

# Subsampled Power Iteration: a Unified Algorithm for Block Models and Planted CSP's

Vitaly Feldman\*      Will Perkins†      Santosh Vempala‡

## Abstract

We present an algorithm for recovering planted solutions in two well-known models, the stochastic block model and planted constraint satisfaction problems, via a common generalization in terms of random bipartite graphs. Our algorithm matches up to a constant factor the best-known bounds for the number of edges (or constraints) needed for perfect recovery and its running time is linear in the number of edges used. The time complexity is significantly better than both spectral and SDP-based approaches.

The main contribution of the algorithm is in the case of unequal sizes in the bipartition that arises in our reduction from the planted CSP. Here our algorithm succeeds at a significantly lower density than the spectral approaches, surpassing a barrier based on the spectral norm of a random matrix.

Other significant features of the algorithm and analysis include (i) the critical use of power iteration with subsampling, which might be of independent interest; its analysis requires keeping track of multiple norms of an evolving solution (ii) the algorithm can be implemented statistically, i.e., with very limited access to the input distribution (iii) the algorithm is extremely simple to implement and runs in linear time, and thus is practical even for very large instances.

---

\*IBM Research - Almaden

†University of Birmingham and the IMA.

‡Georgia Tech. Supported in part by NSF award CCF-1217793.

# 1 Introduction

Partitioning a graph into parts based on the density of the edges within and between the parts is a fundamental algorithmic task both in its own right as a method of clustering data into similar pieces, and as a powerful subroutine of divide-and-conquer algorithms. There are many choices for the number of parts required and the measure of the quality of a partition, and different choices give rise to algorithmic problems such as Max Clique, Max Cut, Uniform Sparsest Cut, and Min Bisection.

While finding an optimal graph partition is often an NP-hard problem in the worst case, the average-case study of graph partitioning problems is particularly rich, as the underlying distributions come from natural and widely studied models of random graphs (we review the previous work in Section 1.2).

The simplest model is the stochastic block model: partition a set of vertices into two equal parts  $A$  and  $B$ , and add edges independently, with probability  $p$  for an edge within a part, and  $q \neq p$  for a crossing edge. The algorithmic task is to recover the partition given the random graph. Generalizations include parts of unequal size, more than two parts, and more than two edge probabilities.

Another broad and fundamental class of algorithmic problems is the class of boolean Constraint Satisfaction Problems (CSP's, defined precisely below). The average-case complexity of  $k$ -CSP's is a large area of research that intersects cryptography, computational complexity, probabilistic combinatorics and statistical physics. In the planted  $k$ -SAT problem each constraint is a disjunction of  $k$  literals, variables or their negations, eg.  $\{\bar{x}_5, x_6, \bar{x}_{10}\}$  and is referred to as  $k$ -clause. A random instance of this problem is produced by choosing a random and uniform assignment  $\sigma$  and then selecting  $k$ -clauses at random independently (but not necessarily uniformly) from the set of  $k$ -clauses satisfied by  $\sigma$ . This distribution is guaranteed to have at least one satisfying assignment,  $\sigma$ . In the 'noisy' version of the problem unsatisfied clauses are also included with some probability. The algorithmic task is to recover the planted assignment  $\sigma$ . An additional model of planted CSP's we consider is Goldreich's pseudorandom generator [42] that has been studied in cryptography. We describe it in more detail below.

## 1.1 Our results and techniques

We propose a natural bipartite stochastic block model that generalizes the classic stochastic block model defined above. The key motivation for the study of this model is that the two types of planted  $k$ -CSP's can be reduced to our block model, thus unifying graph partitioning and planted CSP's into one problem. We then give an algorithm for solving random instances of the model.

The model begins with two vertex sets,  $V_1$  and  $V_2$  (of possibly unequal size), each with a balanced partition,  $(A_1, B_1)$  and  $(A_2, B_2)$  respectively. Edges are added independently at random between  $V_1$  and  $V_2$  with probabilities that depend on which parts the endpoints are in: edges between  $A_1$  and  $A_2$  or  $B_1$  and  $B_2$  are added with probability  $\delta p$ , while the other edges are added with probability  $(2 - \delta)p$ , where  $\delta \in [0, 2]$  and  $p$  is the overall edge density. To obtain the stochastic block model we can identify  $V_1$  and  $V_2$ . To reduce planted CSP's to this model, we first reduce the problem to an instance of noisy  $r$ -XOR-SAT, where  $r$  is the complexity parameter of the planted CSP distribution defined in [35] (see Sec. 2 for details). We then identify  $V_1$  with literals, and  $V_2$  with  $(r - 1)$ -tuples of literals, and add an edge between literal  $l \in V_1$  and tuple  $t \in V_2$  when the  $r$ -clause consisting of their union appears in the formula. The reduction leads to a bipartition with  $V_2$  much larger than  $V_1$ .

Our algorithm is based on applying power iteration with a sequence of matrices subsampled

from the original adjacency matrix. This is in contrast to previous algorithms that compute the eigenvectors (or singular vectors) of the full adjacency matrix. Our algorithm has several advantages. Such an algorithm, for the special case of square matrices, was previously proposed and analyzed in a different context by Korada et al [48].

- Up to a constant factor, the algorithm matches the best-known (and in some cases the best-possible) edge or constraint density needed for complete recovery of the planted partition or assignment. The algorithm for planted CSP's finds the planted assignment using  $O(n^{r/2} \cdot \log n)$  clauses for a clause distribution of complexity  $r$  (see Sec. 2 for the formal definition), nearly matching computational lower bounds for SDP hierarchies [60] and the class of statistical algorithms [35].
- The algorithm is fast, running in time linear in the number of edges or constraints used, unlike other approaches that require computing eigenvectors or solving semi-definite programs.
- The algorithm is conceptually simple and very easy to describe and implement. In fact it can be implemented in the statistical query model, with very limited access to the input graph [35].
- It is based on the idea of iteration with subsampling which may have further applications in the design and analysis of algorithms.
- Most notably, the algorithm succeeds where generic spectral approaches fail. For the case of the planted CSP, when  $|V_2| \gg |V_1|$ , our algorithm succeeds at a polynomial factor sparser density than the approaches of McSherry [55], Coja-Oghlan [19], and Vu [64]. The algorithm succeeds despite the fact that the 'energy' of the planted vector with respect to the random adjacency matrix is far below the spectral norm of the matrix. In previous analyses, this was believed to indicate failure of the spectral approach. For a full discussion, see Section 5.

The remainder of the paper is organized as follows:

- In Section 1.2, we review previous work.
- In Section 2 we formally define the model and present the main theorems.
- In Section 3 we describe the algorithm and analyze its performance.
- In Section 4 we give the reduction of the planted  $k$ -CSP problems to the bipartite stochastic block model.
- In Section 5 we compare our algorithm to other spectral approaches.
- In Section 6 we present full details of the analysis.

## 1.2 Related work

### Planted partitioning

The stochastic block model was introduced in [43]. Boppana [15] gave a spectral-based algorithm for the model, and Jerrum and Sorkin [45] gave a Metropolis approach. Dyer and Frieze [30] and Blum and Spencer [13] give algorithms for the related planted  $k$ -coloring model in which the vertex set is partitioned into  $k$  equal parts and then edges crossing the partition are added independently

at random while edges within the partition are forbidden. Alon and Kahale [5] gave a spectral algorithm for this problem.

Later algorithms [25, 32, 18, 16, 24] improved either the running time or the density at which the algorithms succeed. Of particular note is McSherry’s algorithm [55] which is based on a low-rank projection and is a generic algorithm for many planted partitioning problems, including the stochastic block model, the planted coloring problem, and the planted clique problem. Coja-Oghlan [19] gave a refined general purpose partitioning algorithm and showed that the planted partition in the stochastic block model can be partially recovered when the average degree is just a constant. Vu [64] recently gave a simple SVD-based general partitioning algorithm.

While all of the above works seek to recover the partition at as low a density as possible, only recently have sharp thresholds for the possibility of recovery been identified. Based on ideas from statistical physics, Decelle et al. [29] conjectured that in fact there is a sharp threshold for efficient recovery in the stochastic block model: if  $p = a/n, q = b/n$ , and  $(a - b)^2 < 2(a + b)$  then any non-trivial recovery of the planted partition is impossible, while if  $(a - b)^2 > 2(a + b)$  then there is an efficient algorithm (polynomial in the size of the graph) that gives a partition with significant correlation to the planting. Mossel, Sly, and Neeman proved the lower bound [58], and then Massoulié [54] and Mossel, Neeman, Sly [56] independently analyzed algorithms proving the upper bound. See also [59, 51] for more on related algorithms. Recent work has found algorithms that succeed at the optimal threshold for complete recovery [1, 57].

### Planted $k$ -CSP’s

A width- $k$  CSP is defined by a set of  $m$  predicates denoted by  $P_1, \dots, P_m$  and a set of  $m$   $k$ -tuples of boolean variables from the set  $V = \{x_1, \dots, x_n\}$  denoted by  $C_1, \dots, C_m$ . Each predicate  $P_i$  is a function from  $\{\pm 1\}^k$  to  $\{\pm 1\}$ . Identifying  $+1$  with TRUE and  $-1$  with FALSE, a predicate  $P_i$  is satisfied by an assignment  $\sigma : V \rightarrow \{\pm 1\}$  if the evaluation of the predicate  $P_i$  on the values assigned by  $\sigma$  to the  $k$ -tuple of variables  $C_i = (x_{i_1}, \dots, x_{i_k})$  is TRUE. Given such a  $k$ -CSP the algorithmic task is to find an assignment  $\sigma$  that maximizes the number of satisfied constraints.

It was noted in [9] that drawing satisfied  $k$ -SAT clauses uniformly at random from all those satisfied by  $\sigma$  does not result in a difficult algorithmic problem even if the number of observed clauses is relatively small (simply taking the majority vote for each variable suffices; see [10] for optimal statistical tests in this setting). However, by changing the proportions of clauses depending on the number of satisfied literals under  $\sigma$ , one can create a more challenging distribution over instances. Such ‘quiet plantings’ were further studied in [46, 2, 52, 50]. Algorithms for solving instances with various values of relative proportions for planted 3-SAT were given in [36, 49, 20]. Following [35], we define such problems using a *planting distribution*  $Q$ . This distribution is defined over  $\{\pm 1\}^k$  and for a vector  $z$  it gives the proportion of clauses in which the values  $\sigma$  assigns to the  $k$ -tuple of literals in the clause is  $z$  (see Section 2 for the formal definition).

A related class of problems is one in which for some fixed predicate  $P$ , an instance is generated by choosing a planted assignment  $\sigma$  uniformly at random and generating a set of  $m$  random and uniform  $P$ -constraints. That is, each constraint is of the form  $P(x_{i_1}, \dots, x_{i_k}) = P(\sigma_{i_1}, \dots, \sigma_{i_k})$ , where  $(x_{i_1}, \dots, x_{i_k})$  is a randomly and uniformly chosen  $k$ -tuple of variables (without repetitions). The algorithmic problem is to determine  $\sigma$  given the  $m$   $k$ -tuples of variables and the corresponding values of  $P$  on those tuples. Goldreich [42] proposed a one-way function based on the apparent hardness of these problems. In his proposal the predicate is chosen randomly. The hardness of such problems for other predicates, most notably noisy  $k$ -XOR-SAT, has been used in cryptographic applications including public key cryptosystems [4, 7], and secure two-party computation [44]. It has also been used to derive hardness of approximation [6] (for public discussions of these problems/assumptions

see [8, 63]). Problems of this type are usually referred to as Goldreich’s pseudorandom generator (PRG).

Bogdanov and Qiao [14] show that an SDP-based algorithm of Charikar and Wirth [17] can be used to find the planted assignment for any predicate that is *not* pairwise-independent using  $m = O(n)$  constraints. The same approach can be used to recover the input for any  $t$ -wise independent predicate using  $O(n^{(t+1)/2})$  evaluations via the folklore birthday “paradox”-based reduction to  $t = 1$  (see [60] for details).

Finding the planted assignment in a randomly generated  $k$ -SAT formula is at least as hard as distinguishing between a satisfiable formula generated using a planted assignment and a randomly and uniformly generated  $k$ -SAT formula. Even this seemingly easier problem appears to be hard for certain planting distributions. This problem is a special case of another well-studied hard problem: refuting the satisfiability of SAT formulas in which the goal is to distinguish a satisfiable formula from a randomly an uniformly generated one (see [35] for the details of the connection).

It is important to note that in planted  $k$ -CSP’s the planted assignment becomes identifiable with high probability after at most  $O(n \log n)$  random clauses yet the best known efficient algorithms require  $n^{\Omega(r/2)}$  clauses. Problems exhibiting this type of behavior have attracted significant interest in learning theory [12, 28, 61, 33, 62, 11, 26] and some of the recent hardness results are based on the conjectured computational hardness of the  $k$ -SAT refutation problem [26, 27].

The connection of planted CSP’s to graph partitioning is that many algorithms for planted CSP’s use graph partitioning, and spectral graph partitioning in particular, as a subroutine. Examples of such algorithms for some classes of constraint distributions include Flaxman’s algorithm for planted 3-SAT [36], Krivelevich and Vilenchik’s algorithm [49] that runs in expected polynomial time, and the algorithm of Coja-Oghlan, Cooper, Frieze [20] for planted 3-SAT distributions that include the quiet plantings described above. Many of the same spectral techniques have been applied here as well for the SAT refutation problem [40, 41, 21, 31, 39, 23].

## Comparison with previous work

The algorithm of Mossel, Neeman, and Sly [56] for the case  $n_1 = n_2$  also runs in near linear time, while other known algorithmic approaches for planted partitioning that succeed near the optimal edge density [55, 19, 54] perform eigenvector or singular vector computations and thus require superlinear time, though a careful randomized implementation of low-rank approximations can reduce the running time of McSherry’s algorithm substantially [3].

For planted satisfiability, the algorithm of Flaxman for planted 3-SAT works for a subset of planted distributions (those with distribution complexity at most 2 in our definition below) using  $O(n)$  constraints, while the algorithm of Coja-Oghlan, Cooper, and Frieze [20] works for planted 3-SAT distributions that exclude unsatisfied clauses and uses  $O(n^{3/2} \ln^{10} n)$  constraints.

The only previous algorithm that finds the planted assignment in Goldreich’s PRG for all predicates is the SDP-based algorithm of Bogdanov and Qiao [14] with the folklore generalization to  $r$ -wise independent predicates (*cf.* [60]). Similar to our algorithm, it uses  $\tilde{O}(n^{r/2})$  constraints. This algorithm effectively solves the noisy  $r$ -XOR-SAT instance and therefore can be also used to solve our general version of planted satisfiability using  $\tilde{O}(n^{r/2})$  clauses (via the reduction in Section 4). Notably for both this algorithm and ours, having a completely satisfying planted assignment plays no special role: the number of constraints required depends only on the distribution complexity.

To the best of our knowledge, our algorithm is the first for the planted  $k$ -SAT problem that runs in linear time in the number of constraints used.

Our algorithm is arguably simpler than the approach in [14] and substantially improves the running time even for small  $k$ . Another advantage of our approach is that it can be implemented

using restricted access to the distribution of constraints referred to as statistical queries [47, 34]. Roughly speaking, for the planted SAT problem this access allows an algorithm to evaluate multi-valued functions of a single clause on randomly drawn clauses or to estimate expectations of such functions, *without direct access to the clauses themselves*. Recently, in [35], lower bounds on the number of clauses necessary for a polynomial-time statistical algorithm to solve planted  $k$ -CSPs were proved. It is therefore important to understand the power of such algorithms for solving planted  $k$ -CSPs. A statistical implementation of our algorithm gives an upper bound that nearly matches the lower bound for the problem. See [35] for the formal details of the model and statistical implementation.

Korada, Montanari, and Oh [48] analyzed the ‘Gossip PCA’ algorithm, which for the special case of an equal bipartition is the same as our subsampled power iteration. The assumptions, model, and motivation in the two papers are different and the results incomparable. In particular, while our focus and motivation are on general (nonsquare) matrices, their work considers extracting a planting of rank  $k$  greater than 1 in the square setting. Their results also assume an initial vector with non-trivial correlation with the planted vector. The nature of the guarantees is also different.

Two other algorithms are similar in spirit to our approach: clustering via matrix powering of Zhou and Woodruff [65] and ‘Power Iteration Clustering’ of Lin and Cohen [53]. In each, partitioning is performed by multiplying an initial vector by the adjacency matrix of the random graph repeatedly. These methods are similar to ours in their simplicity; the subsampling in our algorithm allows us to carry out a rigorous analysis through many more iterations.

## 2 Model and results

### Bipartite stochastic block model

**Definition 1.** For  $\delta \in [0, 2] \setminus \{1\}$ ,  $n_1, n_2$  even, and  $\mathcal{P}_1 = (A_1, B_1)$ ,  $\mathcal{P}_2 = (A_2, B_2)$  bipartitions of vertex sets  $V_1, V_2$  of size  $n_1, n_2$  respectively, we define the bipartite stochastic block model  $B(n_1, n_2, \mathcal{P}_1, \mathcal{P}_2, \delta, p)$  to be the random graph in which edges between vertices in  $A_1$  and  $A_2$  and  $B_1$  and  $B_2$  are added independently with probability  $\delta p$  and edges between vertices in  $A_1$  and  $B_2$  and  $B_1$  and  $A_2$  with probability  $(2 - \delta)p$ .

Here  $\delta$  is a fixed constant while  $p$  will tend to 0 as  $n_1, n_2 \rightarrow \infty$ . Note that setting  $n_1 = n_2 = n$ , and identifying  $A_1$  and  $A_2$  and  $B_1$  and  $B_2$  gives the usual stochastic block model (with loops allowed); for edge probabilities  $a/n$  and  $b/n$ , we have  $\delta = 2a/(a+b)$  and  $p = (a+b)/2n$ , the overall edge density. For our application to  $k$ -CSP’s, it will be crucial to allow vertex sets of very different sizes, i.e.  $n_2 \gg n_1$ .

The algorithmic task for the bipartite block model is to recover one or both partitions (completely or partially) using as few edges and as little computational time as possible. In this work we will assume that  $n_1 \leq n_2$ , and we will be concerned with the algorithmic task of recovering the partition  $\mathcal{P}_1$  completely, as this will allow us to solve the planted  $k$ -CSP problems described below. We define complete recovery of  $\mathcal{P}_1$  as finding the exact partition with high probability over the randomness in the graph and in the algorithm.

**Theorem 1.** Assume  $n_1 \leq n_2$ . There is a constant  $C$  so that the Subsampled Power Iteration algorithm described below completely recovers the partition  $\mathcal{P}_1$  in the bipartite stochastic block model  $B(n_1, n_2, \mathcal{P}_1, \mathcal{P}_2, \delta, p)$  with probability  $1 - o(1)$  as  $n_1 \rightarrow \infty$  when  $p \geq \frac{C \log n_1}{(\delta-1)^2 \sqrt{n_1 n_2}}$ . Its running time is  $O\left(\sqrt{n_1 n_2} \cdot \frac{\log n_1}{(\delta-1)^2}\right)$ .

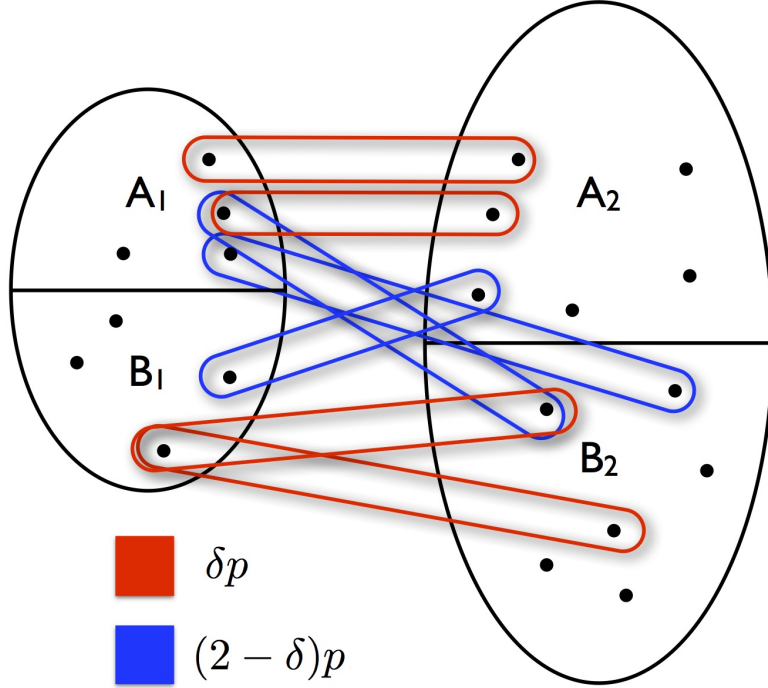


Figure 1: Bipartite stochastic block model.

Note that for the usual stochastic block model this gives an algorithm using  $O(n \log n)$  edges and  $O(n \log n)$  time, which is the best possible for complete recovery since that many edges are needed for every vertex to appear in at least edge. With edge probabilities  $a \log n/n$  and  $b \log n/n$ , our results requires  $(a - b)^2 \geq C(a + b)$  for some absolute constant  $C$ , matching the dependence on  $a$  and  $b$  in [15, 55] (see [1] for a discussion of the best possible threshold for complete recovery).

For any  $n_1, n_2$ , at least  $\sqrt{n_1 n_2}$  edges are necessary for even non-trivial partial recovery, as below that threshold the graph consists only of small components (and even if a correct partition is found on each component, correlating the partitions of different components is impossible). Similarly at least  $\Omega(\sqrt{n_1 n_2} \log n_1)$  are needed for complete recover of  $\mathcal{P}_1$  since below that density, there are vertices in  $V_1$  joined only to vertices of degree 1 in  $V_2$ .

For very lopsided graphs, with  $n_2 \gg n_1 \log^2 n_1$ , the running time is sublinear in the size of  $V_2$ ; this requires careful implementation and is essential to achieving the running time bounds for planted CSP's described below.

### Planted $k$ -CSP's

We now describe a general model for planted satisfiability problems introduced in [35]. For an integer  $k$ , let  $\mathcal{C}_k$  be the set of all ordered  $k$ -tuples of literals from  $x_1, \dots, x_n, \bar{x}_1, \dots, \bar{x}_n$  with no repetition of variables. For a  $k$ -tuple of literals  $C$  and an assignment  $\sigma$ ,  $\sigma(C)$  denotes the vector of values that  $\sigma$  assigns to the literals in  $C$ . A planting distribution  $Q : \{\pm 1\}^k \rightarrow [0, 1]$  is a probability distribution over  $\{\pm 1\}^k$ .

**Definition 2.** Given a planting distribution  $Q : \{\pm 1\}^k \rightarrow [0, 1]$ , and an assignment  $\sigma \in \{\pm 1\}^n$ , we define the random constraint satisfaction problem  $F_{Q, \sigma}(n, m)$  by drawing  $m$   $k$ -clauses from  $\mathcal{C}_k$

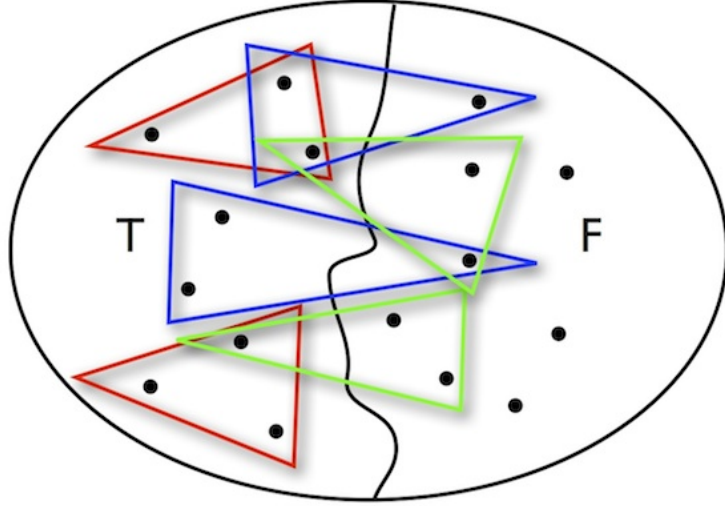


Figure 2: Planted random 3-SAT.

independently according to the distribution

$$Q_\sigma(C) = \frac{Q(\sigma(C))}{\sum_{C' \in \mathcal{C}_k} Q(\sigma(C'))}$$

where  $\sigma(C)$  is the vector of values that  $\sigma$  assigns to the  $k$ -tuple of literals comprising  $C$ .

**Definition 3.** The distribution complexity  $r(Q)$  of the planting distribution  $Q$  is the smallest integer  $r \geq 1$  so that there is some  $S \subseteq [k]$ ,  $|S| = r$ , so that the discrete Fourier coefficient  $\hat{Q}(S)$  is non-zero.

In other words, the distribution complexity of  $Q$  is  $r$  if  $Q$  is an  $(r - 1)$ -wise independent distribution on  $\{\pm 1\}^k$  but not an  $r$ -wise independent distribution. The uniform distribution over all clauses,  $Q \equiv 2^{-k}$ , has  $\hat{Q}(S) = 0$  for all  $|S| \geq 1$ , and so we define its complexity to be  $\infty$ . The uniform distribution does not reveal any information about  $\sigma$ , and so inference is impossible. For any  $Q$  that is not the uniform distribution over clauses, we have  $1 \leq r(Q) \leq k$ .

Note that the uniform distribution on  $k$ -SAT clauses with at least one satisfied literal under  $\sigma$  has distribution complexity  $r = 1$ .  $r = 1$  means that there is a bias towards either true or false literals. In this case, a very simple algorithm is effective: for each variable, count the number of times it appears negated and not negated, and take the majority vote. For distributions with complexity  $r \geq 2$ , the expected number of true and false literals in the random formula are equal and so this simple algorithm fails.

**Theorem 2.** For any planting distribution  $Q$ , there exists an algorithm that for any assignment  $\sigma$ , given an instance of  $F_{Q,\sigma}(n, m)$  completely recovers the planted assignment  $\sigma$  for  $m = O(n^{r/2} \log n)$  using  $O(n^{r/2} \log n)$  time, where  $r \geq 2$  is the distribution complexity of  $Q$ . For distribution complexity  $r = 1$ , there is an algorithm that gives non-trivial partial recovery with  $O(n^{1/2})$  constraints and complete recovery with  $O(n \log n)$  constraints.

We also show that the same result applies to recovering the planted assignment in Goldreich's PRG defined above.



**Theorem 3.** *For any predicate  $P : \{\pm 1\}^k \rightarrow \{\pm 1\}$ , there exists an algorithm that for any assignment  $\sigma$ , given  $m$  random  $P$ -constraints completely recovers the planted assignment  $\sigma$  for  $m = O(n^{r/2} \log n)$  and using  $O(n^{r/2} \log n)$  time, where  $r \geq 2$  is the degree of the lowest-degree non-zero Fourier coefficient of  $P$ . For  $r = 1$ , the algorithm gives non-trivial partial recovery with  $O(n^{1/2})$  constraints and complete recovery with  $O(n \log n)$  constraints.*

### 3 The algorithm

We now present our algorithm for the bipartite stochastic block model. We define vectors  $u$  and  $v$  of dimension  $n_1$  and  $n_2$  respectively, indexed by  $V_1$  and  $V_2$ , with  $u_i = 1$  for  $i \in A_1$ ,  $u_i = -1$  for  $i \in B_1$ , and similarly for  $v$ . To recover the partition  $\mathcal{P}_1$  it suffices to find either  $u$  or  $-u$ . We will find this vector by multiplying a random initial vector  $x_0$  by a sequence of centered adjacency matrices and their transposes.

We form these matrices as follows: let  $G_p$  be the random bipartite graph drawn from the model  $B(n_1, n_2, \mathcal{P}_1, \mathcal{P}_2, \delta, p)$ , and  $T$  a positive integer. Then form  $T$  different bipartite graphs  $G_1, \dots, G_T$  on the same vertex sets  $V_1, V_2$  by placing each edge from  $G_p$  uniformly and independently at random into one of the  $T$  graphs. The resulting graphs have the same marginal distribution.

Next we form the  $n_1 \times n_2$  adjacency matrices  $A_1, \dots, A_T$  for  $G_1, \dots, G_T$  with rows indexed by  $V_1$  and columns by  $V_2$  with a 1 in entry  $(i, j)$  if vertex  $i \in V_1$  is joined to vertex  $j \in V_2$ . Finally we center the matrices by defining  $M_i = A_i - \frac{p}{T} J$  where  $J$  is the  $n_1 \times n_2$  all ones matrix.

In the bipartite block model, these subsampled matrices are nearly independent (see Lemma 2), leading to a strong bound on the number of iterations required to solve the problem. The subsampling also mitigates the influence of high-degree vertices leading to significant improvement over the spectral approach for a large subclass of planted CSP's.

The analysis of the algorithm proceeds by tracking a potential function,  $U_i = x^i \cdot u$  for a sequence of unit vectors  $x^0, x^1, \dots$  of dimension  $n_1$ . We must bound various norms of the  $x^i$ 's as well as norms of a sequence of auxiliary vectors  $y^1, y^2, \dots$  of dimension  $n_2$ . We use superscripts to denote the current step of the iteration and subscripts for the components of the vectors, so  $x_j^i$  is the  $j$ th coordinate of the vector after the  $i$ th iteration.

The basic iterative steps are the multiplications  $y = M^T x$  and  $x = M y$ .

**Algorithm: Subsampled Power Iteration.**

1. Form  $T = 10 \log n_1$  matrices  $M_1, \dots, M_T$  by uniformly and independently assigning each edge of the bipartite block model to a graph  $G_1, \dots, G_T$ , then forming the matrices  $M_i = A_i - \frac{p}{T}J$ , where  $A_i$  is the adjacency matrix of  $G_i$  and  $J$  is the all ones matrix.
2. Sample  $x \in \{\pm 1\}^{n_1}$  uniformly at random and let  $x^0 = \frac{x}{\sqrt{n_1}}$ .
3. For  $i = 1$  to  $T/2$  let

$$y^i = \frac{M_{2i-1}^T x^{i-1}}{\|M_{2i-1}^T x^{i-1}\|}; \quad x^i = \frac{M_{2i} y^i}{\|M_{2i} y^i\|}; \quad z^i = \text{sgn}(x^i).$$

4. For each coordinate  $j \in [n_1]$  take the majority vote of the signs of  $z_j^i$  for all  $i \in \{T/4, \dots, T/2\}$  and call this vector  $\bar{v}$ :

$$\bar{v}_j = \text{sgn} \left( \sum_{i=T/2}^T z_j^i \right).$$

5. Return the partition indicated by  $\bar{v}$ .

The analysis of the resampled power iteration algorithm proceeds in four phases, during which we track the progress of two vectors  $x^i$  and  $y^i$ , as measured by their inner product with  $u$  and  $v$  respectively. We define  $U_i := u \cdot x^i$  and  $V_i := v \cdot y^i$ . Here we give an overview of each phase; the complete analysis is in Section 6.

- **Phase 1.** Within  $\log n_1$  iterations,  $|U_i|$  reaches  $\log n_1$ . We show that conditioned on the value of  $U_i$ , there is at least a  $1/2$  chance that  $|U_{i+1}| \geq 2|U_i|$ ; that  $U_i$  never gets too small; and that in  $\log n_1$  steps, a run of  $\log \log n_1$  doublings pushes the magnitude of  $U_i$  above  $\log n_1$ .
- **Phase 2.** After reaching  $\log n_1$ ,  $|U_i|$  makes steady, predictable progress, doubling at each step whp until it reaches  $\Theta(\sqrt{n_1})$ , at which point we say  $x^i$  has strong correlation with  $u$ .
- **Phase 3.** Once  $x^i$  is strongly correlated with  $u$ , we show that  $z^{i+1}$  agrees with either  $u$  or  $-u$  on a large fraction of coordinates.
- **Phase 4.** We show that taking the majority vote of the coordinate-by-coordinate signs of  $z^i$  over  $O(\log n_1)$  additional iterations gives complete recovery whp.

**Running time**

If  $n_2 = \Theta(n_1)$ , then a straightforward implementation of the algorithm runs in time linear in the number of edges used: each entry of  $x^i = M y^i$  (resp.  $y^i = M^T x^{i-1}$ ) can be computed as a sum over the edges in the graph associated with  $M$ . The rounding and majority vote are both linear in  $n_1$ .

However, if  $n_2 \gg n_1$ , then simply initializing the vector  $y^i$  will take too much time. In this case, we have to implement the algorithm more carefully.

Say we have a vector  $x^{i-1}$  and want to compute  $x^i = M_{2i} y^i$  without storing the vector  $y^i$ . Instead of computing  $y^i = M_{2i-1}^T x^{i-1}$ , we create a set  $S^i \subset V_2$  of all vertices with degree at least

1 in the current graph  $G_{2i-1}$  corresponding to the matrix  $M_{2i-1}$ . The size of  $S^i$  is bounded by the number of edges in  $G_{2i-1}$ , and checking membership can be done in constant time with a data structure of size  $O(|S^i|)$  that requires expected time  $O(|S^i|)$  to create [38].

Recall that  $M_{2i-1} = A_{2i-1} - qJ$ . Then we can write

$$y^i = (A_{2i-1} - qJ)^T x^{i-1} = \hat{y} - q \left( \sum_{j=1}^{n_1} x_j^{i-1} \right) \mathbf{1}_{n_2} = \hat{y} - qL\mathbf{1}_{n_2},$$

where  $\hat{y}$  is 0 on coordinates  $j \notin S^i$ ,  $L = \sum_{j=1}^{n_1} x_j^{i-1}$ , and  $\mathbf{1}_{n_2}$  is the all ones vector of length  $n_2$ .

Then to compute  $x^i = M_{2i}y^i$ , we write

$$\begin{aligned} x^i &= (A_{2i} - qJ)y^i = (A_{2i} - qJ)(\hat{y} - qL\mathbf{1}_{n_2}) \\ &= (A_{2i} - qJ)\hat{y} - qLA_{2i}\mathbf{1}_{n_2} + q^2LJ\mathbf{1}_{n_2} \\ &= A_{2i}\hat{y} - qJ\hat{y} - qLA_{2i}\mathbf{1}_{n_2} + q^2Ln_2\mathbf{1}_{n_1} \end{aligned}$$

We bound the running time of the computation as follows: we can compute  $\hat{y}$  in linear time in the number of edges of  $G_{2i-1}$  using  $S^i$ . Given  $\hat{y}$ , computing  $A_{2i}\hat{y}$  is linear in the number of edges of  $G_{2i}$  and computing  $qJ\hat{y}$  is linear in the number of non-zero entries of  $\hat{y}$ , which is bounded by the number of edges of  $G_{2i-1}$ . Computing  $L = \sum_{j=1}^{n_1} x_j^{i-1}$  is linear in  $n_1$  and gives  $q^2Ln_2\mathbf{1}_{n_1}$ . Computing  $qLA_{2i}\mathbf{1}_{n_2}$  is linear in the number of edges of  $G_{2i}$ . All together this gives our linear time implementation.

## 4 Reduction of planted $k$ -CSP's to the block model

Here we describe how solving the bipartite block model suffices to solve the planted  $k$ -CSP problems.

Consider a planted  $k$ -SAT problem  $F_{Q,\sigma}(n, m)$  with distribution complexity  $r$ . Let  $S \subseteq [k]$ ,  $|S| = r$ , be such that  $\hat{Q}(S) = \eta \neq 0$ . Such an  $S$  exists from the definition of the distribution complexity. We assume that we know both  $r$  and this set  $S$ , as trying all possibilities (smallest first) requires only a constant factor ( $2^r$ ) more time.

We will restrict each  $k$ -clause in the formula to an  $r$ -clause, by taking the  $r$  literals specified by the set  $S$ . If the distribution  $Q$  is known to be symmetric with respect to the order of the  $k$ -literals in each clause, or if clauses are given as unordered sets of literals, then we can simply sample a random set of  $r$  literals (without replacement) from each clause.

We will show that restricting to these  $r$  literals from each  $k$ -clause induces a distribution on  $r$ -clauses defined by  $Q^\delta : \{\pm 1\}^r \rightarrow \mathbb{R}^+$  of the form  $Q^\delta(C) = \delta/2^r$  for  $|C|$  even,  $Q^\delta(C) = (2 - \delta)/2^r$  for  $|C|$  odd, for some  $\delta \in [0, 2]$ ,  $\delta \neq 1$ , where  $|C|$  is the number of TRUE literals in  $C$  under  $\sigma$ . This reduction allows us to focus on algorithms for the specific case of a parity-based distribution on  $r$ -clauses with distribution complexity  $r$ .

Recall that for a function  $f : \{-1, 1\}^k \rightarrow \mathbb{R}$ , its Fourier coefficients are defined for each subset  $S \subset [k]$  as

$$\hat{f}(S) = \mathbb{E}_{x \sim \{-1, 1\}^k} [f(x)\chi_S(x)]$$

where  $\chi_S$  are the Walsh basis functions of  $\{\pm 1\}^k$  with respect to the uniform probability measure, i.e.,  $\chi_S(x) = \prod_{i \in S} x_i$ .

**Lemma 1.** *If the function  $Q : \{\pm 1\}^k \rightarrow \mathbb{R}^+$  defines a distribution  $Q_\sigma$  on  $k$ -clauses with distribution complexity  $r$  and planted assignment  $\sigma$ , then for some  $S \subseteq [k]$ ,  $|S| = r$  and  $\delta \in [0, 2] \setminus \{1\}$ , choosing  $r$  literals with indices in  $S$  from a clause drawn randomly from  $Q_\sigma$  yields a random  $r$ -clause from  $Q_\sigma^\delta$ .*

*Proof.* From Definition 3 we have that there exists an  $S$  with  $|S| = r$  such that  $\hat{Q}(S) \neq 0$ . Note that by definition,

$$\begin{aligned} \hat{Q}(S) &= \mathbb{E}_{x \sim \{\pm 1\}^k} [Q(x) \chi_S(x)] = \frac{1}{2^k} \sum_{x \in \{\pm 1\}^k} Q(x) \chi_S(x) \\ &= \frac{1}{2^k} \left( \sum_{x: \in \{\pm 1\}^k: x_S \text{ even}} Q(x) - \sum_{x: \in \{\pm 1\}^k: x_S \text{ odd}} Q(x) \right) \\ &= \frac{1}{2^k} (\Pr[x_S \text{ even}] - \Pr[x_S \text{ odd}]) \end{aligned}$$

where  $x_S$  is  $x$  restricted to the coordinates in  $S$ , and so if we take  $\delta = 1 + 2^k \hat{Q}(S)$ , the distribution induced by restricting  $k$ -clauses to the  $r$ -clauses specified by  $S$  is  $Q_\sigma^\delta$ . Note that by the definition of the distribution complexity,  $\hat{Q}(T) = 0$  for any  $1 \leq |T| < r$ , and so the original and induced distributions are uniform over any set of  $r - 1$  coordinates. □

First consider the case  $r = 1$ . Restricting each clause to  $S$  for  $|S| = 1$ , induces a noisy 1-XOR-SAT distribution in which a random true literal appears with probability  $\delta$  and random false literal appears with probability  $2 - \delta$ . The simple majority vote algorithm described above suffices: set each variable to  $+1$  if it appears more often positively than negated in the restricted clauses of the formula; to  $-1$  if it appears more often negated; and choose randomly if it appears equally often. Using  $c\sqrt{t \log(1/\epsilon)}$  clauses for  $c = O(1/|1 - \delta|^2)$  this algorithm will give an assignment that agrees with  $\sigma$  (or  $-\sigma$ ) on  $n/2 + t\sqrt{n}$  variables with probability at least  $1 - \epsilon$ ; using  $cn \log n$  clauses it will recover  $\sigma$  exactly with probability  $1 - o(1)$ .

Now assume that  $r \geq 2$ . We describe how the parity distribution  $Q_\sigma^\delta$  on  $r$ -constraints induces a bipartite block model. Let  $V_1$  be the set of  $2n$  literals of the given variable set, and  $V_2$  the collection of all  $(r - 1)$ -tuples of literals. We have  $n_1 = |V_1| = 2n$  and  $n_2 = |V_2| = \binom{2n}{r-1}$ . We partition each set into two parts as follows:  $A_1 \subset V_1$  is the set of false literals under  $\sigma$ , and  $B_1$  the set of true literals.  $A_2 \subset V_2$  is the set of  $(r - 1)$ -tuples with an even number of true literals under  $\sigma$ , and  $B_2$  the set of  $(r - 1)$ -tuples with an odd number of true literals.

For each  $r$ -constraint  $(l_1, l_2, \dots, l_r)$ , we add an edge in the block model between the tuples  $l_1 \in V_1$  and  $(l_2, \dots, l_r) \in V_2$ . A constraint drawn according to  $Q_\sigma^\delta$  induces a random edge between  $A_1$  and  $A_2$  or  $B_1$  and  $B_2$  with probability  $\delta/2$  and between  $A_1$  and  $B_2$  or  $B_1$  and  $A_2$  with probability  $1 - \delta/2$ , exactly the distribution of a single edge in the bipartite block model.

Now the model in Definition 2 is that of  $m$  clauses selected independently with replacement according to a given distribution, while in Definition 1, each edge is present independently with a given probability. To reduce from  $m$  independent edges with replacement to the binomial model, we can fix some  $\epsilon > 0$  (e.g.  $\epsilon = 1/2$ ), draw a Poisson random variable  $Z$  with mean  $(1 - \epsilon)m$ , and select the first  $Z$  of the  $m$  edges (whp  $Z \leq m$ ), discarding any multiple edges. By Poisson thinning, this leaves us with a graph where each edge  $e$  appears independently with probability  $p_e$ , where  $p_e = \Pr[\text{Poisson}((1 - \epsilon)m \cdot q_e) \geq 1]$  where  $q_e$  is the probability of edge  $e$  in the single edge distribution. In particular, if for example  $e$  joins a vertex in  $A_1$  to a vertex in  $A_2$  and  $\epsilon = 1/2$ , then

$q_e = \delta/2 \cdot \frac{2}{n_1 n_2}$  and

$$p_e = 1 - \exp\left(-\frac{\delta m}{2n_1 n_2}\right) = \delta p$$

where  $p = \frac{m}{2n_1 n_2}(1 + o(1))$ .

Recovering the partition  $\mathcal{P}_1 = A_1 \cup B_1$  in this bipartite block model partitions the literals into true and false sets giving  $\sigma$  (up to sign).

The reduction from Goldreich's PRG to the bipartite block model is even simpler. By definition, the value of the predicate is correlated with the parity function of some  $r$  of the  $k$  inputs of the predicate (see for example [14]). Therefore the input can be seen as produced by the noisy  $r$ -XOR predicate on random and uniform  $r$ -tuples of variables. The  $r$ -tuples for which this predicate is equal to 1 give an instance of noisy  $r$ -XOR-SAT. A bipartite block model can now be formed on the set of variables and  $(r - 1)$ -tuples of variables (instead of literals) analogously to the construction above.

The key feature of our bipartite block model algorithm is that it uses  $\tilde{O}(\sqrt{n_1 n_2})$  edges (i.e.  $p = \tilde{O}((n_1 n_2)^{-1/2})$ ), corresponding to  $\tilde{O}(n^{r/2})$  clauses in the planted CSP.

## 5 Comparison with spectral approach

As noted above, many approaches to graph partitioning problems and planted satisfiability problems use eigenvectors or singular vectors. These algorithms are essentially based on the signs of the top eigenvector of the centered adjacency matrix being correlated with the planted vector. This is fairly straightforward to establish when the average degree of the random graph is large enough. However, in the stochastic block model, for example, when the average degree is a constant, vertices of large degree dominate the spectrum and the straightforward spectral approach fails (see [51] for a discussion and references).

In the case of the usual block model,  $n_1 = n_2 = n$ , while our approach has a fast running time, it does not save on the number of edges required as compared to the standard spectral approach: both require  $\Omega(n \log n)$  edges. However, when  $n_2 \gg n_1$ , eg.  $n_1 = \Theta(n), n_2 = \Theta(n^{k-1})$  as in the case of the planted  $k$ -CSP's for odd  $k$ , this is no longer the case.

Consider the general-purpose partitioning algorithm of [55]. Let  $G$  be the matrix of edge probabilities:  $G_{ij}$  is the probability that the edge between vertices  $i$  and  $j$  is present. Let  $G_u, G_v$  denote columns of  $G$  corresponding to vertices  $u, v$ . Let  $\sigma^2$  be an upper bound of the variance of an entry in the adjacency matrix,  $s_m$  the size of the smallest part in the planted partition,  $q$  the number of parts,  $\delta$  the failure probability of the algorithm, and  $c$  a universal constant. Then the condition for the success of McSherry's partitioning algorithm is:

$$\min_{u,v \text{ in different parts}} \|G_u - G_v\|^2 > cq\sigma^2(n/s_m + \log(n/\delta))$$

Similar conditions appear in [19, 64]. In our case, we have  $q = 4$ ,  $n = n_1 + n_2$ ,  $s_m = n_1/2$ ,  $\sigma^2 = \Theta(p)$ , and  $\|G_u - G_v\|^2 = 4(\delta - 1)^2 p^2 n_2$ . When  $n_2 \gg n_1 \log n$ , the condition requires  $p = \Omega(1/n_1)$ , while our algorithm succeeds when  $p = \Omega(\log n_1 / \sqrt{n_1 n_2})$ . In our application to planted CSP's with odd  $k$  and  $n_1 = 2n, n_2 = \binom{2n}{k-1}$ , this gives a polynomial factor improvement.

In fact, previous spectral approaches to planted CSP's or random  $k$ -SAT refutation worked for even  $k$  using  $n^{k/2}$  constraints [40, 22, 32], while algorithms for odd  $k$  only worked for  $k = 3$  and used considerably more complicated constructions and techniques [31, 39, 20]. In contrast to previous approaches, our algorithm unifies the algorithm for planted  $k$ -CSP's for odd and even  $k$ , works for odd  $k > 3$ , and is particularly simple and fast.

We now describe why previous approaches faced a spectral barrier for odd  $k$ , and how our algorithm surmounts it.

The previous spectral algorithms for even  $k$  constructed a similar graph to the one in the reduction above: vertices are  $k/2$ -tuples of literals, and with edges between two tuples if their union appears as a  $k$ -clause. The distribution induced in this case is the stochastic block model. For odd  $k$ , such a reduction is not possible, and one might try a bipartite graph, with either the reduction described above, or with  $\lfloor k/2 \rfloor$ -tuples and  $\lceil k/2 \rceil$ -tuples (our analysis works for this reduction as well). However, with  $\tilde{O}(k/2)$  clauses, the spectral approach of computing the largest or second largest singular vector of the adjacency matrix does not work.

Consider  $M$  from the distribution  $M(p)$ . Let  $u$  be the  $n_1$  dimensional vector indexed as the rows of  $M$  whose entries are 1 if the corresponding vertex is in  $A_1$  and  $-1$  otherwise. Define the  $n_2$  dimensional vector  $v$  analogously. The next propositions summarize properties of  $M$ .

**Proposition 1.**  $\mathbb{E}(M) = (\delta - 1)puv^T$ .

**Proposition 2.** Let  $M_1$  be the rank-1 approximation of  $M$  drawn from  $M(p)$ . Then  $\|M_1 - \mathbb{E}(M)\| \leq 2\|M - \mathbb{E}(M)\|$ .

*Proof.* Using the triangle inequality and then the optimality of  $M_1$ ,  $\|M_1 - \mathbb{E}(M)\| \leq \|M - \mathbb{E}(M)\| + \|M - M_1\| \leq 2\|M - \mathbb{E}(M)\|$ .  $\square$

The above propositions suffice to show high correlation between the top singular vector and the vector  $u$  when  $n_2 = \Theta(n_1)$  and  $p = \Omega(\log n_1/n_1)$ . This is because the norm of  $\mathbb{E}(M)$  is  $p\sqrt{n_1n_2}$ ; this is higher than  $O(\sqrt{pn_2})$ , the norm of  $M - \mathbb{E}(M)$  for this range of  $p$ . Therefore the top singular vector of  $M$  will be correlated with the top singular vector of  $\mathbb{E}(M)$ . The latter is a rank-1 matrix with  $u$  as its left singular vector.

However, when  $n_2 \gg n_1$  (eg.  $k$  odd) and  $p = \tilde{O}((n_1n_2)^{-1/2})$ , the norm of the zero-mean matrix  $M - \mathbb{E}(M)$  is in fact much larger than the norm of  $\mathbb{E}(M)$ . Letting  $x^{(i)}$  be the vector of length  $n_1$  with a 1 in the  $i$ th coordinate and zeroes elsewhere, we see that  $\|Mx^{(i)}\|_2 \approx \sqrt{pn_2}$ , and so  $\|M - \mathbb{E}(M)\| = \Omega(\sqrt{pn_2})$ , while  $\|\mathbb{E}(M)\| = O(p\sqrt{n_1n_2})$ ; the former is  $\Omega((n_2/n_1)^{1/4})$  while the latter is  $O(1)$ . In other words, the top singular value of  $M$  is much larger than the value obtained by the vector corresponding to the planted assignment! The picture is in fact richer: the straightforward spectral approach succeeds for  $p \gg n_1^{-2/3}n_2^{-1/3}$ , while for  $p \ll n_1^{-2/3}n_2^{-1/3}$ , the top left singular vector of the centered adjacency matrix is asymptotically uncorrelated with the planted vector [37]. In spite of this, one can exploit correlations to recover the planted vector below this threshold with our resampling algorithm, which in this case provably outperforms the spectral algorithm.

## 6 Analysis of the subsampled power iteration algorithm

We abuse notation and let  $A_1, B_1, A_2, B_2$  denote the sets of coordinates of the corresponding vertex sets. Recall that  $u \in \{\pm 1\}^{n_1}$  is 1 on  $A_1$  and  $-1$  on  $B_1$ , and  $v \in \{\pm 1\}^{n_2}$  is 1 on  $A_2$ ,  $-1$  on  $B_2$ . Set  $T = 10 \log n_2$ ,  $p = \frac{100T}{(\delta-1)^2\sqrt{n_1n_2}}$  and  $q = p/T$ . For convenience we denote  $d = 100/(\delta-1)^2$ . We assume WLOG that  $\delta > 1$ .

Recall that the sequence of matrices  $M_1, \dots, M_T$  is formed by taking  $G_p$  and randomly assigning each edge to one of  $T$  different bipartite graphs, then forming the corresponding centered adjacency matrices. The marginal distribution of each  $M_i$  is a random  $n_1 \times n_2$  matrix with independent entries such that the entry  $(i, j)$  takes value  $1 - q$  with probability  $\delta q$ ,  $-q$  otherwise if  $i \in A_1, j \in A_2$  or  $i \in B_1, j \in B_2$ , and value  $1 - q$  with probability  $(2 - \delta)q$ ,  $-q$  otherwise if  $i \in A_1, j \in B_2$  or  $i \in B_1, j \in A_2$ .

The matrices are not independent, but are nearly independent. Consider the distribution of  $M_i$  conditioned on the matrices  $M_1, \dots, M_{i-1}$ , call this set of edges. Let  $E_{i-1}$  be the set of all edges from  $G_p$  that are assigned to one of  $M_1, M_2, \dots, M_{i-1}$ . Conditioned on  $M_1, \dots, M_{i-1}$ , the entries of  $M_i$  are independent.  $M_i$  is necessarily  $-q$  in every entry  $(u, v)$  with  $(u, v) \in E_{i-1}$ . All other entries take the values  $1 - q$  with probabilities

$$\begin{aligned} \rho_i &:= \frac{\Pr[(u, v) \in G_i \wedge (u, v) \notin G_1, \dots, G_{i-1}]}{\Pr[(u, v) \notin G_1, \dots, G_{i-1}]} \\ &= \frac{\delta q}{1 - \delta p^{\frac{i-1}{T}}} = \delta q + O(pq) \quad \text{if } u \in A_1, v \in A_2 \text{ or } u \in B_1, v \in B_2 \end{aligned}$$

and

$$\begin{aligned} \bar{\rho}_i &:= \frac{\Pr[(u, v) \in G_i \wedge (u, v) \notin G_1, \dots, G_{i-1}]}{\Pr[(u, v) \notin G_1, \dots, G_{i-1}]} \\ &= \frac{(2 - \delta)q}{1 - (2 - \delta)p^{\frac{i-1}{T}}} = (2 - \delta)q + O(pq) \quad \text{if } u \in A_1, v \in B_2 \text{ or } u \in A_2, v \in B_1 \end{aligned}$$

and the value  $-q$  otherwise. The deviation from the fully independent setting is the  $O(pq)$  term.

Let  $H_{i-1}$  be the event that (1)  $|E_{i-1}| \leq 2n_1n_2p$ , (2) each vertex of  $V_1$  appears in  $E_{i-1}$  at most  $3n_2p$  times, and (3) each vertex of  $V_2$  appears in  $E_{i-1}$  at most  $3T(\delta - 1)^{-2}$  times.  $H_{i-1}$  holds for all  $1 \leq i \leq T$  whp from simple Chernoff bounds. We will condition on the set  $E_{i-1}$  and the event  $H_{i-1}$ , to calculate the effect of multiplying a unit vector by  $M_i$  or  $M_i^T$ . The calculations are based on bounding two deviations from the simpler calculations involving the marginal distribution of  $M_i$ : the deviations from the probabilities  $\rho_i$  and  $\bar{\rho}_i$  differing from  $\delta q$  and  $(2 - \delta)q$ , and the deviations from the entries that are fixed to  $-q$ . We write  $g(n) = f(n) + O(h(n))$  to denote two-sided error, i.e.  $f(n) - Ch(n) \leq g(n) \leq f(n) + Ch(n)$ .

**Lemma 2.** *Let  $x$  and  $y$  be unit vectors of dimension  $n_1$  and  $n_2$  respectively. Then*

1.  $\mathbb{E}[u \cdot (M_i y) | E_{i-1}, H_{i-1}] = (\delta - 1)n_1q(v \cdot y)(1 + o(1)) + O(n_1n_2p^2\|y\|_\infty)$
2.  $\text{var}[u \cdot (M_i y) | E_{i-1}, H_{i-1}] = n_1q(1 + o(1)) + O(n_1n_2p^2\|y\|_\infty^2)$ .
3.  $\mathbb{E}[v \cdot (M_i^T x) | E_{i-1}, H_{i-1}] = (\delta - 1)n_2q(u \cdot x)(1 + o(1)) + O(n_2p\|x\|_\infty)$
4.  $\text{var}[v \cdot (M_i^T x) | E_{i-1}, H_{i-1}] = n_2q(1 + o(1)) + O(n_2p^2\|x\|_\infty^2)$ .
5.  $\mathbb{E}[\|M_i y\|_2^2 | E_{i-1}, H_{i-1}] = n_1q(1 + o(1)) + (\delta - 1)^2n_1q^2(v \cdot y)^2(1 + o(1)) + O(n_1n_2^2p^4\|y\|_\infty^2)$ .
6.  $\mathbb{E}[\|M_i^T x\|_2^2 | E_{i-1}, H_{i-1}] = n_2q(1 + o(1)) + (\delta - 1)^2n_2q^2(u \cdot x)^2(1 + o(1)) + O(n_2p^2\|x\|_\infty^2)$ .
7.  $\text{var}[\|M_i y\|_2^2 | E_{i-1}, H_{i-1}] = O(n_1q(\|y\|_1 \cdot \|y\|_\infty^3 + q^3\|y\|_1^4 + q\|y\|_\infty^2 \cdot \|y\|_1^2 + q + q^2\|y\|_1^2))$ .
8.  $\text{var}[\|M_i^T x\|_2^2 | E_{i-1}, H_{i-1}] = O(n_2n_1q\|x\|_\infty^4)$ .

*Proof.* If  $j \in A_1$ ,

$$\begin{aligned} \mathbb{E}[(M_i y)_j | E_{i-1}, H_{i-1}] &= - \sum_{l=1}^{n_2} qy_l + \sum_{\substack{l \in A_2 \\ (j,l) \notin E_{i-1}}} \rho_i y_l + \sum_{\substack{l \in B_2 \\ (j,l) \notin E_{i-1}}} \bar{\rho}_i y_l \\ &= (\delta - 1)q(v \cdot y)(1 + O(p)) + O(n_2pq\|y\|_\infty) \\ \text{var}(M_i y)_j &= \sum_{\substack{l \in A_2 \\ (j,l) \notin E_{i-1}}} \rho_i(1 - \rho_i)y_l^2 + \sum_{\substack{l \in B_2 \\ (j,l) \notin E_{i-1}}} \bar{\rho}_i(1 - \bar{\rho}_i)y_l^2 \end{aligned}$$

and similarly for  $j \in B_1$ .

This gives

$$\begin{aligned}\mathbb{E}[u \cdot (M_i y) | E_{i-1}, H_{i-1}] &= (\delta - 1)n_1 q(v \cdot y)(1 + O(p)) + O(n_1 n_2 p^2 \|y\|_\infty) \\ \text{var}[u \cdot (M_i y) | E_{i-1}, H_{i-1}] &= n_1 q \|y\|_2^2 + O(n_1 q^2 \|y\|_2^2 + n_1 n_2 p^2 \|y\|_\infty^2) \\ &= n_1 q \|y\|_2^2 (1 + O(p)) + O(n_1 n_2 p^2 \|y\|_\infty^2)\end{aligned}$$

Then if  $j \in A_2$ ,

$$\begin{aligned}\mathbb{E}[(M_i^T x)_j | E_{i-1}, H_{i-1}] &= -\sum_{l=1}^{n_1} q x_l + \sum_{\substack{l \in A_1 \\ (l,j) \notin E_{i-1}}} \rho_i x_l + \sum_{\substack{l \in B_1 \\ (l,j) \notin E_{i-1}}} \bar{\rho}_i x_l \\ &= (\delta - 1)q(u \cdot x)(1 + O(p)) + O(p \|x\|_\infty) \\ \text{var}[(M_i^T x)_j | E_{i-1}, H_{i-1}] &= \sum_{\substack{l \in A_1 \\ (l,j) \notin E_{i-1}}} \rho_i (1 - \rho_i) x_l^2 + \sum_{\substack{l \in B_1 \\ (l,j) \notin E_{i-1}}} \bar{\rho}_i (1 - \bar{\rho}_i) x_l^2\end{aligned}$$

and similarly for  $j \in B_2$ .

This gives

$$\begin{aligned}\mathbb{E}[v \cdot (M_i^T x) | E_{i-1}, H_{i-1}] &= (\delta - 1)n_2 q(u \cdot x)(1 + O(p)) + O(n_2 p \|x\|_\infty) \\ \text{var}[v \cdot (M_i^T x) | E_{i-1}, H_{i-1}] &= n_2 q \|x\|_2^2 (1 + O(p)) + O(n_2 p^2 \|x\|_\infty^2)\end{aligned}$$

Finally we have

$$\mathbb{E}[\|M_i y\|_2^2 | E_{i-1}, H_{i-1}] = n_1 q \|y\|_2^2 (1 + O(p)) + (\delta - 1)^2 q^2 n_1 (v \cdot y)^2 (1 + O(p)) + O(n_1 n_2^2 p^4 \|y\|_\infty^2)$$

and

$$\begin{aligned}\text{var}[\|M_i y\|_2^2 | E_{i-1}, H_{i-1}] &= \sum_{i=1}^{n_1} \text{var}((M y)_i)^2 \\ &\leq \sum_{i=1}^{n_1} \mathbb{E}((M y)_i)^4 \\ &= O(n_1 (\|y\|_\infty^3 \|y\|_1 q + q^4 \|y\|_1^4 + q^2 \|y\|_2^4 + q^3 \|y\|_1^2 \|y\|_2^2 + q^2 \|y\|_\infty^2 \|y\|_1^2)) \\ &= O(n_1 q (\|y\|_\infty^3 \|y\|_1 + q^3 \|y\|_1^4 + q + q^2 \|y\|_1^2 + q \|y\|_\infty^2 \|y\|_1^2))\end{aligned}$$

and

$$\mathbb{E}[\|M_i^T x\|_2^2 | E_{i-1}, H_{i-1}] = n_2 q \|x\|_2^2 (1 + O(p)) + (\delta - 1)^2 q^2 n_2 (u \cdot x)^2 (1 + O(p)) + O(n_2 p^2 \|x\|_\infty^2)$$

and

$$\begin{aligned}\text{var}[\|M_i^T x\|_2^2 | E_{i-1}, H_{i-1}] &= \sum_{i=1}^{n_2} \text{var}((M^T x)_i)^2 \\ &\leq \sum_{i=1}^{n_2} \mathbb{E}((M^T x)_i)^4 \\ &= n_2 \|x\|_\infty^4 \cdot O(n_1 q + n_1^2 q^2 + n_1^3 q^3 + n_1^4 q^4) \\ &= O(n_2 n_1 q \|x\|_\infty^4)\end{aligned}$$

□



For cleaner notation in the rest of the proof we will write simply  $\mathbb{E}[\cdot]$  for  $\mathbb{E}[\cdot|E_{i-1}, H_{i-1}]$  when working with the matrix  $M_i$ .

Next we show the normalizing factors  $\|M_i y\|_2$  and  $\|M_i^T x\|_2$  are concentrated at each step; the  $l_\infty$  norms of the  $x^i$ 's are bounded over all iterations, and the  $l_\infty$  and  $l_1$  norms of the  $y^i$ 's are bounded. This proposition is critical in ensuring steady progress of our potential functions.

**Lemma 3.** *With probability  $1 - O(Tn_1^{-1/6})$ , for all  $i = 1, \dots, T$ ,*

1.  $\|M_i y^i\|_2^2 = (n_1 q \|y^i\|_2^2 + (\delta - 1)^2 n_1 q^2 (v \cdot y^i)^2)(1 + o(1))$
2.  $\|M_i^T x^i\|_2^2 = (n_2 q \|x^i\|_2^2 + (\delta - 1)^2 n_2 q^2 (u \cdot x^i)^2)(1 + o(1))$
3.  $\|x^i\|_\infty \leq n_1^{-1/3}$
4.  $\|y^i\|_\infty \leq n_2^{-1/4} n_1^{-1/12}$
5.  $\|y^i\|_1 \leq 4\sqrt{n_2 n_1 q}$

*Proof.* We begin by showing that

$$\left| \left\{ j : |y_j^i| > \sqrt{\frac{2qn_1}{n_2}} \right\} \right| \leq 3n_2 n_1 q. \quad (1)$$

We bound the number  $L$  of  $(1-q)$  entries in  $M_{i-1}$ .  $L$  is stochastically bounded by a  $\text{Binom}(n_2 n_1, 2q)$  random variable, and so,

$$\Pr[L \geq 3n_2 n_1 q] \leq e^{-qn_2 n_1} = e^{-\Theta(\sqrt{n_2 n_1})}.$$

The remaining entries have value  $-q$ . If the  $j$ th row of  $M_{i-1}$  has only  $-q$  entries, then

$$\begin{aligned} |y_j^i| &\leq \frac{q \|x^{i-1}\|_1}{\sqrt{n_2 q / 2}} \\ &\leq \sqrt{\frac{2qn_1}{n_2}} \end{aligned}$$

using (2) inductively. This proves (1).

To prove (5), partition the coordinates of  $y^i$  into two sets  $\Delta$  and  $\bar{\Delta}$ , with  $\Delta$  corresponding to rows of  $M_{i-1}$  with every entry  $-q$ , and  $\bar{\Delta}$  the rest. Then

$$\begin{aligned} \|y^i\|_1 &\leq \sum_{j \in \Delta} |y_j^i| + \sum_{j \in \bar{\Delta}} |y_j^i| \\ &\leq \sqrt{\frac{2qn_1}{n_2}} |\Delta| + \sqrt{|\bar{\Delta}|} \quad \text{using part (2) inductively} \\ &\leq \sqrt{2n_2 n_1 q} + \sqrt{3n_2 n_1 q} \\ &\leq 4\sqrt{qn_2 n_1} \end{aligned}$$

We show by induction that whp the following hold for  $i = 1, \dots, T$ :

1.  $\|M_i y^i\|_2^2 = \mathbb{E}(\|M_i y^i\|_2^2) (1 + O(n_1^{-1/8}))$
2.  $\|M_i^T x^i\|_2^2 = \mathbb{E}(\|M_i^T x^i\|_2^2) (1 + O(n_1^{-1/12}))$
3.  $\|x^i\|_\infty \leq n_1^{-1/3}$
4.  $\|y^i\|_\infty \leq n_2^{-1/4} n_1^{-1/12}$

Conditional on  $y^i$  and  $x^i$  respectively, we have

$$\begin{aligned}\mathbb{E}[\|M_i y^i\|_2^2] &= n_1 q + (\delta - 1)^2 n_1 q^2 (v \cdot y^i)^2 + O(n_1 q^2) \\ \mathbb{E}[\|M_i^T x^i\|_2^2] &= n_2 q + (\delta - 1)^2 n_2 q^2 (u \cdot x^i)^2 + O(n_2 q^2)\end{aligned}$$

Using Chebyshev and part (3),

$$\begin{aligned}\Pr\left[\left|\|M_i y^i\|_2^2 - \mathbb{E}(\|M_i y^i\|_2^2)\right| > \alpha \mathbb{E}(\|M_i y^i\|_2^2)\right] \\ \leq \frac{\text{var}(\|M_i y^i\|_2^2)}{n_1^2 q^2 \alpha^2} \\ = \alpha^{-2} \cdot O\left(\frac{\|y\|_1 \cdot \|y\|_\infty^3 + q^3 \|y\|_1^4 + q \|y\|_\infty^2 \cdot \|y\|_1^2 + q + q^2 \|y\|_1^2}{n_1 q}\right) \\ = \alpha^{-2} \cdot O\left(\frac{q^{1/2} n_2^{-1/4} n_1^{1/4} + q^5 n_2^2 n_1^2 + q^2 n_2^{1/2} n_1^{5/6} + q + q^3 n_2 n_1}{n_1 q}\right) \\ = \alpha^{-2} \cdot O\left(q^{-1/2} n_2^{-1/4} n_1^{-3/4} + q^4 n_2^2 n_1 + q n_2^{1/2} n_1^{-1/6} + n_1^{-1} + q^2 n_2\right) \\ = \alpha^{-2} \cdot O\left(n_1^{-1/2} + n_1^{-1} + n_1^{-2/3} + n_1^{-1} + n_1^{-1}\right) \\ = O\left(\frac{1}{n_1^{1/2} \alpha^2}\right) \\ = O\left(n_1^{-1/4}\right) \quad \text{for } \alpha = n_1^{-1/8}.\end{aligned}$$

Similarly, using Chebyshev and part (4),

$$\begin{aligned}\Pr\left[\left|\|M_i^T x^i\|_2^2 - \mathbb{E}(\|M_i^T x^i\|_2^2)\right| > \alpha \mathbb{E}(\|M_i^T x^i\|_2^2)\right] \\ \leq \frac{\text{var}(\|M_i^T x^i\|_2^2)}{n_2^2 q^2 \alpha^2} \\ = O\left(\frac{n_1 \|x^i\|_\infty^4}{n_2 q \alpha^2}\right) \\ = O\left(\frac{1}{n_2 n_1^{1/3} q \alpha^2}\right) \\ = O\left(\frac{n_1^{1/6}}{n_2^{1/2} \alpha^2}\right) \\ = O\left(n_1^{-1/6}\right) \quad \text{for } \alpha = n_1^{-1/12}.\end{aligned}$$

To prove (3), note that

$$\|x^{i+1}\|_\infty = \max_{j \in [n_1]} \frac{|(M_i^T y^i)_j|}{\|M_i^T y^i\|_2}$$

Using part (1),  $\|M_i^T y^i\|_2 \geq \sqrt{n_1 q}/2$  with probability  $1 - O(n_1^{-1/4})$ . Therefore it suffices to show that for every  $j = 1, \dots, n_1$ ,

$$|(M_i^T y^i)_j| \leq \frac{n_1^{-1/3} \sqrt{n_1 q}}{2} = \frac{\sqrt{d} n_1^{1/6} n_2^{-1/4}}{2}.$$

To this end we will show that for any  $j$ ,

$$\Pr \left[ |(M_i^T y^i)_j| > \frac{\sqrt{d} n_1^{1/6} n_2^{-1/4}}{2} \right] \leq \frac{1}{n_1^2} \quad (2)$$

Again partition the coordinates of  $y^i$ , with  $\Delta$  being the set of  $j$  so that  $|y_j^i| \leq \sqrt{\frac{2qn_1}{n_2}}$  and  $\bar{\Delta}$  the rest. The contribution to  $|(M_i^T y^i)_j|$  from  $\Delta$  is bounded by

$$(n_2 q + m_j) \sqrt{\frac{2qn_1}{n_2}}$$

where  $m_j$  is the number of  $1 - q$  entries in the  $j$ th row of  $M_i^T$ . This number  $m_j$  is dominated by a  $\text{Binom}(n_2, 2q)$  random variable and so with probability  $1 - \exp(-n_2 q)$ ,  $m_j \leq 3n_2 q$ . Therefore, the contribution from  $\Delta$  is bounded by

$$\begin{aligned} (n_2 q + 3n_2 q) \sqrt{\frac{2qn_1}{n_2}} &\leq 5\sqrt{n_2 n_1 q^3} \\ &= O\left((n_2 n_1)^{-1/4}\right) = o(n_1^{1/6} n_2^{-1/4}). \end{aligned}$$

The contribution to  $|(M_i^T y^i)_j|$  from  $\bar{\Delta}$  is bounded by

$$(3n_2 n_1 q \cdot q + \bar{m}_j \cdot 1) n_2^{-1/4} n_1^{-1/12}$$

where we have used (4) and (1)), and  $\bar{m}_j$  is the number of  $1 - q$  entries in the  $j$ th row of  $M_i^T$  whose column has index in  $\bar{\Delta}$ .  $\bar{m}_j$  is dominated by a  $\text{Binom}(3n_2 n_1 q, q)$  random variable, and so with probability  $1 - O(\exp(-\Omega(3n_2 n_1 13/12q^2)))$ ,  $\bar{m}_j \leq 3n_2 n_1 q^2 \cdot n_1^{1/12}$  in which case we have that the contribution from  $\bar{\Delta}$  is bounded by

$$\begin{aligned} &3n_2 n_1 q^2 n_2^{-1/4} n_1^{-1/12} + 3n_2 n_1 q^2 n_2^{-1/4} \\ &= 3d^2 n_2^{-1/4} n_1^{-1/12} + 3d^2 n_2^{-1/4} \\ &\leq \frac{\sqrt{d} n_1^{1/6} n_2^{-1/4}}{4} \end{aligned}$$

proving inequality (2). (We remark that for this part, the loose bounds we have above suffice; it is the next part that controls parameter settings).

To prove (4), set  $\lambda = n_2^{-1/4} n_1^{-1/12}$ .

$$\|y^{i+1}\|_\infty = \max_{j \in [n_2]} \frac{|(M_i^T x^i)_j|}{\|M_i^T x^i\|_2}$$

Using part (2),  $\|M_i^T x^i\|_2 \geq \sqrt{n_2 q}/2$  with probability  $1 - O(n_1^{-1/6})$ . Therefore it suffices to show that for every  $j = 1, \dots, n_2$ ,

$$\begin{aligned} |(M_i^T x^i)_j| &\leq \frac{\lambda \sqrt{n_2 q}}{2} \\ &= \frac{\lambda \sqrt{d} n_2^{1/4}}{2n_1^{1/4}} \end{aligned}$$

We will show that for any  $j$ ,

$$\Pr \left[ |(M_i^T x^i)_j| > \frac{\lambda \sqrt{d} n_2^{1/4}}{2n_1^{1/4}} \right] \leq \frac{1}{n_2^2} \quad (3)$$

We partition the coordinates of  $x^i$  according to their magnitude, in bins  $B_1, \dots, B_L$ , defined for  $l < L$  as

$$B_l = \left\{ i : |x_i| \in \left( \frac{n_1^{-1/3}}{2^l}, \frac{n_1^{-1/3}}{2^{l-1}} \right] \right\}$$

with the interval for  $B_L$  being  $[0, n_1^{-1/3}/2^{L-1}]$ . We set  $L = \lceil \log(n_1^{1/6}) \rceil$ . Let

$$t_l = |B_l| \leq 2^{2l} n_1^{2/3}$$

using the fact that  $x^i$  has unit 2-norm.

We will bound the probability that bin  $l$  contributes more than  $\beta_l$  towards the value of  $|(M_i^T x^i)_j|$ , with

$$\beta_l = \frac{\lambda \sqrt{d} n_2^{1/4}}{4n_1^{1/4} l^2}$$

If all bins fall within these bounds, then

$$|(M_i^T x^i)_j| \leq \sum_l \beta_l \leq \frac{\lambda \sqrt{d} n_2^{1/4}}{2n_1^{1/4}}$$

and therefore  $\|y^{i+1}\|_\infty \leq n_2^{-1/4} n_1^{-1/12}$ .

Let  $Z_l \sim \text{Binom}(t_l, q)$ . The contribution of bin  $l$  is bounded by the maximum of  $\frac{n_1^{-1/3}}{2^{l-1}} Z_l$  and  $|q \sum_r x_r^i| \leq q \sqrt{n_1} \leq \beta_l$ . To bound the first term, let

$$\begin{aligned} m_l &= \beta_l 2^{l-1} n_1^{1/3} \\ &= \frac{\lambda 2^l \sqrt{d} n_2^{1/4} n_1^{1/12}}{8l^2} \\ &= \frac{2^l \sqrt{d}}{8l^2} \end{aligned}$$

and consider

$$\begin{aligned}
\Pr [Z_l \geq m_l] &\leq 2 \binom{t_l}{m_l} q^{m_l} \\
&\leq 2 \left( \frac{et_l q}{m_l} \right)^{m_l} \\
&\leq 2 \left( \frac{e2^{2l} n_1^{2/3} d(n_2 n_1)^{-1/2}}{\sqrt{d} 2^l / (8l^2)} \right)^{\sqrt{d} 2^l / (8l^2)} \\
&= 2 \left( \frac{8el^2 2^l \sqrt{d} n_1^{1/6}}{n_2^{1/2}} \right)^{\sqrt{d} 2^l / (8l^2)} \\
&\leq 2 \left( \frac{8e\sqrt{d} \log^2(n_1) n_1^{1/3}}{n_2^{1/2}} \right)^{\sqrt{d} 2^l / (8l^2)} \\
&\leq 2 \left( \frac{8e\sqrt{d} \log^2(n_1)}{n_2^{1/6}} \right)^{\sqrt{d}/4} \\
&\leq \frac{(8e\sqrt{d} \log^2(n_1))^7}{n_2^{7/3}} \quad \text{for } \sqrt{d}/4 \geq 7.
\end{aligned}$$

Taking a union bound over all  $L$  bins, we have (3). □

Next we show that the vector  $y^i$  reaches high correlation with  $v$  after  $T/2$  steps. Recall the definitions  $V_i := v \cdot y^i$  and  $U_i := u \cdot x^i$ .

**Proposition 3.** *With probability  $1 - O((\ln n_1)^{-2})$ , one of the following happens:*

1. For all  $i \in \{T/2, \dots, T\}$ ,

$$V_i \geq \frac{(\delta - 1)\sqrt{n_2 n_1 q}}{4}$$

2. For all  $l \in \{T/2, \dots, T\}$ ,

$$V_i \leq -\frac{(\delta - 1)\sqrt{n_2 n_1 q}}{4}$$

First we need the following bounds on the progress of  $U_i$ :

**Proposition 4.** *The following bounds on  $U_i$  hold:*

1. With probability at least  $1/2$ ,  $|U_i| \geq 1/4$  regardless of the value of  $V_i$ .
2. If  $1/4 \leq |U_i| \leq \ln n_1$ , then with probability at least  $1/2$ ,  $|U_{i+1}| \geq 2|U_i|$ .
3.  $\Pr[|U_{i+1}| \geq 2|U_i|] \geq 1 - \frac{1}{|U_i|^2}$  for  $\ln n_1 \leq |U_i| \leq \sqrt{n_1}/4$ .
4. If  $U_i \geq \sqrt{n_1}/4$ , then  $\Pr[U_{i+1} \geq \sqrt{n_1}/2] \geq 1 - O(1/\sqrt{n_1 n_2})$ . Similarly, if  $U_i \leq -\sqrt{n_1}/4$ , then  $\Pr[U_{i+1} \leq -\sqrt{n_1}/2] \geq 1 - O(1/\sqrt{n_1 n_2})$ .
5. If  $U_i \geq \sqrt{n_1}/2$ , then  $V_{i+1} \geq \frac{(\delta-1)\sqrt{n_2 n_1 q}}{4}$  with probability  $1 - O(1/\sqrt{n_1 n_2})$ .

1) and 2) ensure that Phase 1 succeeds, and that  $U_i$  attains value  $\ln n_1$  within  $\ln n_1$  steps. 3) and 4) ensure steady progress in Phase 2 and that once  $U_i$  attains a high value, it maintains it. 5) connects the two potential functions by showing that  $V_{i+1}$  is large if  $U_i$  is large.

*Proof of Proposition 4.* 1. The variance of  $u \cdot (M_i y^i)$  is  $\sim n_1 q$ , and so a Berry-Esseen bound gives that with probability at least  $1/2$ ,  $|u \cdot M_i y^i| \geq \sqrt{n_1 q}/4$ . Then using Lemma 3, we have that  $\|M_i y^i\|_2 = \sqrt{n_1 q}(1 + o(1))$  whp, and so with probability at least  $1/2$ ,  $|U_i| = |u \cdot x^i| \geq 1/4$ .

2. We prove this in two steps. The expectation of  $v \cdot (M_i^T x^i)$  is  $(\delta - 1)n_2 q(u \cdot x^i)$ , with variance  $n_2 q$ . Both are  $\omega(1)$ , and the expectation is at least  $(\delta - 1)/4$  times the variance in absolute value, and so whp,  $v \cdot (M_i^T x^i) = (\delta - 1)n_2 q(u \cdot x^i)(1 + o(1))$ . Using Lemma 3 again, we have that whp,  $V_{i+1} = (\delta - 1)\sqrt{n_2 q}(u \cdot x^i)$ .

Conditioning on this value, we have

$$\mathbb{E}[u \cdot (M_{i+1} y^{i+1})] = (\delta - 1)^2 \sqrt{n_2 q} n_1 q (u \cdot x^i) (1 + o(1))$$

and its variance is  $n_1 q$ . With probability  $1/2$  we have  $|u \cdot (M_{i+1} y^{i+1})| \geq (\delta - 1)^2 \sqrt{n_2 q} n_1 q (u \cdot x^i) (1 - o(1))$ , and then normalizing with Lemma 3 we have  $|U_{i+1}| \geq (\delta - 1)^2 \sqrt{n_2 n_1} q |U_i|$ , which from our choice of  $q$ , is at least  $2|U_i|$ .

3. Similar to the above. Apply Chebyshev so that  $v \cdot (M_i^T x^i) = (\delta - 1)n_2 q(u \cdot x^i)(1 + o(1))$  with probability  $1 - o(1)$ , and normalize so that  $v \cdot y^{i+1} = (\delta - 1)\sqrt{n_2 q}(u \cdot x^i)(1 + o(1))$  whp. Now the expectation of  $u \cdot (M_{i+1} y^{i+1})$  is  $(\delta - 1)^2 \sqrt{n_2 q} n_1 q (u \cdot x^i)(1 + o(1))$  with variance  $n_1 q$ , and so applying Chebyshev, we have

$$\begin{aligned} \Pr[|u \cdot (M_{i+1} y^{i+1})| < (\delta - 1)^2 \sqrt{n_2 q} n_1 q |u \cdot x^i|/2] &\leq \frac{n_1 q}{(\delta - 1)^4 n_2 n_1^2 q^3 (u \cdot x^i)^2 / 4} \\ &= \frac{4}{(\delta - 1)^4 n_2 n_1 q^2 (u \cdot x^i)^2} \\ &\leq \frac{1}{25(u \cdot x^i)^2} \end{aligned}$$

Then normalizing, and using Lemma 2 and part (2) above, we get

$$\begin{aligned} |U_{i+1}| &\geq \frac{(\delta - 1)^2 \sqrt{n_2 q} n_1 q |U_i|}{2\sqrt{n_1 q + (\delta - 1)^2 n_1 q^2 (V_{i+1})^2}} \\ &\geq \frac{(\delta - 1)^2 \sqrt{n_2 q} n_1 q |U_i|}{2\sqrt{n_1 q + (\delta - 1)^4 n_2 n_1 q^3 (U_i)^2}} \\ &\geq 2|U_i|. \end{aligned}$$

4.5. Chebyshev again. □

*Proof of Proposition 3.* In the first phase, we show that it takes  $\ln n_1$  iterations for  $|U_i|$  to reach  $\ln n_1$  whp. Next, it takes a further  $\ln n_1$  iterations to reach  $\sqrt{n_1}/2$ . Finally,  $|U_i|$  will remain above  $\sqrt{n_1}/2$  whp for an additional  $2 \ln n_1$  iterations.

Step 1: We call a step from  $U_i$  to  $U_{i+1}$  ‘good’ if  $|U_{i+1}| \geq 2|U_i|$ , or if  $|U_{i+1}| \geq 1/4$  following a ‘bad’ step. A run of  $\ln \ln n_1$  good steps must end with  $|U_i| \geq \ln n_1$ . As long as  $|U_i| < \ln n_1$ , the proposition above shows that the probability of a good step is at least  $1/2$ , so in  $\ln n_1$  steps, with probability  $1 - o(1)$  we will either have such a run of  $\ln \ln n_1$  good steps or reach  $\ln n_1$  even earlier.

Step 2: Once we have  $|U_i| \geq \ln n_1$ , the value will double whp in successive steps until  $|U_i| \geq \sqrt{n_1}/4$ . This takes at most  $\ln n_1$  steps. The total error probability, by part 3) of Proposition 4 is a geometric series that sums to  $O(1/(\ln n_1)^2)$ .

Step 3: Once  $|U_i| \geq \sqrt{n_1}/4$  then for the next  $2 \ln n_1$  steps,  $U_{i+1}, U_{i+2}, \dots$ , we have  $|U_i| \geq \sqrt{n_1}/2$ , with total error probability  $O(T/\sqrt{n_1 n_2})$ .

Step 4: Finally we use part 5) of Proposition 4 to conclude that  $y^i$  has high correlation with  $v$ .  $\square$

We now use Proposition 3 to prove the main theorem.

*Proof of Theorem 1.* Now that we know whp  $y^{T/2}, y^{T/2+1}, \dots$  all have large correlation with  $v$ , we show that taking the majority vote for each coordinate of  $z^{T/2+1}, z^{T/2+2}, \dots$  recovers  $\pm u$  whp.

Take the first case from Proposition 3, with  $V_i \geq \frac{(\delta-1)\sqrt{n_2 n_1 q}}{4}$ . Assume  $j \in A_1$ , then we have, conditioned on the value of  $V_i$

$$\begin{aligned} \Pr[z_j^{i+1} = 1] &= \Pr[x_j > 0] \\ &\geq 1 - \frac{\text{var}((My^i)_j)}{(\mathbb{E}((My^i)_j))^2} \\ &\geq 1 - \frac{32q}{(\delta-1)^4 q^3 n_1 n_2} \\ &= 1 - \frac{32}{100^2} \geq .9 \end{aligned}$$

Now an application of Azuma's inequality shows that with probability at least  $1 - o(n_1^{-2})$ ,  $\sum_{i=T/2}^T z_j^i > 0$ . Similarly, for  $j \in B_1$ , we have  $\sum_{i=T/2}^T z_j^i > 0$  with probability at least  $1 - o(n_1^{-2})$ , and so whp the majority vote recovers  $u$  exactly. The same argument shows that if the second case of Proposition 3 holds, then we find  $-u$  whp.  $\square$

## References

- [1] Emmanuel Abbe, Afonso S Bandeira, and Georgina Hall. Exact recovery in the stochastic block model. *arXiv preprint arXiv:1405.3267*, 2014.
- [2] Dimitris Achlioptas, Haixia Jia, and Cristopher Moore. Hiding satisfying assignments: Two are better than one. *J. Artif. Intell. Res.(JAIR)*, 24:623–639, 2005.
- [3] Dimitris Achlioptas and Frank McSherry. Fast computation of low rank matrix approximations. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 611–618. ACM, 2001.
- [4] Michael Alekhnovich. More on average case vs approximation complexity. *Computational Complexity*, 20(4):755–786, 2011.
- [5] Noga Alon and Nabil Kahale. A spectral technique for coloring random 3-colorable graphs. *SIAM Journal on Computing*, 26(6):1733–1748, 1997.

- [6] Benny Applebaum. Pseudorandom generators with long stretch and low locality from random local one-way functions. In *Proceedings of the 44th symposium on Theory of Computing*, pages 805–816. ACM, 2012.
- [7] Benny Applebaum, Boaz Barak, and Avi Wigderson. Public-key cryptography from different assumptions. In *Proceedings of the 42nd ACM symposium on Theory of computing*, pages 171–180. ACM, 2010.
- [8] Boaz Barak. Truth vs proof: The unique games conjecture and Feige’s hypothesis. <http://windowsontheory.org/2012/07/31/truth-vs-proof-the-unique-games-conjecture-and-feiges-hypothesis/>, July 2012.
- [9] Wolfgang Barthel, Alexander K Hartmann, Michele Leone, Federico Ricci-Tersenghi, Martin Weigt, and Riccardo Zecchina. Hiding solutions in random satisfiability problems: A statistical mechanics approach. *Physical review letters*, 88(18):188701, 2002.
- [10] Quentin Berthet. Optimal testing for planted satisfiability problems. *arXiv preprint arXiv:1401.2205*, 2014.
- [11] Quentin Berthet and Philippe Rigollet. Complexity theoretic lower bounds for sparse principal component detection. In *COLT*, pages 1046–1066, 2013.
- [12] Avrim Blum. Learning boolean functions in an infinite attribute space. *Machine Learning*, 9:373–386, 1992.
- [13] Avrim Blum and Joel Spencer. Coloring random and semi-random k-colorable graphs. *Journal of Algorithms*, 19(2):204–234, 1995.
- [14] Andrej Bogdanov and Youming Qiao. On the security of goldreich’s one-way function. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 392–405. Springer, 2009.
- [15] Ravi B Boppana. Eigenvalues and graph bisection: An average-case analysis. In *Foundations of Computer Science, 1987., 28th Annual Symposium on*, pages 280–285. IEEE, 1987.
- [16] Julia Böttcher. Coloring sparse random k-colorable graphs in polynomial expected time. *Mathematical Foundations of Computer Science 2005*, page 156, 2005.
- [17] Moses Charikar and Anthony Wirth. Maximizing quadratic programs: extending grothendieck’s inequality. In *Foundations of Computer Science, 2004. Proceedings. 45th Annual IEEE Symposium on*, pages 54–60. IEEE, 2004.
- [18] Amin Coja-Oghlan. A spectral heuristic for bisecting random graphs. *Random Structures & Algorithms*, 29:3:351–398, 2006.
- [19] Amin Coja-Oghlan. Graph partitioning via adaptive spectral techniques. *Combinatorics, Probability & Computing*, 19(2):227, 2010.
- [20] Amin Coja-Oghlan, Colin Cooper, and Alan Frieze. An efficient sparse regularity concept. *SIAM Journal on Discrete Mathematics*, 23(4):2000–2034, 2010.
- [21] Amin Coja-Oghlan, Andreas Goerdt, and André Lanka. Strong refutation heuristics for random k-sat. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 310–321. Springer, 2004.



- [22] Amin Coja-Oghlan, Andreas Goerdt, André Lanka, and Frank Schädlich. Certifying unsatisfiability of random  $2k$ -sat formulas using approximation techniques. In *Fundamentals of Computation Theory*, pages 15–26. Springer, 2003.
- [23] Amin Coja-Oghlan, Andreas Goerdt, André Lanka, and Frank Schädlich. Techniques from combinatorial approximation algorithms yield efficient algorithms for random  $2k$ -sat. *Theoretical Computer Science*, 329(1):1–45, 2004.
- [24] Amin Coja-Oghlan and André Lanka. Finding planted partitions in random graphs with general degree distributions. *SIAM Journal on Discrete Mathematics*, 23(4):1682–1714, 2009.
- [25] Anne Condon and Richard M Karp. Algorithms for graph partitioning on the planted partition model. *Random Structures & Algorithms*, 18(2):116–140, 2001.
- [26] Amit Daniely, Nati Linial, and Shai Shalev-Shwartz. More data speeds up training time in learning halfspaces over sparse vectors. In *NIPS*, pages 145–153, 2013.
- [27] Amit Daniely and Shai Shalev-Shwartz. Complexity theoretic limitations on learning dnf’s. *CoRR*, abs/1404.3378, 2014.
- [28] S. Decatur, O. Goldreich, and D. Ron. Computational sample complexity. *SIAM Journal on Computing*, 29(3):854–879, 1999.
- [29] Aurelien Decelle, Florent Krzakala, Cristopher Moore, and Lenka Zdeborová. Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. *Physical Review E*, 84(6):066106, 2011.
- [30] Martin E. Dyer and Alan M. Frieze. The solution of some random np-hard problems in polynomial expected time. *Journal of Algorithms*, 10(4):451–489, 1989.
- [31] Uriel Feige and Eran Ofek. Easily refutable subformulas of large random 3cnf formulas. In *Automata, languages and programming*, pages 519–530. Springer, 2004.
- [32] Uriel Feige and Eran Ofek. Spectral techniques applied to sparse random graphs. *Random Structures & Algorithms*, 27(2):251–275, 2005.
- [33] V. Feldman. Attribute efficient and non-adaptive learning of parities and DNF expressions. *Journal of Machine Learning Research*, (8):1431–1460, 2007.
- [34] Vitaly Feldman, Elena Grigorescu, Lev Reyzin, Santosh Vempala, and Ying Xiao. Statistical algorithms and a lower bound for planted clique. In *Proceedings of the 45th annual ACM symposium on Symposium on theory of computing*, pages 655–664. ACM, 2013.
- [35] Vitaly Feldman, Will Perkins, and Santosh Vempala. On the complexity of random satisfiability problems with planted solutions. *arXiv preprint arXiv:1311.4821*, 2013.
- [36] Abraham Flaxman. A spectral technique for random satisfiable 3cnf formulas. In *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 357–363. Society for Industrial and Applied Mathematics, 2003.
- [37] Laura Florescu and Will Perkins. Spectral thresholds in the bipartite stochastic block model. preprint, 2015.

- [38] Michael L Fredman, János Komlós, and Endre Szemerédi. Storing a sparse table with 0 (1) worst case access time. *Journal of the ACM (JACM)*, 31(3):538–544, 1984.
- [39] Joel Friedman, Andreas Goerdt, and Michael Krivelevich. Recognizing more unsatisfiable random k-sat instances efficiently. *SIAM Journal on Computing*, 35(2):408–430, 2005.
- [40] Andreas Goerdt and Michael Krivelevich. Efficient recognition of random unsatisfiable k-sat instances by spectral methods. In *STACS 2001*, pages 294–304. Springer, 2001.
- [41] Andreas Goerdt and André Lanka. Recognizing more random unsatisfiable 3-sat instances efficiently. *Electronic Notes in Discrete Mathematics*, 16:21–46, 2003.
- [42] Oded Goldreich. Candidate one-way functions based on expander graphs. *IACR Cryptology ePrint Archive*, 2000:63, 2000.
- [43] Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137, 1983.
- [44] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Cryptography with constant computational overhead. In *Proceedings of the 40th annual ACM symposium on Theory of computing*, pages 433–442. ACM, 2008.
- [45] Mark Jerrum and Gregory B Sorkin. The metropolis algorithm for graph bisection. *Discrete Applied Mathematics*, 82(1):155–175, 1998.
- [46] Haixia Jia, Cristopher Moore, and Doug Strain. Generating hard satisfiable formulas by hiding solutions deceptively. In *PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE*, volume 20, page 384. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2005.
- [47] Michael Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM (JACM)*, 45(6):983–1006, 1998.
- [48] Satish Babu Korada, Andrea Montanari, and Sewoong Oh. Gossip pca. In *Proceedings of the ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems*, pages 209–220. ACM, 2011.
- [49] Michael Krivelevich and Dan Vilenchik. Solving random satisfiable 3cnf formulas in expected polynomial time. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 454–463. ACM, 2006.
- [50] Florent Krzakala, Marc Mézard, and Lenka Zdeborová. Reweighted belief propagation and quiet planting for random k-sat. *Journal on Satisfiability, Boolean Modeling and Computation*, 8:149–171, 2014.
- [51] Florent Krzakala, Cristopher Moore, Elchanan Mossel, Joe Neeman, Allan Sly, Lenka Zdeborová, and Pan Zhang. Spectral redemption in clustering sparse networks. *Proceedings of the National Academy of Sciences*, 110(52):20935–20940, 2013.
- [52] Florent Krzakala and Lenka Zdeborová. Hiding quiet solutions in random constraint satisfaction problems. *Physical review letters*, 102(23):238701, 2009.
- [53] Frank Lin and William W Cohen. Power iteration clustering. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 655–662, 2010.

- [54] Laurent Massoulié. Community detection thresholds and the weak ramanujan property. In *STOC 2014: 46th Annual Symposium on the Theory of Computing*, pages 1–10, 2014.
- [55] Frank McSherry. Spectral partitioning of random graphs. In *Foundations of Computer Science, 2001. Proceedings. 42nd IEEE Symposium on*, pages 529–537. IEEE, 2001.
- [56] Elchanan Mossel, Joe Neeman, and Allan Sly. A proof of the block model threshold conjecture. *arXiv preprint arXiv:1311.4115*, 2013.
- [57] Elchanan Mossel, Joe Neeman, and Allan Sly. Consistency thresholds for binary symmetric block models. *arXiv preprint arXiv:1407.1591*, 2014.
- [58] Elchanan Mossel, Joe Neeman, and Allan Sly. Reconstruction and estimation in the planted partition model. *Probability Theory and Related Fields*, pages 1–31, 2014.
- [59] Raj Rao Nadakuditi and Mark EJ Newman. Graph spectra and the detectability of community structure in networks. *Physical review letters*, 108(18):188701, 2012.
- [60] Ryan O’Donnell and David Witmer. Goldreich’s prg: Evidence for near-optimal polynomial stretch. In *Conference on Computational Complexity*, 2014.
- [61] R. Servedio. Computational sample complexity and attribute-efficient learning. *Journal of Computer and System Sciences*, 60(1):161–178, 2000.
- [62] Shai Shalev-Shwartz, Ohad Shamir, and Eran Tromer. Using more data to speed-up training time. In *AISTATS*, pages 1019–1027, 2012.
- [63] Luca Trevisan. More ways to prove unsatisfiability of random k-sat. <http://lucatrevisan.wordpress.com/2007/08/21/more-ways-to-prove-unsatisfiability-of-random-k-sat/>, August 2007.
- [64] Van Vu. A simple svd algorithm for finding hidden partitions. *arXiv preprint arXiv:1404.3918*, 2014.
- [65] Hanson Zhou and David Woodruff. Clustering via matrix powering. In *Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 136–142. ACM, 2004.