by verifying the three conditions of Proposition 7.1 for a particular choice of parameters:

Proposition 7.2. *Assume $a > 20$. Let $B = A - \lambda J$ with $\lambda = \frac{a+b}{2n}$, $R = \frac{a-b}{2n}\sigma\sigma^\top$, and $Z_R = \sigma\sigma^\top$. Conditions (1–3) of Proposition 7.1 hold a.a.s. with $\alpha = O(n\sqrt{a+b})$, $F(\beta) = \beta \cdot O(n/(a-b))$, and $\|\cdot\|$ as the Frobenius norm $\|\cdot\|_2$.*

Concretely, this means that we solve the following SDP:

SDP 7.3. Maximize $\langle A - \lambda J, Z\rangle$ subject to $Z \succeq 0$ and $\mathrm{diag}(Z) \leq I$, where $\lambda = \frac{a+b}{2n}$.

Note that the regularization constant λ is necessary because if we simply take $B = A$ then condition 1 of Proposition 7.1 fails. In [GV15] they estimate λ from the empirical average degree, but their arguments also apply to the case where we deterministically take $\lambda = \frac{a+b}{2n}$. (This requires knowledge of the parameters a, b but we will address this issue later.)

In [GV15], condition 1 is shown in Lemma 5.1, and conditions 2 and 3 are implicit in the proof of Lemma 5.2. These require a technical condition: $\max\{a(1 - a/n), b(1 - b/n)\} \geq 20$, which for large n simply amounts to $a > 20$. By Propositions 7.1 and 7.2, we now attain the result $\|Z_R - \widehat{Z}\|_2^2 \leq O(n^2\sqrt{a+b}/(a-b))$ a.a.s. It is also shown in [GV15] how to translate this to a precise partial recovery result:

Proposition 7.4. *Let $Z_R = \sigma\sigma^\top$. For any $\frac{1}{2} < \eta \leq 1$, η-partial recovery succeeds a.a.s. in the stochastic block model, provided that $\|Z_R - \widehat{Z}\|_2^2 \leq (1-\eta)n^2$, by taking the signs in the top eigenvector of $\widehat{Z}$.*

Corollary 7.5. *There exists a constant C such that η-partial recovery succeeds a.a.s. in the stochastic block model as $n \rightarrow \infty$ whenever $a > 20$ and $(a - b)^2 \geq C(1 - \eta)^{-2}(a + b)$.*

The constant C is quite large: [GV15] sets $C = 10^4$, although no attempt is made to optimize this constant. Nevertheless this result is only off by a constant from the threshold, which is $(a - b)^2 > 2(a + b)$. Our random vs. semirandom separation becomes small as k grows, so it remains plausible that semidefinite programs can achieve partial recovery when $k(1 - 2\varepsilon)^2 > 1 + o(1)$ as $k \to \infty$. In fact, SDPs are known to distinguish random block model graphs from Erdős–Rényi graphs in such a range [MS15], and it would be interesting to determine whether this carries through to partial recovery in the semirandom model.

7.2 SDPs and Semirandom Models

We now turn to a semirandom view of the general Guédon–Vershynin framework. In the semirandom block model, a monotone adversary is allowed to make changes aligned with the ground truth Z_R; more formally, it can add a matrix S to the observed adjacency A, where S is symmetric and has 1's in some same-spin entries where A is 0, and -1's in some opposite-spin entries where A is 1. It is easily verified that Z_R maximizes $\langle S, - \rangle$ over Ω: every matrix in Ω has entries in $[-1, 1]$, and Z_R has ± 1 entries that match the sign pattern of S.

Following this observation, we propose a precise definition of semirandom models for Guédon–Vershynin problems:

Definition 7.6. A matrix S is a **monotone change** if Z_R is a maximizer for $\langle S, - \rangle$ over Ω. In the **semirandom model**, a preliminary objective B is generated from the random model, and then an adversary modifies the objective by any monotone change S, yielding an observed matrix $B + S$.

Note that the set of such monotone changes S forms a cone, which matches the intuition that semirandom models can make unbounded changes, but only in certain directions aligned with the ground truth.

Any analysis following the framework of Proposition 7.1 generalizes in a formal sense to the semirandom model:

Proposition 7.7. *Suppose that (B, R, Z_R) satisfy conditions (1–3) of Proposition 7.1 for some $\alpha, F(\beta), \|\cdot\|$. Then it remains the case that $\|Z_R - \widehat{Z}_S\|^2 \leq F(2K_G \alpha)$, where $\widehat{Z}_S$ is any maximizer of $\langle B + S, - \rangle$ over Ω, and S is any monotone change (in the sense above), which is allowed to depend adversarially on B.*

Proof. It suffices to establish conditions (1–3) of Proposition 7.1 with $R + S$ playing the role of R and $B + S$ playing the role of B:

(1) As Z_R maximizes both $\langle R, - \rangle$ and $\langle S, - \rangle$ over Ω, it certainly maximizes their sum $\langle R + S, - \rangle$.

(2) We have $\|(B + S) - (R + S)\|_{\infty \to 1} = \|B - R\|_{\infty \to 1}$, which by hypothesis is at most α with high probability.

(3) Suppose that $Z \in \Omega$ and $\langle R + S, Z_R - Z \rangle \leq \beta$. Then

$$\langle R, Z_R - Z \rangle \leq \beta - \langle S, Z_R - Z \rangle \leq \beta,$$

since Z_R is a maximizer of $\langle S, - \rangle$ over Ω. It follows from the original condition (3) that $\|Z_R - Z\|^2 \leq F(\beta)$ in the appropriate norm.

$\square$

We have shown that any *analysis* of an SDP following the Guédon–Vershynin framework transfers to the semirandom model. A recovery guarantee for the same SDP by different means would not automatically yield such a guarantee in the semirandom model. This differs from exact recovery problems, where semirandom guarantees can take the success of the SDP as a black box, and certify that any successful instance remains successful under monotone changes [FK01].

Corollary 7.8. *Let A be sampled from the semirandom block model with parameters a, b, n. Suppose that we know these parameters, so that we can run SDP 7.3 with the deterministic choice $\lambda = \frac{a+b}{2n}$. If $a > 20$ and $(a - b)^2 \geq 10^4(1 - \eta)^{-2}(a + b)$ then this SDP achieves η-partial recovery a.a.s. (after taking the sign of the top eigenvector of the SDP output $\widehat{Z}$).*

This dependence on parameter knowledge is slightly unwieldy; on the other hand, estimating the correct regularization parameter λ from $A + S$ is not immediately easy, since model-specific estimators are precisely the sort of techniques that semirandom models aim to penalize. Instead, we show that deterministically taking $\log n/n$ in place of λ avoids this dependence. Indeed, the same SDP appears in the exact recovery literature [ABH15] with deterministic regularization parameter $\frac{1}{2}$, which is comparatively very large, but comes at the cost of requiring precisely balanced communities. Our approach here is a middle ground that allows for natural $\sqrt{n}$-scale deviations in the balance of the two communities. There is nothing special about the value $\log n/n$ and it can in fact be taken to be anything of order strictly between $1/n$ and $1/\sqrt{n}$.

Proposition 7.9. *The Guédon–Vershynin SDP (SDP 7.3) with regularization parameter $\lambda' = \log n/n$ achieves η-partial recovery against the semi-random model a.a.s. so long as $a > 20$ and $(a - b)^2 \geq 10^4(1 - \eta)^{-2}(a + b)$.*

Proof. In light of Proposition 7.7, it is sufficient to show that this SDP works in the random model. We check the three conditions for (B', R', Z_R) where $B' = A - \lambda'J = B + (\lambda - \lambda')J$, $R' = \frac{a-b}{2n}\sigma\sigma^\top + (\lambda - \lambda')J = R + (\lambda - \lambda')J$, and $Z_R = \sigma\sigma^\top$ where $\lambda = \frac{a+b}{2n}$ and $\lambda' = \frac{\log n}{n}$. We will use the fact that the three conditions are satisfied for (B, R, Z_R), and we will achieve nearly the same parameters $\alpha, F(\beta)$ as in that case.

(1) We certify that Z_R maximizes $\langle R', - \rangle$ by SDP duality. The dual SDP reads

$$\text{minimize} \sum_v \gamma_v \quad \text{subject to } \Lambda \triangleq \text{diag}(\gamma) - R' \succeq 0,$$

where v runs over all vertices in the graph. To certify that Z_R is optimal, it suffices by complementary slackness to find γ such that $\Lambda \succeq 0$ and $\langle \Lambda, Z_R \rangle = 0$. Since Λ and Z_R are PSD, the second condition is equivalent to $\Lambda Z_R = 0$. As the columns of Z_R are spanned by σ, this second condition reads $\Lambda\sigma = 0$; expanding,

$$\gamma_v = \frac{a - b}{2} + \left(\frac{a + b}{n} - \frac{\log n}{n}\right)\left(|C_v| - |\overline{C}_v|\right) \quad \forall v$$

where C_v is the set of nodes with the same spin as node v, and $\overline{C}_v$ is the complement of C_v. By Hoeffding we have $\left||C_v| - |\overline{C}_v|\right| \leq \sqrt{n}\log n$ a.a.s. and so

$$\gamma_v = \frac{a - b}{2} - O(\log^2 n/\sqrt{n}) = \frac{a - b}{2} - o(1).$$

In particular, $\gamma_v \geq 0$ for all v a.a.s. This choice of γ guarantees $\Lambda\sigma = 0$ so it remains to show that $\Lambda \succeq 0$. Since $\Lambda\sigma = 0$, it suffices to show that $v^\top \Lambda v \geq 0$ for all $v \perp \sigma$. But note that

$$\Lambda = \left(\frac{\log n}{n} - \frac{a + b}{2n}\right)J - \frac{a - b}{2n}\sigma\sigma^\top + \text{diag}(\gamma),$$

23

$$v^\top \Lambda v = v^\top \left[\left(\frac{\log n}{n} - \frac{a+b}{2n} \right) J + \mathrm{diag}(\gamma) \right] v \geq 0,$$

since $\gamma \geq 0$ a.a.s. so that both J and $\mathrm{diag}(\gamma)$ are PSD. This establishes optimality of Z_R.

(2) We have $B' - R' = B - R$ and so this step follows from the corresponding step for (B, R, Z_R).

(3) Suppose that $Z \in \Omega$ satisfies $\langle R', Z_R - Z \rangle \leq \beta$. Letting $\nu = \lambda' - \lambda$ we have $0 \leq \nu \leq \frac{\log n}{n}$ and $R' = R - \nu J$. Notice that $\langle Z, J \rangle \geq 0$ since both matrices are PSD, while $\langle Z_R, J \rangle = \frac{1}{2}(|C_+| - |\overline{C}_-|)^2 \leq n \log^2 n$ a.a.s. where C_+, C_- are the community sizes. Thus,

$$\begin{aligned}
\langle R, Z_R - Z \rangle &= \langle R', Z_R - Z \rangle + \nu \langle J, Z_R - Z \rangle \\
&\leq \beta + \nu \cdot n \log^2 n - \nu \langle J, Z \rangle \\
&\leq \beta + \log^3 n.
\end{aligned}$$

Applying condition 3 for (B, R, Z_R), we obtain $\|Z_R - Z\|_2^2 \leq (\beta + \log^3 n) \cdot O(n/(a-b))$.

By Proposition 7.4, we obtain partial recovery by rounding the leading eigenvector, under the same conditions as in Corollary 7.5, and with the same constant C. $\qquad\square$

This analysis completes the proof of Theorem 1.3 from the introduction.

Independently, [MMV15b] proved a result on SDP robustness that matches ours. Additionally, they consider robustness to a different adversary that can only add or remove $o(n)$ edges but is not required to be monotone. Our methods also extend to this case (and match their results) because $o(n)$ changes can only change the cut norm by $o(n)$.

8   Robustness of Recursive Majority in the Broadcast Tree Model

Here we establish that recursive majority achieves reconstruction in the semirandom broadcast tree model, even with respect to the strong adversary that has total control over entire subtrees. As one would imagine, in recursive majority the root spin is estimated as the majority vote of its children's estimated spins, which in turn are estimated as the majority vote of their children, and so on down to the known leaf spins. In the random model, this algorithm falls short of the Kesten–Stigum bound by a factor of $\sqrt{2/\pi}$ as $k \to \infty$ [Mos98]. Nevertheless, it remains the algorithm of choice in practice for tree reconstruction problems, often by the name of 'parsimony' [Mos04]. It seems that the advantages of recursive majority over majority only become clear when studying these algorithms through semirandom models!

To keep things simple, we will first consider trees in which each non-leaf node has exactly k children, matching the setting of [Mos98] and much of the tree reconstruction literature. At the end, we show how to adapt our proof in a straightforward manner to the case where the number of children of each non-leaf node is a Poisson random variable.

Proposition 8.1. *Let $M_k(p)$ be the probability of a majority 'yes' vote of k voters each voting 'yes' with probability p:*

$$M_k(p) \triangleq \mathbb{P}\left[\mathrm{Binom}(k, p) > \frac{k}{2} \right] + \frac{1}{2} \mathbb{P}\left[\mathrm{Binom}(k, p) = \frac{k}{2} \right].$$

Let (T, σ) be drawn from the strong semirandom model on the regular tree with braching factor k, height R, and noise ε. Let ε_k^* be defined by

$$\frac{1}{1 - \varepsilon_k^*} = \max_{q \in (0,1]} \frac{M_k(q)}{q},$$

24

and let q_k^* be the maximizer. So long as $\varepsilon \le \varepsilon_k^*$, the recursive majority algorithm correctly recovers the root spin from the topology and leaf spins, with probability at least $p_k^* = q_k^*/(1 - \varepsilon_k^*) > \frac{1}{2}$. Conversely, if $\varepsilon > \varepsilon_k^*$, there exists an adversary forcing recursive majority to fail with probability $1 - o(1)$ as $R \to \infty$.

Proof. Let (T, σ) be drawn from the strong semirandom model, starting with a k-regular tree of height R. Let p_R denote the success probability on such a tree. As a base case, we have $p_0 = 1$. For each child of the root, the spin matches that of the root with probability $(1 - \varepsilon)$, in which case the child will be correctly labeled by recursive majority with probability p_{R-1}, by induction. The other children are under adversary control. Then the number of recovered child labels agreeing with the root label stochastically dominates $\mathrm{Binom}(k, p_{R-1}(1 - \varepsilon))$, for any choice of adversary. The recovered root label is the majority vote of the recovered child labels, so we obtain that

$$p_R \ge M_k(p_{R-1}(1 - \varepsilon)).$$

Moreover this inequality is exactly tight for a certain adversary, which replaces each replaceable subtree by a path down to a single leaf of spin opposite to the root.

We are interested in the limiting recovery probability $p_\infty = \lim_{R \to \infty} p_R$; this is the greatest fixed point in $[0, 1]$ of the function $p \mapsto M_k(p(1 - \varepsilon))$. Equivalently, $q_\infty = p_\infty(1 - \varepsilon)$ is the greatest solution in $[0, 1-\varepsilon]$ of the fixed-point equation $q/(1-\varepsilon) = M_k(q)$. Geometrically, this is the greatest intersection point of the graph of M_k with the line of slope $1/(1 - \varepsilon)$ through the origin.

Before proceeding, it is worth noting the geometry of the graph of M_k. From the definition, M_k is monotone on $[0, 1]$, with $M_k(0) = 0$, $M_k(\frac{1}{2}) = \frac{1}{2}$, and $M_k(1) = 1$. By expanding the probability mass function of the binomial distribution, M_k is a degree k polynomial, and thus smooth. Less obviously, M_k is strictly convex on $[0, \frac{1}{2}]$ and strictly concave on $[\frac{1}{2}, 1]$ [Mos98].

The line of slope 2 ($\varepsilon = \frac{1}{2}$) does not intersect the graph of M_k nontrivially, since $M_k(q) \le \frac{1}{2}$ on $[0, \frac{1}{2}]$ and $M_k(q) \le 1$ on $[\frac{1}{2}, 1]$. The line of slope 1 ($\varepsilon = 0$) certainly intersects the graph of M_k at $\frac{1}{2}$ and 1. Hence there exists some maximal $0 < \varepsilon_k^* < \frac{1}{2}$, at which the line of slope $1/(1 - \varepsilon_k^*)$ intersects the graph of M_k tangentially at some q_k^*. Equivalently, we can characterize this slope as the maximum slope of the line defined by the origin and any point on the graph of M_k:

$$\frac{1}{1 - \varepsilon_k^*} = \max_{q \in (0, 1]} \frac{M_k(q)}{q}.$$
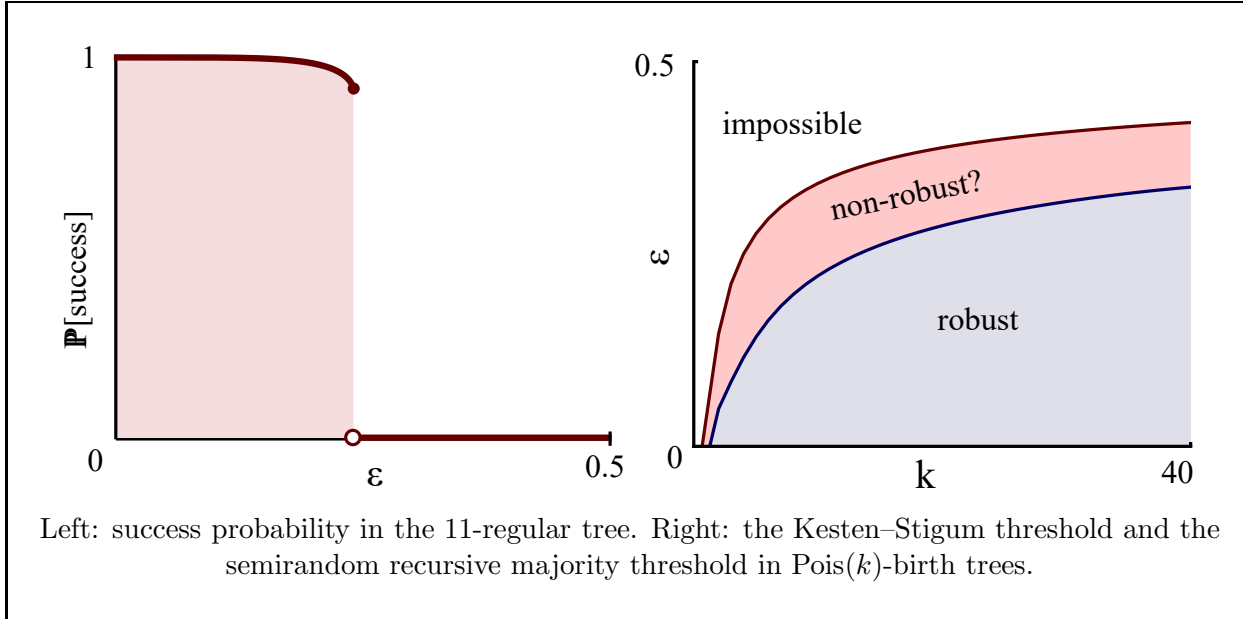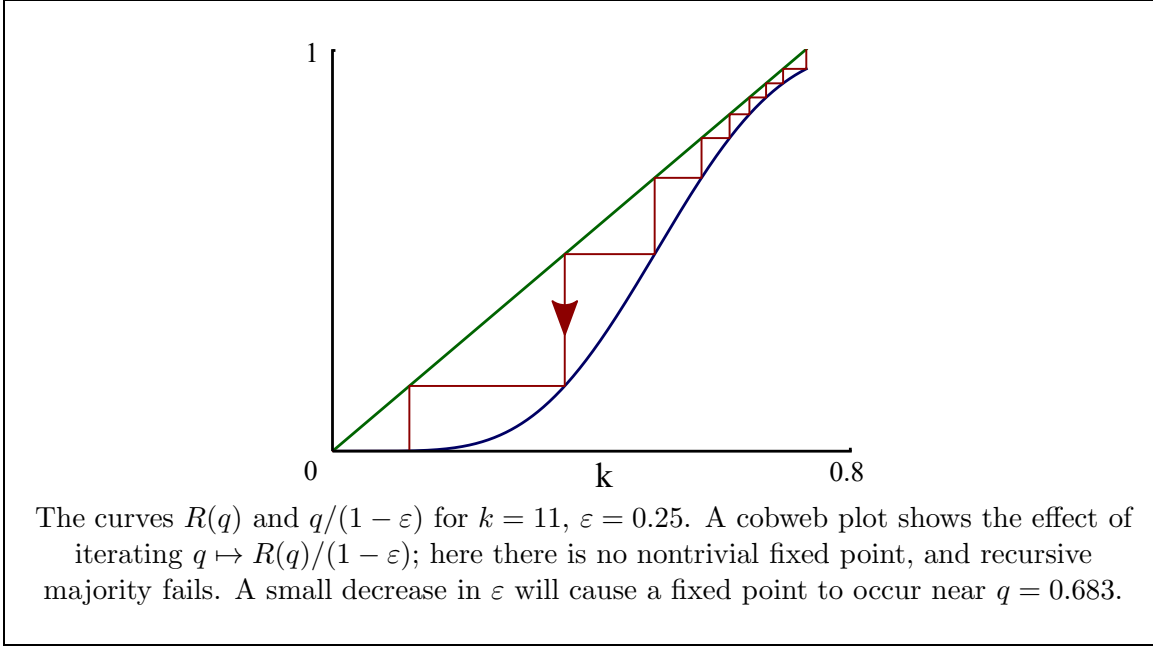
By the concavity and convexity properties of M_k, the graph of M_k lies below the line of slope 1 on $(0, \frac{1}{2})$, and above on $(\frac{1}{2}, 1)$. It follows that $q_k^* > \frac{1}{2}$.

As we sweep q from q_k^* to 1, the slope $R(q)/q$ passes continuously from $1/(1 - \varepsilon_k^*)$ to 1, granting an intersection $q \ge q_k^*$ satisfying $R(q)/q = 1/(1 - \varepsilon)$ for every $\varepsilon \in [0, \varepsilon_k^*]$, by the intermediate value theorem. Thus, whenever $0 \le \varepsilon \le \varepsilon_k^*$, recursive majority achieves recovery with probability

$$p_\infty = \frac{q_\infty}{1 - \varepsilon} \ge q_\infty \ge q_k^* > \frac{1}{2}.$$

Whenever $\varepsilon > \varepsilon_k^*$, the two curves do not intersect on $[0, 1]$ except at 0, so the limiting success probability p_∞ is 0. $\qquad\square$

It is interesting that, as the noise ε varies, we see the success probability jump discontinuously from $p_k^* > \frac{1}{2}$ to zero. This contrasts with the behavior of recursive majority in the ordinary broadcast tree model, in which the error probability transitions continuously to $\frac{1}{2}$ at a threshold [Mos98].

The curves $R(q)$ and $q/(1-\varepsilon)$ for $k = 11$, $\varepsilon = 0.25$. A cobweb plot shows the effect of iterating $q \mapsto R(q)/(1-\varepsilon)$; here there is no nontrivial fixed point, and recursive majority fails. A small decrease in ε will cause a fixed point to occur near $q = 0.683$.

Left: success probability in the 11-regular tree. Right: the Kesten–Stigum threshold and the semirandom recursive majority threshold in $\mathrm{Pois}(k)$-birth trees.

For any fixed k, computing the critical value ε_k^* amounts to maximizing the polynomial $\frac{M_k(q)}{q}$, or equivalently finding the unique root of its derivative in $(0, 1]$. For example it is easily computed that $\varepsilon_3^* = \frac{1}{9}$. However, we would like some asymptotic understanding of the values ε_k^* as $k \to \infty$.

Proposition 8.2. *The following asymptotic expression holds for ε_k^* as $k \to \infty$:*

$$\varepsilon_k^* = \frac{1}{2} - \left(\frac{1}{2} + o(1)\right) \sqrt{\frac{\log k}{k}}.$$

Proof. By the maximality property defining ε_k^* and q_k^*, we must have

$$0 = \frac{\mathrm{d}}{\mathrm{d}q} \frac{M_k(q)}{q}\Big|_{q=q_k^*} = (q_k^*)^{-2}(q_k^* \cdot M_k'(q_k^*) - M_k(q_k^*)),$$

from which it follows that $M_k'(q_k^*) = \frac{M_k(q_k^*)}{q_k^*} = \frac{1}{1-\varepsilon_k^*}$. Thus, as $\varepsilon_k^* \in [0, \frac{1}{2}]$, we know that

$$1 = \frac{1}{1-0} \le M_k'(q_k^*) = \frac{M_k(q_k^*)}{q_k^*} \le \frac{1}{1-\frac{1}{2}} = 2.$$

In analyzing recursive majority in the random model, Mossel [Mos98] computed this derivative using Russo's formula. Let us assume for the moment that k is odd; the even case is similar. Then

$$M_k'(q) = (q(1-q))^{(k-1)/2} k \binom{k-1}{(k-1)/2}.$$

If we evaluate this at $q = \frac{1}{2} + \frac{1}{2}\sqrt{\alpha \log k / k}$, we obtain:

$$
\begin{aligned}
M_k'(q) &= \left(\frac{1}{4} - \frac{1}{4}\frac{\alpha \log k}{k}\right)^{(k-1)/2} k \binom{k-1}{(k-1)/2} \\
&= (1 + o(1)) \left(1 - \frac{\alpha \log k}{k}\right)^{(k-1)/2} \sqrt{\frac{k}{\pi}} \\
&= (1 + o(1)) \left(1 - \frac{\frac{1}{2}\alpha \log k}{(k-1)/2}\right)^{(k-1)/2} \sqrt{\frac{k}{\pi}} \\
&= (1 + o(1)) \exp(-\frac{1}{2}\alpha \log k)\sqrt{\frac{k}{\pi}} \\
&= (1 + o(1)) \sqrt{\frac{1}{\pi}} k^{\frac{1}{2} - \frac{1}{2}\alpha}.
\end{aligned}
$$

In the second-to-last step, we have used the asymptotic identity $(1 - x/k)^k = (1 + o(1)) \exp(-x)$ as $k \to \infty$, which still holds when x depends on k so long as $x = o(\sqrt{k})$. Now for $\alpha > 1$, the derivative $M_k'(q)$ tends to 0 as $k \to \infty$, while for $\alpha < 1$, this tends to ∞ as $k \to \infty$. Thus, writing $q_k^* = \frac{1}{2} + \frac{1}{2}\sqrt{\alpha_k^* \log k / k}$, we must have $\alpha_k^* = 1 + o(1)$. We immediately obtain a lower bound on ε_k^*:

$$\varepsilon_k^* = 1 - \frac{q_k^*}{M_k(q_k^*)} \ge \frac{q_k^*}{1} = \frac{1}{2} + \left(\frac{1}{2} + o(1)\right) \sqrt{\frac{\log k}{k}}.$$

For a matching upper bound, apply Hoeffding's inequality for a binomial tail bound, to find

$$M_k(q) \ge 1 - \exp(-\alpha \log k / 2) = 1 - k^{-\alpha/2},$$

and use the maximality of q_k^* to find

$$
\begin{aligned}
\varepsilon_k^* = 1 - \frac{q_k^*}{M_k(q_k^*)} &\le 1 - \frac{q}{M_k(q)} \\
&\le 1 - \frac{\frac{1}{2} + \frac{1}{2}\sqrt{\alpha \log k / k}}{1 - k^{-\alpha/2}} \\
&= 1 - \left(\frac{1}{2} + \frac{1}{2}\sqrt{\frac{\alpha \log k}{k}}\right) \sum_{j=0}^{\infty} (k^{-\alpha/2})^j \\
&= \frac{1}{2} - \frac{1}{2}\sqrt{\frac{\alpha \log k}{k}} - O(k^{-\alpha/2}),
\end{aligned}
$$

so that $\frac{1}{2} - (\frac{1}{2} + \beta)\sqrt{\frac{\log k}{k}}$ is an asymptotic upper bound for every $\beta > 0$, which completes the proof. $\square$

Note that $p_k^* = q_k^*/(1 - \varepsilon)$ is the probability of success at the threshold ε_k^*, so this proof also provides some sense of the critical success probability:

$$p_k^* = \frac{\frac{1}{2} + (\frac{1}{2} + o(1))\sqrt{\log k/k}}{1 - \left(\frac{1}{2} - (\frac{1}{2} + o(1))\sqrt{\log k/k}\right)} = 1 - o\left(\sqrt{\frac{\log k}{k}}\right).$$

So we observe a very strong threshold: as we vary ε, there is a discrete jump from very likely success of recursive majority to almost-sure failure!

This result does not change if we pass back to the Poisson-birth tree used throughout the rest of this paper, rather than the k-regular tree. Here $R_{\mathrm{Pois}(k)}(q) = \mathbb{E}_{\ell \sim \mathrm{Pois}(k)}[R_\ell(q)]$, which leads to a Poisson-averaged derivative, and the same threshold in q for when the derivative passes from large to small. By Poissonization, we can also write

$$R_{\mathrm{Pois}(k)}(q) = \mathbb{P}\left[\mathrm{Pois}(kq) > \mathrm{Pois}(k(1-q))\right] + \frac{1}{2}\mathbb{P}\left[\mathrm{Pois}(kq) = \mathrm{Pois}(k(1-q))\right],$$

and the standard Chernoff bound for "Poisson races" yields, for $q = \frac{1}{2} + \nu$,

$$1 - R_{\mathrm{Pois}(k)}(q) \leq \exp\left(-k\left(\sqrt{\frac{1}{2} + \nu} - \sqrt{\frac{1}{2} - \nu}\right)^2\right)$$

$$= \exp(-k(1 - \sqrt{1 - 4\nu^2}))$$

$$= \exp(-2k(\nu^2 + O(\nu^4))),$$

which is as strong as the Hoeffding bound used above.

The results above constitute a proof for Theorem 1.5: recursive majority achieves recovery against the strong semirandom model, so long as

$$\varepsilon < \frac{1}{2} - \left(\frac{1}{2} + o(1)\right)\sqrt{\frac{\log k}{k}},$$

or, rearranging,

$$\frac{k}{\log k}(1 - 2\varepsilon)^2 > 1 + o(1),$$

with asymptotics holding as $k \to \infty$.

It is interesting to see how the semirandom model can yield recovery results across an entire family of previously studied distributions. The asymmetric broadcast trees studied in [BCMR06] may be described as Markov chains on the tree with transition matrix

$$\begin{pmatrix} 1 - \varepsilon + \delta & \varepsilon + \delta \\ \varepsilon - \delta & 1 - \varepsilon - \delta \end{pmatrix},$$

where the 'asymmetry' δ may be positive or negative. The strong semirandom adversary can simulate these models: starting from the broadcast tree model with noise $\varepsilon + |\delta|$, an adversary searches the tree top-down for either +1-to-−1 or −1-to-+1 transitions, depending on the sign of δ, and flips the entire resulting subtree with probability $2|\delta|/(\varepsilon+|\delta|)$, recursing into subtrees. Thus, the semirandom recovery results above guarantee for free that recursive majority is competitive against all of these models simultaneously, as are all algorithms robust to the strong semirandom model. By contrast, these models each admit a simple but brittle "weighted majority" reconstruction algorithm, where the weighting must be perfectly tuned to the specific model — a tiny deviation in the parameters (ε, δ) will cause those census algorithms to fail as $R \to \infty$.

9   Conclusion

In revisiting the stochastic block model from the perspective of semirandom models, we showed that there is a tension between establishing sharp thresholds and obtaining algorithms with natural robustness guarantees. There are many more classical problems in statistics and machine learning where semirandom models could offer a promising way to move beyond average-case analysis and explore issues related to robustness. In particular, belief propagation is one of the most far-reaching heuristics, but at high 'temperature' it can result in algorithms such as majority that we have shown are not robust. At low 'temperature' it leads to more robust algorithms, like recursive majority, that are connected to convex optimization. Can exploring 'temperature' further lead to interesting, provable tradeoffs between robustness and statistical power that result in a richer understanding of how belief propagation performs in practice?

Acknowledgements

References

[ABH15]   E. Abbe, A. S. Bandeira, and G. Hall. Exact recovery in the stochastic block model. *IEEE Transactions on Information Theory*, 2015.

[AL14]   A. A. Amini and E. Levina. On semidefinite relaxations for the block model. *arXiv:1406.5647*, June 2014.

[AS00]   N. Alon and J. H. Spencer. *The probabilistic method*. Wiley-Interscience series in discrete mathematics and optimization. Wiley, New York, Chichester, Weinheim, 2000.

[AS15]   E. Abbe and C. Sandon. Community detection in general stochastic block models: fundamental limits and efficient recovery algorithms. In *56th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2015.

[BCLS87] T. N. Bui, S. Chaudhuri, F. T. Leighton, and M. Sipser. Graph bisection algorithms with good average case behavior. *Combinatorica*, 7(2):171–191, 1987.

[BCMR06] C. Borgs, J. Chayes, E. Mossel, and S. Roch. The Kesten-Stigum Reconstruction Bound Is Tight for Roughly Symmetric Binary Channels. In *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 518–530, 2006.

[Bop87]   R. B. Boppana. Eigenvalues and graph bisection: An average-case analysis. In *28th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 280–285, 1987.

[BS95]   A. Blum and J. Spencer. Coloring random and semi-random k-colorable graphs. *J. Algorithms*, 19(2):204–234, 1995.

[DKMZ11] A. Decelle, F. Krzakala, C. Moore, and L. Zdeborová. Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. *Physical Review E*, 84(6):066106, December 2011.

[EKPS00] W. Evans, C. Kenyon, Y. Peres, and L. J. Schulman. Broadcasting on trees and the Ising model. *The Annals of Applied Probability*, 10(2):410–433, May 2000.

[FK00] U. Feige and R. Krauthgamer. Finding and certifying a large hidden clique in a semi-random graph. *Random Struct. Algorithms*, 16(2):195–208, 2000.

[FK01] U. Feige and J. Kilian. Heuristics for semirandom graph problems. *Journal of Computing and System Sciences*, 63:639–671, 2001.

[GV15] O. Guédon and R. Vershynin. Community detection in sparse networks via Grothendieck's inequality. *Probability Theory and Related Fields*, 2015.

[HLL83] P. W. Holland, K. B. Laskey, and S. Leinhardt. Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137, 1983.

[HWX15] B. Hajek, Y. Wu, and J. Xu. Achieving exact cluster recovery threshold via semidefinite programming. In *Information Theory (ISIT), 2015 IEEE International Symposium on*, pages 1442–1446, June 2015.

[JMR15] A. Javanmard, A. Montanari, and F. Ricci-Tersenghi. Phase Transitions in Semidefinite Relaxations. *arXiv:1511.08769*, November 2015.

[JS98] M. Jerrum and G. B. Sorkin. The metropolis algorithm for graph bisection. *Discrete Applied Mathematics*, 82(1-3):155–175, 1998.

[KMM11] A. Kolla, K. Makarychev, and Y. Makarychev. How to play unique games against a semi-random adversary: Study of semi-random models of unique games. In *IEEE 52nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 443–452, 2011.

[KS66] H. Kesten and B. P. Stigum. A Limit Theorem for Multidimensional Galton-Watson Processes. *The Annals of Mathematical Statistics*, 37(5):1211–1223, October 1966.

[LLDM08] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney. Statistical properties of community structure in large social and information networks. In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, pages 695–704. ACM, 2008.

[Mas14] L. Massoulié. Community detection thresholds and the weak Ramanujan property. *Proceedings of the 46th Annual ACM Symposium on Theory of Computing (STOC '14)*, pages 694–703, 2014.

[McS01] F. McSherry. Spectral partitioning of random graphs. In *42nd Annual Symposium on Foundations of Computer Science, FOCS 2001*, pages 529–537, 2001.

[MM05] M. Mézard and A. Montanari. Reconstruction on Trees and Spin Glass Transition. *Journal of Statistical Physics*, 124(6), 2005.

[MMV12] K. Makarychev, Y. Makarychev, and A. Vijayaraghavan. Approximation algorithms for semi-random partitioning problems. In *Proceedings of the 44th Symposium on Theory of Computing Conference (STOC)*, pages 367–384, 2012.

[MMV15a] K. Makarychev, Y. Makarychev, and A. Vijayaraghavan. Correlation clustering with noisy partial information. In *Proceedings of The 28th Conference on Learning Theory (COLT)*, pages 1321–1342, 2015.

[MMV15b] K. Makarychev, Y. Makarychev, and A. Vijayaraghavan. Learning communities in the presence of errors. *arXiv:1511.03229*, 2015.

[MNS13] E. Mossel, J. Neeman, and A. Sly. A Proof Of The Block Model Threshold Conjecture. *arXiv:1311.4115*, November 2013.

[MNS14a] E. Mossel, J. Neeman, and A. Sly. Belief propagation, robust reconstruction and optimal recovery of block models. In *Proceedings of The 27th Conference on Learning Theory, COLT 2014*, pages 356–370, 2014.

[MNS14b] E. Mossel, J. Neeman, and A. Sly. Reconstruction and estimation in the planted partition model. *Probability Theory and Related Fields*, pages 1–31, 2014.

[Mos98] E. Mossel. Recursive reconstruction on periodic trees. *Random Struct. Algorithms*, 1998.

[Mos04] E. Mossel. Survey: Information flow on trees. *arXiv:math/0406446*, June 2004.

[MS10] C. Mathieu and W. Schudy. Correlation clustering with noisy input. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 712–728, 2010.

[MS15] A. Montanari and S. Sen. Semidefinite Programs on Sparse Random Graphs. *arXiv:1504.05910*, November 2015.

[PW15] W. Perry and A. S. Wein. A semidefinite program for unbalanced multisection in the stochastic block model. *arXiv:1507.05605*, July 2015.

[Sly11] A. Sly. Reconstruction for the Potts model. *The Annals of Probability*, 39(4):1365–1406, July 2011.

A Explicit Computation of the Lower Bound

Here we compute an explicit lower bound on the separation $k^6 - k'(k, \varepsilon)^6$ between the average branching factor in the random model and that of the six-level-periodic branching rule described at the end of Section 4. We will need to assume $k \geq 9$.

Fix k, ε, δ. We will further modify the six-level-periodic rule by considering a node to be good if it has three *children* of degree not equal to 2 and we disregard the contribution of the parent to this rule, to simplify the computation; this only further reduces cutting.

In this one-period tree, the 'base' levels are 0 (the root) and 6 (the leaves), and the 'marking' level is level 3. In expectation, there are k^2 nodes on level 2. Let us focus on the subtree descending from one of these nodes. Let $p = \mathbb{P}[\text{Pois}(k) = 1]$; then by Poissonization, before cutting, each node v on level 2 gives birth to $\text{Pois}(kp)$ degree-2 children and $\text{Pois}(k(1 - p))$ degree-not-2 children. In particular, v is good with probability $\mathbb{P}[\text{Pois}(k(1 - p)) \geq 3]$. Supposing that this is the case, let us see how many leaves get cut. In expectation, v has kp degree-2 children on level 3, each with exactly one child on level 4, which is good with probability $\mathbb{P}[\text{Pois}(k(1 - p)) \geq 3]$. In that case, it gives rise to $kp + \mathbb{E}[\text{Pois}(k(1 - p)) \mid \geq 3] \, \mathbb{E}[\text{Pois}(k) \mid \neq 1]$ leaves, in expectation; but the level-3 degree-2 node and its subtree gets cut with probability $\delta\varepsilon^2$.

The expected number of leaves cut in the subtree descending from a level-0 node is:

$$k^6 - (k')^6 \geq k^2 \cdot \mathbb{P}[\mathrm{Pois}(k(1-p)) \geq 3]^2 \cdot kp \cdot \delta\varepsilon^2 \cdot (kp + \mathbb{E}[\mathrm{Pois}(k(1-p)) \mid \geq 3]\,\mathbb{E}[\mathrm{Pois}(k) \mid \neq 1])$$
$$= k^3 p \delta\varepsilon^2 \cdot \mathbb{P}[\mathrm{Pois}(k(1-p)) \geq 3]^2 \cdot (kp + \mathbb{E}[\mathrm{Pois}(k(1-p)) \mid \geq 3]\,\mathbb{E}[\mathrm{Pois}(k) \mid \neq 1]).$$

Whenever $k \geq 9$ and $k(1-2\varepsilon)^2 > 1$, we must have $\varepsilon \geq \frac{1}{3}$ and $\delta\varepsilon^2 = (1-2\varepsilon)^2 \geq \frac{1}{k}$ (from the definition (1) of δ). In this range we have

$$k^6 - (k')^6 \geq k^2 p \cdot \mathbb{P}[\mathrm{Pois}(k(1-p)) \geq 3]^2 \cdot (kp + \mathbb{E}[\mathrm{Pois}(k(1-p)) \mid \geq 3]\,\mathbb{E}[\mathrm{Pois}(k) \mid \neq 1]) \triangleq \mathcal{K}(k).$$

Now, choosing any ε such that

$$(k^6 - \mathcal{K}(k))^{1/6} < \frac{1}{(1-2\varepsilon)^2} < k,$$

the semirandom model will be impossible while the random model is possible. This concludes our explicit computation of a separation. We have not made any attempt to optimize it, and that remains an interesting open question for future work.

B   Proof of Lemma 5.8

Recall that m is the number of isolated `tagged` nodes in G, and g is the number of `good` nodes. Let α be the number of excess $+1$ spins among the isolated `tagged` nodes, i.e. there are $m/2 + \alpha$ `tagged` isolated nodes of spin $+1$ and $m/2 - \alpha$ of spin -1. Similarly let β be the number of excess $+1$ spins among the `good` nodes. Write $\alpha = \alpha_A + \alpha_{BC}$ and $\beta = \beta_A + \beta_{BC}$, where α_A denotes the number of excess $+1$ spins among the isolated `tagged` nodes of A, and likewise for the others. We will need a number of results on the sizes of these values.

Lemma B.1. *We have the following results a.a.s.*

- $|\alpha_A|, |\beta_A| = O(n^{1/8})$,

- $|\alpha_{BC}|, |\beta_{BC}| = O(n^{1/2} \log^5 n)$,

- $m, g = \Theta(n)$.

Proof. The first result is easy: α_A and β_A are bounded above by $|A|$, which is $O(n^{1/8})$ by assumption. We will establish the remaining concentration results through the bounded differences method; however, this bounded difference property will require controlling the maximum degree of a node. Note that a.a.s. every node of G_{pre} (and thus G) has degree at most $\log n$, by a Chernoff-and-union argument. Let $\mathrm{trunc}(G_{\mathrm{pre}})$ denote the graph obtained from G_{pre} by simultaneously removing all edges incident to nodes of degree larger than $\log n$. Note now that the radius-4 neighborhood of any node in $\mathrm{trunc}(G_{\mathrm{pre}})$ has size at most $O(\log^4 n)$.

Let $\mathcal{C}(\sigma, G_{\mathrm{pre}})$ denote the census (number of $+1$ spins minus number of -1 spins) among those nodes that are 'cuttable' in $\mathrm{trunc}(G_{\mathrm{pre}})$, i.e. nodes v that are degree-2 in $\mathrm{trunc}(G_{\mathrm{pre}})$, with two opposite-sign (to v) neighbors that each have at least 3 non-degree-2 neighbors. This property of v depends only on the radius-3 neighborhood in $\mathrm{trunc}(G_{\mathrm{pre}})$. Given any vertex u in G_{pre}, if we add or remove any number of edges incident to u in G_{pre}, we can only change $\mathcal{C}(\sigma, G_{\mathrm{pre}})$ by at most $O(\log^4 n)$, as we can only change the 'cuttable' status of the previous and new radius-4 neighborhoods of v in $\mathrm{trunc}(G_{\mathrm{pre}})$. (Here we need radius 4 instead of 3 because by changing edges incident to v we can push the neighbors of v over the $(\log n)$-degree cutoff.) This constitutes a bounded differences property for the function $\mathcal{C}$.

We now apply the following concentration inequality:

Lemma B.2. *Let F be a function of a random graph of size n with independent edges (not necessarily identically distributed). Suppose that, when we add or remove any number of edges incident to any given vertex v, the value of F changes by at most c. Then*

$$\mathbb{P}[|F - \mathbb{E}[F]| \geq \lambda c\sqrt{n}] \leq 2\exp(-\lambda^2/2).$$

This result is classical, and is based on applying Azuma–Hoeffding to a *vertex exposure martingale*; see for example Chapter 7 of [AS00]. Note that $\mathbb{E}[\mathcal{C}] = 0$ due to spin-reversal symmetry. It follows that $|\mathcal{C}| \leq \sqrt{n}\log^5 n$ a.a.s. Moreover, $\mathrm{trunc}(G_{\mathrm{pre}}) = G_{\mathrm{pre}}$ a.a.s., so that the true census of cuttable nodes in G_{pre} is at most $\sqrt{n}\log^5 n$ a.a.s. Our adversary turns a subset of these cuttable nodes into isolated `tagged` nodes, choosing each independently with probability δ; applying Hoeffding's inequality separately to the $+1$ and -1 cuttable nodes, we see that the census of `tagged` isolated nodes in G is $O(\sqrt{n}\log^5 n)$ a.a.s. But this census equals $\alpha_A + \alpha_{BC}$, and $|\alpha_A| \leq O(n^{1/8})$, so we have $|\alpha_{BC}| = O(\sqrt{n}\log^5 n)$ a.a.s. The same argument applies (with a strict subset of the technical mess) to show that $\beta_{BC} = O(\sqrt{n}\log^5 n)$, and that m and g concentrate within $O(\sqrt{n}\log^5 n)$ of their expectations.

It is also straightforward to see that $\mathbb{E}[m]$ and $\mathbb{E}[g]$ are $\Theta(n)$; this amounts to showing that the probability of any vertex being `tagged` isolated in G, or `good` in G, is bounded above zero as $n \to \infty$. But these two properties are local, depending only on a neighborhood in the graph, so this is clear. This completes the proof of the concentration results (Lemma B.1). □

Now we proceed to the proof of Lemma 5.8. We need to show that for a.a.e. (σ_{BC}, G), it holds for all σ_A, σ'_A that

$$\frac{\left(\frac{g/2+\beta_A+\beta_{BC}}{2}\right)^{m/2-\alpha_A-\alpha_{BC}}\left(\frac{g/2-\beta_A-\beta_{BC}}{2}\right)^{m/2+\alpha_A+\alpha_{BC}}}{\left(\frac{g/2+\beta'_A+\beta_{BC}}{2}\right)^{m/2-\alpha'_A-\alpha_{BC}}\left(\frac{g/2-\beta'_A-\beta_{BC}}{2}\right)^{m/2+\alpha'_A+\alpha_{BC}}} = 1 + o(1).$$

We will need sufficiently tight asymptotics for powers of binomials in this regime:

$$\log\left(\frac{g/2+\beta_A+\beta_{BC}}{2}\right)^{m/2-\alpha_A-\alpha_{BC}}$$
$$= \left(\frac{m}{2} - \alpha_A - \alpha_{BC}\right)\left(\log\left(\frac{g}{2}+\beta_A+\beta_{BC}\right) + \log\left(\frac{g}{2}+\beta_A+\beta_{BC}-1\right) - \log 2\right)$$
$$= \left(\frac{m}{2} - \alpha_A - \alpha_{BC}\right)\left(2\log\frac{g}{2} + \log\left(1+\frac{\beta_A+\beta_{BC}}{g/2}\right) + \log\left(1+\frac{\beta_A+\beta_{BC}-1}{g/2}\right) - \log 2\right)$$
$$= \left(\frac{m}{2} - \alpha_A - \alpha_{BC}\right)\left(2\log\frac{g}{2} + \frac{\beta_A+\beta_{BC}}{g/2} + \frac{\beta_A+\beta_{BC}-1}{g/2} - 4\frac{\beta_{BC}^2}{g^2} - \log 2\right) + O(n^{-3/8}\log^5 n),$$

from the Taylor series for the logarithm,

$$= \left(\frac{m}{2} - \alpha_A - \alpha_{BC}\right)\left(2\log\frac{g}{2} - \log 2\right) + \frac{m}{2}\left(\frac{2\beta_A+2\beta_{BC}-1}{g/2} - 4\frac{\beta_{BC}^2}{g^2}\right) - 2\frac{\alpha_{BC}\beta_{BC}}{g/2} + O(n^{-3/8}\log^5 n).$$

We now apply these asymptotics to the task at hand:

$$\log \frac{\left(\frac{g/2+\beta_A+\beta_{BC}}{2}\right)^{m/2-\alpha_A-\alpha_{BC}}\left(\frac{g/2-\beta_A-\beta_{BC}}{2}\right)^{m/2+\alpha_A+\alpha_{BC}}}{\left(\frac{g/2+\beta'_A+\beta_{BC}}{2}\right)^{m/2-\alpha'_A-\alpha_{BC}}\left(\frac{g/2-\beta'_A-\beta_{BC}}{2}\right)^{m/2+\alpha'_A+\alpha_{BC}}}$$

$$= \left(\frac{m}{2}-\alpha_A-\alpha_{BC}\right)\left(2\log\frac{g}{2}-\log 2\right)+\frac{m}{2}\left(\frac{2\beta_A+2\beta_{BC}-1}{g/2}-4\frac{\beta_{BC}^2}{g^2}\right)-2\frac{\alpha_{BC}\beta_{BC}}{g/2}$$

$$+\left(\frac{m}{2}+\alpha_A+\alpha_{BC}\right)\left(2\log\frac{g}{2}-\log 2\right)+\frac{m}{2}\left(\frac{-2\beta_A-2\beta_{BC}-1}{g/2}-4\frac{\beta_{BC}^2}{g^2}\right)-2\frac{\alpha_{BC}\beta_{BC}}{g/2}$$

$$-\left(\frac{m}{2}-\alpha'_A-\alpha_{BC}\right)\left(2\log\frac{g}{2}-\log 2\right)-\frac{m}{2}\left(\frac{2\beta'_A+2\beta_{BC}-1}{g/2}-4\frac{\beta_{BC}^2}{g^2}\right)+2\frac{\alpha_{BC}\beta_{BC}}{g/2}$$

$$-\left(\frac{m}{2}+\alpha'_A+\alpha_{BC}\right)\left(2\log\frac{g}{2}-\log 2\right)-\frac{m}{2}\left(\frac{-2\beta'_A-2\beta_{BC}-1}{g/2}-4\frac{\beta_{BC}^2}{g^2}\right)+2\frac{\alpha_{BC}\beta_{BC}}{g/2}$$

$$+O(n^{-3/8}\log^5 n)$$

$$= O(n^{-3/8}\log^5 n) = o(1),$$

as every non-error term cancels. The result now follows: if the logarithm of an expression is $o(1)$ then the expression itself is $1+o(1)$.

C   Proof of Equation (3)

Enumerate all possible values $E_1, \ldots, E_m$ for (σ_D, G). Let $x_i = \mathbb{P}[E_i]$ and let $y_i = \mathbb{P}[E_i$ and $\sigma \in \Omega]$. We know $\sum_i x_i = 1$ and $\sum_i y_i = \mathbb{P}[\sigma \in \Omega] = 1 - \epsilon_1$ for some $\epsilon_1 = o(1)$. We want to show that for some $\epsilon_2 = o(1)$ we have $\frac{y_i}{x_i} \geq 1 - \epsilon_2$ with probability $p = 1 - o(1)$; here the probability is over i drawn proportional to x_i. If I is the set of i for which $y_i/x_i \geq 1 - \epsilon_2$, we have $p = \sum_{i \in I} x_i$. This means

$$1 - \epsilon_1 = \sum_i y_i = \sum_i x_i \frac{y_i}{x_i} = \sum_{i \in I} x_i \frac{y_i}{x_i} + \sum_{i \notin I} x_i \frac{y_i}{x_i} \leq p + (1-p)(1-\epsilon_2)$$

and so $p \geq 1 - \frac{\epsilon_1}{\epsilon_2}$. We need $p = 1 - o(1)$ and $\epsilon_2 = o(1)$ so it suffices to take any ϵ_2 such that $\epsilon_2 \to 0$ and $\frac{\epsilon_1}{\epsilon_2} \to 0$, i.e. ϵ_2 goes to 0 slower than ϵ_1 does.

D   Extensions to Dissortative Models

Throughout this paper, we have assumed $a > b$; such block models are often called 'assortative'. However, everything does carry through equally in the 'dissortative' case $b > a$. Here we briefly sketch the relevant changes.

Our semirandom models are certainly designed for an assortative model, and are entirely too powerful in the dissortative case — for example, by randomly adding and removing edges appropriately, the current semirandom block model can simulate the Erdős–Rényi distribution $G(n,k)$ when starting from any dissortative model, which clearly reveals no community structure. Instead, the semirandom model in the dissortative case should add edges between communities, and remove edges within communities, so that these monotone changes are aligned with the latent structure to be recovered. Similarly, the semirandom tree models are able to cut or replace any subtree that follows a same-spin edge.

This leads to many sign changes throughout the paper. We can still couple the resulting graph neighborhoods to a tree distribution; although this tree distribution might look quite different at a

glance, it couples perfectly with an assortative tree model, corresponding via $\varepsilon \mapsto 1 - \varepsilon$, by flipping all spins at odd levels of the tree. Thus we obtain a random vs. semirandom separation for the dissortative tree model. Our recovery results for recursive majority in Section 8 carry through this coupling also, guaranteeing robust recovery by recursive anti-majority in the dissortative tree model.

Much of the "no long-range correlations" argument of Section 5 carries through unchanged, but precursors G_{pre} of G now reconnect `tagged` isolated nodes with two *same-spin* `good` nodes. Hence the new formula for $|L(\sigma, G)|$ is

$$|L(\sigma, G)| = \binom{g/2 + \beta}{2}^{m/2 + \alpha} \binom{g/2 - \beta}{2}^{m/2 - \alpha},$$

which still satisfies Lemma 5.8, by negating each β variable everywhere in the proof in Appendix B. So we also obtain a random vs. semirandom separation in the dissortative block model.

The SDP upper bounds in Section 7 carry through with very few changes of sign. One verifies that the unchanged reference objective $\frac{a-b}{2n}\sigma\sigma^\top$ does maximize $\langle S, - \rangle$ where S is the matrix form of any semirandom change as redefined above. A sign does flip in the proof of Proposition 7.9: we now have

$$\gamma_v = \frac{b - a}{2} - O(\log^2 n / \sqrt{n}).$$

Overall we obtain the same guarantees on semirandom partial recovery as in the assortative model, requiring $b > 20$ rather than $a > 20$.