On the Complexity of Random Satisfiability Problems with Planted Solutions[*]

Vitaly Feldman[†] Will Perkins[‡] Santosh Vempala[§]

Abstract

The problem of identifying a planted assignment given a random k-SAT formula consistent with the assignment exhibits a large algorithmic gap: while the planted solution becomes unique and can be identified given a formula with $O(n \log n)$ clauses, there are distributions over clauses for which the best known efficient algorithms require $n^{k/2}$ clauses. We propose and study a unified model for planted k-SAT, which captures well-known special cases. An instance is described by a planted assignment σ and a distribution on clauses with k literals. We define its *distribution complexity* as the largest r for which the distribution is not r-wise independent ($1 \leq r \leq k$ for any distribution with a planted assignment).

Our main result is an unconditional lower bound, tight up to logarithmic factors, for *statistical* (query) algorithms [Kea98, FGR$^+$12], matching known upper bounds, which, as we show, can be implemented using a statistical algorithm. Since known approaches for problems over distributions have statistical analogues (spectral, MCMC, gradient-based, convex optimization etc.), this lower bound provides a rigorous explanation of the observed algorithmic gap. The proof introduces a new general technique for the analysis of statistical query algorithms. It also points to a geometric *paring* phenomenon in the space of all planted assignments.

We describe consequences of our lower bounds to Feige's refutation hypothesis [Fei02] and to lower bounds on general convex programs that solve planted k-SAT. Our bounds also extend to other planted k-CSP models, and, in particular, provide concrete evidence for the security of Goldreich's one-way function and the associated pseudorandom generator when used with a sufficiently hard predicate [Gol00].

[*] An extended abstract of this paper appeared at STOC, 2015 [FPV15].

[†] Google Research. Work done while at IBM Research and visiting the Simons Institute, UC Berkeley

[‡] University of Birmingham. Part of the work done while at Georgia Tech. Supported in part by an NSF postdoctoral fellowship.

[§] Georgia Tech. Supported in part by NSF award CCF-1217793.

Contents

1 Introduction

Boolean satisfiability and constraint satisfaction problems are central to complexity theory; they are canonical NP-complete problems and their approximate versions are also hard. Are they easier on average for natural distributions? An instance of random satisfiability is generated by fixing a distribution over clauses, then drawing i.i.d. clauses from this distribution. The average-case complexity of satisfiability problems is also motivated by its applications to models of disorder in physical systems, and to cryptography, which requires problems that are hard on average.

Here we study *planted satisfiability*, in which an assignment is fixed in advance, and clauses selected from a distribution defined by the planted assignment. Planted satisfiability and, more generally, random models with planted solutions appear widely in several different forms such as network clustering with planted partitions (the stochastic block model and its variants), random k-SAT with a planted assignment, and a proposed one-way function from cryptography [Gol00].

It was noted in [BHL$^+$02] that drawing satisfied k-SAT clauses uniformly at random from all those satisfied by an assignment $\sigma \in \{\pm 1\}^n$ often does not result in a difficult instance of satisfiability even if the number of observed clauses is relatively small. However, by changing the proportions of clauses depending on the number of satisfied literals under σ, one can create more challenging distributions over instances. Such "quiet plantings" have been further studied in [JMS05, AJM05, KZ09, KMZ14]. Algorithms for planted 3-SAT with various relative proportions were given by Flaxman [Fla03] and Coja-Oghlan *et al.* [COCF10], the first of which works for $\Theta(n \log n)$ clauses but excludes distributions close to 3-XOR-SAT, and the second of which works for all planted 3-SAT distributions but requires $\Theta(n^{3/2} \ln^{10} n)$ clauses (note that a satisfiable k-XOR-SAT formula can be viewed as a satisfiable k-SAT formula with the same literals since XOR implies OR). As k increases, the problem exhibits a larger algorithmic gap: the number of clauses required by known algorithms to efficiently identify a planted assignment is $\Omega(n^{k/2})$ while the number at which the planted assignment is the unique satisfying assignment is $O(n \log n)$.

We give a simple model for producing instances of planted k-SAT that generalizes and unifies past work on specific distributions for planted satisfiability. In this model, each clause C, a k-tuple of the $2n$ literals (variables and their negations), is included in the random formula with probability proportional to $Q(y)$ where $y \in \{\pm 1\}^k$ is the value of the literals in C on the planted assignment σ. Here Q can be an arbitrary probability distribution over $\{\pm 1\}^k$. By choosing Q supported only on k-bit strings with at least one true value, we can ensure that only satisfiable k-SAT formulas will be produced, but the model is more general and allows "noisy" versions of satisfiability. We refer to an instance obtained by taking Q to be uniform over k-bit strings with an even number of 1's as k-XOR-SAT (since each clause also satisfies an XOR constraint).

We identify the parameter of Q that determines (up to lower order terms) the number of clauses that existing efficient algorithms require. It is the largest r such that the distribution Q is $(r-1)$-wise independent but not r-wise. Equivalently, it is the size of the smallest non-empty subset of k indices for which the discrete Fourier coefficient of Q is nonzero. This is always an integer between 1 and k for any distribution besides the uniform distribution on all clauses. Known algorithms use $\tilde{O}(n^{r/2})$ clauses in general to identify the planted solution (with the exception of special cases which can be solved using Gaussian elimination and other algebraic techniques; see more detail below). In [FPV14] we gave an algorithm based on a subsampled power iteration that uses $\tilde{O}(n^{r/2})$ clauses to identify the planted assignment for any Q.

Our general formulation of the planted k-SAT problem and the notion of distribution complexity reveal a connection between planted k-SAT and the problem of inverting a PRG based on

Goldreich's candidate one-way function [Gol00], for which the link between r-wise independence and algorithmic tractability was known before [MST06, AM09, BQ09, ABR12]. In this problem for a fixed predicate $P : \{\pm 1\}^k \to \{-1, 1\}$, we are given access to samples from a distribution P_σ, for a planted assignment $\sigma \in \{\pm 1\}^n$. A random sample from this distribution is a randomly and uniformly chosen ordered k-tuple of variables (without repetition) $x_{i_1}, \ldots, x_{i_k}$ together with the value $P(\sigma_{i_1}, \ldots, \sigma_{i_k})$. As in the problem above, the goal is to recover σ given m random and independent samples from P_σ or at least to be able to distinguish any planted distribution from one in which the value is a uniform random coin flip (in place of $P(\sigma_{i_1}, \ldots, \sigma_{i_k})$). The number of evaluations of P for which the problem remains hard determines the *stretch* of the pseudo-random generator (PRG). We note that despite the similarities between these two types of planted problems, we are not aware of any reductions between them (in Appendix 6 we show some relationships between these models and an even more general planted CSP model of Abbe and Montanari [AM15]).

Bogdanov and Qiao [BQ09] show that an SDP-based algorithm of Charikar and Wirth [CW04] can be used to find the input (which is the planted assignment) for any predicate that is *not* pairwise-independent using $m = O(n)$ such evaluations. The same approach can be used to recover the input for any $(r-1)$-wise (but not r-wise) independent predicate using $O(n^{r/2})$ evaluations [App16].

Another important family of algorithms for recovering the planted assignment in Goldreich's PRG is algebraic, based on Gaussian elimination and its generalizations [MST06, AL16]. These attacks algorithms are not captured by the framework of statistical algorithms we work with in this paper. While algebraic approaches also apply to planted satisfiability problems, almost all planting functions Q (in a measure-theoretic sense) are resilient against such algorithms, and any planted satisfiability problem can be made resistant by adding an ϵ-fraction of uniformly random constraints.

The assumption that recovering the planted assignment in this problem is hard for some predicate has been used extensively in complexity theory and cryptography [Ale11, Gol00, IKOS08, ABW10, App13], and the hardness of a decision version of this planted k-CSP is stated as the DCSP hypothesis in [BKS13]. Applebaum [App13] reduced the search problem (finding the planted assignment) to the decision problem (distinguishing the output from uniformly random). Our lower bounds below will be for this second, a priori easier, task.

Nearly optimal integrality gaps for LP and SDP hierarchies were recently given for this problem [OW14] (and references therein) for $\Omega(n^{r/2-\epsilon})$ evaluations of a predicate that is $(r-1)$-wise but not r-wise independent. Goldreich's PRG is shown to be an ϵ-biased generator in [MST06, ABR12], and lower bounds against DPLL-style algorithms are given in [CEMT09]. Applebaum and Lovett [AL16] give lower bounds against algebraic attacks in a framework based on polynomial calculus.

For a survey of these developments, see [App16].

1.1 Summary of results

For the planted k-SAT problems and the planted k-CSPs arising from Goldreich's construction we address the following question: *How many random constraints are needed to efficiently recover the planted assignment?*

For these problems we prove unconditional lower bounds for a broad class of algorithms. Statistical (query) algorithms, defined by Kearns in the context of PAC learning [Kea98] and by Feldman *et al.* [FGR$^+$12] for general problems on distributions, are algorithms that can be implemented without explicit access to random clauses, only being able to estimate expectations of functions of a random constraint to a desired accuracy. Many of algorithmic approaches used in machine learning theory and practice have been shown to be implementable using statistical queries (e.g. [BFKV98, DV08, BDMN05, CKL$^+$06, BF15]; see [Fel17] for a brief overview) including most standard approaches to convex optimization [FGV15]. Other common techniques such as Expectation Maximization (EM) [DLR77], MCMC optimization [TW87, GS90], (generalized) method of moments [Han12], and simulated annealing [KJV83, Č85] are also known to fit into this framework. The only known problem for which a superpolynomial separation between the complexity of statistical algorithms and the usual computational complexity is known is solving linear equations over a finite field (which can be done via Gaussian elimination).

The simplest form of algorithms that we refer to as statistical are algorithm that can be implemented using evaluations of Boolean functions on a random sample. Formally, for a distribution D over some domain (in our case all k-clauses) 1-STAT oracle is the oracle that given any function $h : X \to \{0, 1\}$ takes a random sample x from D and returns $h(x)$. While lower bounds for this oracle are easiest to state and interpret, the strongest form of our lower bounds is for algorithms that use VSTAT oracle defined in [FGR$^+$12]. VSTAT(t) oracle captures the information about the expectation of a given function that is obtained by estimating it on t independent samples.

Definition 1.1. *Let D be the input distribution over the domain X. For an integer parameter $t > 0$, for any query function $h : X \to [0, 1]$, VSTAT(t) returns a value $v \in [p - \tau, p + \tau]$ where $p = \mathbb{E}_D[h(x)]$ and $\tau = \max\left\{\frac{1}{t}, \sqrt{\frac{p(1-p)}{t}}\right\}$.*

This oracle is based on the well-known statistical query oracle defined by Kearns [Kea98] that uses the same tolerance τ for all query functions. The VSTAT(t) oracle corresponds more tightly to access to t samples and allows us to prove upper and lower bounds that closely correspond to known algorithmic bounds.

We show that the distribution complexity parameter r characterizes the number of constraints (up to lower order terms) that an efficient statistical algorithm needs to solve instances of either problem. For brevity we state the bound for the planted k-SAT problem but identical bounds apply to Goldreich's k-CSP. Our lower bound shows that any polynomial time statistical algorithm needs $\tilde{\Omega}(n^r)$ constraints to even *distinguish* clauses generated from a distribution with a planted assignment from uniformly random constraints (the decision problem). In addition, exponential time is required if $\tilde{\Omega}(n^{r-\epsilon})$ clauses are used for any $\epsilon > 0$.

More formally, for a clause distribution Q and an assignment σ let Q_σ denote the distribution over clauses proportional to Q for the planted assignment σ (see Sec. 2 for a formal definition). Let U_k denote the uniform distribution over k-clauses.

Theorem 1.2. *Let Q be a distribution over k-clauses of complexity r. Then any (randomized) statistical algorithm that, given access to a distribution D that equals U_k with probability $1/2$ and equals Q_σ with probability $1/2$ for a randomly and uniformly chosen $\sigma \in \{\pm 1\}^n$, decides correctly whether $D = Q_\sigma$ or $D = U_k$ with probability at least $2/3$ needs either (1) $\Omega(q)$ calls to VSTAT($\frac{n^r}{(\log q)^r}$) for any $q \geq 1$ or (2) $\Omega((\frac{n}{\log n})^r)$ calls to 1-STAT.*

It is easy to see that this lower bound is essentially tight for statistical algorithms using the VSTAT oracle (since noisy r-XOR-SAT can be solved using a polynomial (in n^r) number of queries to VSTAT($O(n^r)$) that can determine the probability of each clause). Surprisingly, this lower bound is quadratically larger than the upper bound of $\tilde{O}(n^{r/2})$ that can be achieved using samples themselves [FPV14]. While unusual, this is consistent with a common situation where an implementation using a statistical oracle requires polynomially more samples (for example in the case of algorithms for learning halfspaces). Still this discrepancy is an interesting one to investigate in order to better understand the power of statistical algorithms and lower bounds against them. We show that there exist natural strengthenings of the VSTAT and 1-STAT oracles that bridge this gap. Specifically, we extend the oracles to functions with values in a larger discrete range $\{0, 1, \ldots, L-1\}$ for $L \geq 2$: 1-MSTAT(L) oracle is the oracle that given any function $h : X \to \{0, 1, \ldots, L-1\}$ takes a random sample x from D and returns $h(x)$ and VSTAT is extended similarly to MVSTAT (we postpone the formal details and statements for this oracle to Section 2.2). This strengthening interpolates between the full access to samples which corresponds to $L = |X_k|$ and the standard statistical query oracles (corresponding to $L = 2$) and hence is a natural one to investigate.

We prove nearly matching upper and lower bounds for the stronger oracle: (a) there is an efficient statistical algorithm that uses $\tilde{O}(n^{r/2})$ calls to 1-MSTAT($O(n^{\lceil r/2 \rceil})$) and identifies the planted assignment; (b) there is no algorithm that can solve the problem described in Theorem 1.2 using less than $\tilde{O}(n^{r/2})$ calls to 1-MSTAT($n^{r/2}$). We state the upper bound more formally:

Theorem 1.3. *Let Q be a clause distribution of distribution complexity r. Then there exists an algorithm that uses $O(n^{r/2} \log^2 n)$ calls to 1-MSTAT($n^{\lceil r/2 \rceil}$) and time linear in the number of oracle calls to identify the planted assignment with probability $1 - o(1)$.*

We prove this bound by showing that the algorithm from [FPV14] based on a subsampled power iteration can be implemented using statistical query oracles. The same upper bound holds for Goldreich's planted k-CSP.

In addition to providing a matching lower bound, the algorithm gives an example of statistical query algorithm for performing power iteration to compute eigenvectors or singular vectors. Spectral algorithms are among the most commonly used for problems with planted solutions (including Flaxman's algorithm [Fla03] for planted satisfiability) and our lower bounds can be used to derive lower bounds against such algorithms. The alternative approach for solving planted constraint satisfaction problems with $O(n^{r/2})$ samples is to use an SDP solver as shown in [BQ09] (with the "birthday paradox" as shown in [OW14]; see also [App16]). This approach can also be implemented using statistical queries, although a direct implementation using a generic SDP solver such as the one we describe in Section 4 will require quadratically more samples and will not give a non-trivial statistical algorithm for the problem (since solving using $O(n^r)$ clauses is trivial).

We now briefly mention some of the corollaries and applications of our results.

Evidence for Feige's hypothesis: A closely related problem is refuting the satisfiability of a random k-SAT formula (with no planting), a problem conjectured to be hard by Feige [Fei02]. A refutation algorithm takes a k-SAT formula Φ as an input and returns either SAT or UNSAT. If Φ is satisfiable, the algorithm always returns SAT and for Φ drawn uniformly at random from all k-SAT formulas of n variables and m clauses the algorithm must return UNSAT with probability at least 2/3. For this refutation problem, an instance becomes unsatisfiable w.h.p. after $O(n)$ clauses, but algorithmic bounds are as high as those for finding a planted assignment under the noisy XOR distribution: $O(n^{k/2})$ clauses suffice[FGK05, COGLS04, HPS09, GL03, FO04, AOW15].

To relate this problem to our lower bounds we define an equivalent distributional version of the problem. In this version the input formula is obtained by sampling m i.i.d. clauses from some unknown distribution D over clauses. The goal is to say UNSAT (with probability at least 2/3) when clauses are sampled from the uniform distribution and to say SAT for every distribution supported on simultaneously satisfiable clauses.

In the distributional setting, an immediate consequence of Theorem 1.2 is that Feige's hypothesis holds for the class of statistical query algorithms. The proof (see Thm. 3.8) follows from the fact that our decision problem (distinguishing between a planted k-SAT instance and the uniform k-SAT instance) is a special case of the distributional refutation problem.

Hard instances of k-SAT: Finding distributions of planted k-SAT instances that are algorithmically intractable has been a pursuit of researchers in both computer science and physics. The distribution complexity parameter defined here generalizes the notion of "quiet plantings" studied in physics [BHL$^+$02, JMS05, KZ09, KMZ14] to an entire hierarchy of "quietness". In particular, there are easy to generate distributions of satisfiable k-SAT instances with distribution complexity as high as $k-1$ ($r = k$ can be achieved using XOR constraints but these instances are solvable by Gaussian elimination). These instances can also serve as strong tests of industrial SAT solvers as well as the underlying hard instances in cryptographic applications. In recent work, Blocki *et al.* extended our lower bounds from the Boolean setting to $\mathcal{Z}_d$ and applied them to show the security of a class of humanly computable password protocols [BBDV14].

Lower bounds for convex programs: Our lower bounds imply limitations of using convex programs to recover planted solutions. For example, any convex program whose objective is the sum of objectives for individual constraints (as is the case for canonical LPs/SDPs for CSPs) and distinguishes between a planted CSP instance and a uniformly generated one must have dimension at least $\tilde{\Omega}(n^{r/2})$. In particular, this lower bound applies to lift-and-project hierarchies where the number of solution space constraints increases (and so does the cost of finding a violated constraint), but the dimension remains the same. Moreover, since our bounds are for detecting planted solutions, they imply large integrality gaps for convex relaxations of this dimension. These bounds follow from statistical implementations of algorithms for convex optimization given in [FGV15]. We emphasize that the lower bounds apply to convex relaxations themselves and make no assumptions on how the convex relaxations are solved (in particular, the solver does not need to be a statistical algorithm). An example of such lower bound is given below. Roughly speaking, the corollary says that any convex program whose objective value is significantly higher for the uniform distribution over clauses, U_k, compared to a planted distribution Q_σ must have a large dimension, independent of the number of constraints.

Corollary 1.4. *Let Q be a distribution over k-clauses of complexity r. Assume that there exists a mapping that maps each k-clause $C \in X_k$ to a convex function $f_C : K \to [-1,1]$ over some bounded, convex and compact N-dimensional set K. Further assume that for some $\epsilon > 0$ and $\alpha \in \mathbb{R}$:*

$$\Pr_{\sigma\in\{\pm1\}^n}\left[\min_{x\in K}\left\{\mathop{\mathbb{E}}_{C\sim Q_\sigma}[f_C(x)]\right\} \le \alpha\right] \ge 1/2.$$

and

$$\min_{x\in K}\left\{\mathop{\mathbb{E}}_{C\sim U_k}[f_C(x)]\right\} > \alpha + \epsilon.$$

Then $N = \tilde{\Omega}\left(n^{r/2}\cdot\epsilon\right)$.

We note that conditions on the value of the convex program that we make are weaker than the standard conditions that a convex relaxation must satisfy. Specifically, it is usually assumed that a convex relaxation does not increase the value of the objective (for example, the value for a satisfiable instance must be 0) and also that the minimum of the objective function for all "bad" instances will be noticeably larger than that of the "good" instances. In Section 4 we also prove lower bounds against convex programs in exponentially high dimension as long as the appropriate norms of points in the domain and gradients are not too large. We are not aware of this form of lower bounds against convex programs for planted satisfiability stated before. We also remark that our lower bounds are incomparable to lower bounds for programs given in [OW14] since they analyze a specific SDP for which the mapping $\mathcal{M}$ maps to functions over an $O(n^k)$-dimensional set K is defined using a high level of the Sherali-Adams or Lovász-Schrijver hierarchies. Further details are given in Section 4.

1.2 Overview of the technique

Our proof of the lower bound builds on the notion of statistical dimension given in [FGR$^+$12] which itself is based on ideas developed in a line of work on statistical query learning [Kea98, BFJ$^+$94, Fel12].

Our primary technical contribution is a new, stronger notion of statistical dimension and its analysis for planted k-CSP problems. The statistical dimension in [FGR$^+$12] is based on upper-bounding average or maximum pairwise correlations between appropriately defined density functions. While these dimensions can be used for our problem (and, indeed, were a starting point for this work) they do not lead to the tight bounds we seek. Specifically, at best they give lower bounds for VSTAT($n^{r/2}$), whereas we will prove lower bounds for VSTAT(n^r) to match the current best upper bounds.

Our stronger notion directly examines a natural operator, which, for a given function, evaluates how well the expectation of the function discriminates between different distributions. We show that a norm of this operator for large sets of input distributions gives a lower bound on the complexity of any statistical query algorithm for the problem. Its analysis for our problem is fairly involved and a key element of the proof is the use of concentration of polynomials on $\{\pm 1\}^n$ (derived from the hypercontractivity results of Bonami and Beckner [Bon70, Bec75]).

We remark that the k-XOR-SAT problem is equivalent to PAC learning of (general) parity functions from random k-sparse examples. The latter is the classic problem addressed by Kearns original lower bound [Kea98]. While superficially the planted setting is similar to learning of k-sparse parities from random uniform examples for which optimal statistical query lower bounds are well-known and easy to derive, the problems, techniques and the resulting bounds are qualitatively different. One significant difference is that the correlation between parity functions on the uniform distribution is 0, whereas in our setting the distributions are not uniform and pairwise correlations between them can be relatively large. Moreover, as mentioned earlier, the techniques based on pairwise correlations do not suffice for the strong lower bounds we give.

Our stronger technique gives further insight into the complexity of statistical algorithms and has a natural interpretation in terms of the geometry of the space of all planted assignments with a metric defined (between pairs of assignments) to capture properties of statistical algorithms. The fraction of solutions that are at distance greater than some threshold from a fixed assignment goes up sharply from exponentially small to a polynomial fraction as the distance threshold increases. We call this a 'paring' transition as a large number of distributions become amenable to being separated from the planted solution and discarded.

We conjecture that our lower bounds hold for *all* algorithms with the exception of those based on Gaussian elimination. Formalizing "based on Gaussian elimination" requires substantial care. Indeed, in an earlier version of this work we excluded Gaussian elimination by only excluding density functions of low algebraic degree. (Here algebraic degree refers to the degree of the polynomial over $\mathcal{Z}_2^k$ required to represent the function. For example, the parity function equals to $x_1 + x_2 + \cdots + x_k$ and therefore has algebraic degree 1). This resulted in a conjecture that was subsequently disproved by Applebaum and Lovett [AL16] using an algorithm that combines Gaussian elimination with fixing some of the variables. An alternative approach to excluding Gaussian elimination-based methods is to exploit their fragility to even low rates of random noise. Here random noise would correspond to mixing in of random and uniform constraints to the distribution. In other words for $\alpha \in [0,1]$, Q becomes $Q^\alpha = (1-\alpha)Q + \alpha U_k$. Observe that for all constant $\alpha < 1$, the complexity of Q^α is the same as the complexity of Q.

Conjecture 1.5. *Let Q be any distribution over k-clauses of complexity r and $\alpha \in (0,1)$. Then any polynomial-time (randomized) algorithm that, given access to a distribution D that equals either U_k or Q_σ^α for some $\sigma \in \{\pm 1\}^n$, decides correctly whether $D = Q_\sigma^\alpha$ or $D = U_k$ with probability at least $2/3$ needs at least $\tilde{\Omega}(n^{r/2})$ clauses.*

We conjecture that an analogous statement also holds for Goldreich's k-CSP. Note that in this case mixing in an α-fraction of random and uniform constraints can be equivalently seen as flipping the given value of the predicate with probability $\alpha/2$ randomly and independently for each constraint.

1.3 Other related work

Hypergraph Partitioning. Another closely related model to planted satisfiability is random hypergraph partitioning, in which a partition of the vertex set is fixed, then k-uniform hyperedges added with probabilities that depend on their overlap with the partition. To obtain a planted satisfiability model from a planted hypergraph, let the vertex set be the set of $2n$ literals, with the partition given by the planted assignment σ. A k-clause is then a k-uniform hyperedge. The two models are not exactly equivalent, as in planted satisfiability we have the extra information that pairs of literals corresponding to the same variable must receive different assignments; however, to the best of our knowledge all the known algorithmic approaches to planted satisfiability work for planted hypergraph partitioning as well. The Goldreich CSP model is closely related to a hypergraph version of the censored block model [AM15, ABBS14] in which random hyperedges are labeled with values that depend on how the edges overlap with a planted partition.

The case $k = 2$ of k-uniform hypergraph partitioning is called the stochastic block model. The input is a random graph with different edge probabilities within and across an unknown partition of the vertices, and the algorithmic task is to recover partial or complete information about the partition given the resulting graph. Work on this model includes Bopanna [Bop87], McSherry's general-purpose spectral algorithm [McS01], and Coja-Oghlan's algorithm that works for graphs of constant average degree [CO06].

Recently an intriguing threshold phenomenon was conjectured by Decelle, Krzakala, Moore, and Zdeborová [DKMZ11]: there is a sharp threshold separating efficient partial recovery of the partition from information-theoretic impossibility of recovery. This conjecture was proved in a series of works [MNS15, Mas14, MNS13]. In the same work Decelle et al. conjecture that for a planted q-coloring, there is a gap of algorithmic intractability between the impossibility threshold and the efficient recovery threshold. Neither lower bounds nor an efficient algorithm at their conjectured threshold are currently known. In our work we consider planted bipartitions of k-uniform hypergraphs, and show that the behavior is dramatically different for $k \geq 3$. Here, while the information theoretic threshold is still at a linear number of hyperedges, we give evidence that the efficient recovery threshold can be much larger, as high as $\tilde{\Theta}(n^{k/2})$. In fact, our lower bounds hold for the problem of distinguishing a random hypergraph with a planted partition from a uniformly random one and thus give computational lower bounds for checking hypergraph quasirandomness (see [Tre08] for more on this problem). Throughout the paper we will use the terminology of planted satisfiability (assignments, constraints, clauses) but all results apply also to random hypergraph partitioning.

Shattering and paring. Random satisfiability problems (without a planted solution) such as k-SAT and k-coloring random graphs exhibit a shattering phenomenon in the solution space for large enough k [KMRT$^+$07, ACO08]: as the density of constraints increases, the set of all solutions evolves from a large connected cluster to a exponentially large set of well-separated clusters. The shattering threshold empirically coincides with the threshold for algorithmic tractability (while this is the case for large k, for $k = 3, 4$ there is some evidence that the survey propagation algorithm may succeed beyond the shattering threshold [MPRT16]). Shattering has also been used to prove that certain algorithms fail at high enough densities [GS14].

Both the shattering and paring phenomena give an explanation for the failure of known algorithms on random instances. Both capture properties of local algorithms, in the sense that in both cases, the performance of Gaussian elimination, an inherently global algorithm, is unaffected by the geometry of the solution space: both random k-XOR-SAT and random planted k-XOR-SAT are solvable at all densities despite exhibiting shattering and paring respectively.

The paring phenomenon differs from shattering in several significant ways. As the paring transition is a geometric property of a carefully chosen metric, there is a direct and provable link between paring and algorithmic tractability, as opposed to the empirical coincidence of shattering and algorithmic failure. In addition, while shattering is known to hold only for large enough k, the paring phenomenon holds for all k, and already gives strong lower bounds for 3-uniform constraints.

One direction for future work would be to show that the paring phenomenon exhibits a sharp threshold; in other words, improve the analysis of the statistical dimension of planted satisfiability in Section 5 to remove the logarithmic gap between the upper and lower bounds. An application of such an improvement would be to apply the lower bound framework to the planted coloring conjecture from [DKMZ11]; as the gap between impossibility and efficient recovery is only a constant factor there, the paring transition would need to be located more precisely.

2 Definitions

2.1 Planted satisfiability

We now define a general model for planted satisfiability problems that unifies various previous ways to produce a random k-SAT formula where the relative probability that a clause is included in the formula depends on the number of satisfied literals in the clause [Fla03, JMS05, AJM05, KV06b, KZ09, COCF10, KMZ14].

Fix an assignment $\sigma \in \{\pm 1\}^n$. We represent a k-clause by an ordered k-tuple of literals from $x_1, \dots x_n, \overline{x}_1, \dots \overline{x}_n$ with no repetition of variables and let X_k be the set of all such k-clauses. For a k-clause $C = (l_1, \dots, l_k)$ let $\sigma(C) \in \{\pm 1\}^k$ be the k-bit string of values assigned by σ to literals in C, that is $\sigma(l_1), \dots, \sigma(l_k)$, where $\sigma(l_i)$ is the value of literal l_i in assignment σ with -1 corresponding to TRUE and 1 to FALSE. In a planted model, we draw clauses with probabilities that depend on the value of $\sigma(C)$.

A planted distribution Q_σ is defined by a distribution Q over $\{\pm 1\}^k$, that is a function $Q : \{\pm 1\}^k \to \mathbb{R}^+$ such that

$$\sum_{y \in \{\pm 1\}^k} Q(y) = 1.$$

To generate a random formula, $F(Q, \sigma, m)$ we draw m i.i.d. k-clauses according to the probability distribution Q_σ, where

$$Q_\sigma(C) = \frac{Q(\sigma(C))}{\sum_{C' \in X_k} Q(\sigma(C'))}.$$

By concentrating the support of Q only on satisfying assignments of an appropriate predicate we can generate satisfiable distributions for any predicate, including k-SAT, k-XOR-SAT, and k-NAE-SAT. In most previously considered distributions Q is a symmetric function, that is Q_σ depends only on the number of satisfied literals in C. For brevity in such cases we define Q as a function from $\{0, \ldots k\}$ to $\mathbb{R}^+$. For example, the planted uniform k-SAT distribution fixes one assignment $\sigma \in \{\pm 1\}^n$ and draws m clauses uniformly at random conditioned on the clauses being satisfied by σ. In our model, this corresponds to setting $Q(0) = 0$ and $Q(i) = 1/(2^k - 1)$ for $i \geq 1$. Planted k-XOR-SAT, on the other hand, corresponds to setting $Q(i) = 0$ for i even, and $Q(i) = 1/2^{k-1}$ for i odd.

Problems. The algorithmic problems studied in this paper can be stated as follows: Given the function Q and a sample of m independent clauses drawn according to Q_σ, recover σ, or some τ correlated with σ. Note that since unsatisfiable clauses are allowed to have non-zero weight, for some distributions the problem is effectively satisfiability with random noise. Our lower bounds are for the potentially easier problem of distinguishing a randomly and uniformly chosen planted distribution from the uniform distribution over k-clauses. Namely, let $\mathcal{D}_Q$ denote the set of all distributions Q_σ, where $\sigma \in \{\pm 1\}^k$ and U_k be the uniform distribution over k-clauses. Let $\mathcal{B}(\mathcal{D}_Q, U_k)$ denote the decision problem in which given samples from an unknown input distribution $D \in \mathcal{D}_Q \cup \{U_k\}$ the goal is to output 1 if $D \in \mathcal{D}_Q$ and 0 if $D = U_k$.

In Goldreich's planted k-CSP problem for a predicate $P : \{\pm 1\}^k \to \{-1, 1\}$, we are given access to samples from a distribution P_σ, where σ is a planted assignment in $\{\pm 1\}^n$. A random sample from this distribution is a randomly and uniformly chosen ordered k-tuple of variables (without repetition) $x_{i_1}, \ldots, x_{i_k}$ together with the value $P(\sigma_{i_1}, \ldots, \sigma_{i_k})$. To allow for predicates with random noise and further generalize the model we also allow any real-valued $P : \{\pm 1\}^k \to [-1, 1]$. For such P, instead of the value $P(\sigma_{i_1}, \ldots, \sigma_{i_k})$, a randomly and independently chosen value $b \in \{-1, 1\}$ such that $\mathbb{E}[b] = P(\sigma_{i_1}, \ldots, \sigma_{i_k})$ is output.

As in the problem above, the goal is to recover σ given m random and independent samples from P_σ or at least to be able to distinguish any planted distribution from one in which the value is a uniform random coin flip (or, equivalently, the distribution obtained when the function $P \equiv 0$). Our goal is to understand the smallest number m of k-clauses that suffice to find the planted assignment or at least to distinguish a planted distribution from a uniform one.

For a clause distribution Q, we define its *distribution complexity* $r(Q)$ as the smallest integer $r \geq 1$ for which there exists a set $S \subseteq [k]$ of size r and

$$\hat{Q}(S) \doteq \frac{1}{2^k} \cdot \sum_{y \in \{\pm 1\}^k} \left[Q(y) \prod_{i \in S} y_i \right] \neq 0. \tag{1}$$

$\hat{Q}(S)$ is the Fourier coefficient of the function Q on the set S (see Sec. 5 for a formal definition). For a symmetric function the value of $\hat{Q}(S)$ depends only on $|S|$ and therefore we refer to the value of the coefficient for sets of size ℓ by $\hat{Q}(\ell)$.

To see the difference between a hard and easy distribution Q, first consider planted uniform k-SAT: $Q(0) = 0$, $Q(i) = 1/(2^k - 1)$ for $i \geq 1$. The distribution complexity of Q is $r = 1$.

Next, consider the noisy parity distribution (or noisy planted k-XOR-SAT) with $Q(l) = \delta/2^{k-1}$ for l even, and $Q(l) = (2-\delta)/2^{k-1}$ for l odd, for some $\delta \neq 1$. In this case, we have $\hat{Q}(l) = 0$ for $1 \leq l \leq k-1$, and so the distribution complexity of Q is $r = k$. We will see that such parity-type distributions are in fact the hardest for statistical algorithms to detect.

2.2 Statistical algorithms

We can define planted satisfiability as the problem of identifying an unknown distribution D on a domain X given m independent samples from D. For us, X is the set of all possible k-clauses or k-hyperedges, and each partition or assignment σ defines a unique distribution D_σ over X.

Extending the work of Kearns [Kea98] in learning theory, Feldman *et al.* [FGR$^+$12] defined statistical query algorithms for problems over distributions. Roughly speaking, these are algorithms that do not see samples from the distribution but instead have access to estimates of the expectation of any bounded function of a sample from the distribution. More formally, a statistical algorithm can access the input distribution via one of the following oracles.

Definition 2.1 (1-MSTAT(L) oracle)**.** *Let D be the input distribution over the domain X. Given any function $h : X \to \{0, 1, \ldots, L-1\}$, 1-MSTAT($L$) takes a random sample x from D and returns $h(x)$.*

This oracle is a generalization of the 1-STAT oracle from [FGR$^+$12] and was first defined by Ben-David and Dichterman in the context of PAC learning [BD98]. It was also more recently studied in [SD15, SVW15]. For the planted SAT problem this oracle allows an algorithm to evaluate a multi-valued function on a random clause. By repeating the query, the algorithm can estimate the expectation of the function as its average on independent samples. Being able to output one of multiple possible values gives the algorithm considerable flexibility, e.g., each value could correspond to whether a clause has a certain pattern on a subset of literals. With $L = n^k$, the algorithm can identify the random clause. We will therefore be interested in the trade-off between L and the number of queries needed to solve the problem.

The next oracle is from [FGR$^+$12].

Definition 2.2 (VSTAT oracle)**.** *Let D be the input distribution over the domain X. For an integer parameter $t > 0$, for any query function $h : X \to [0, 1]$, VSTAT(t) returns a value $v \in [p - \tau, p + \tau]$ where $p = \mathbb{E}_D[h(x)]$ and $\tau = \max\left\{\frac{1}{t}, \sqrt{\frac{p(1-p)}{t}}\right\}$.*

The definition of τ means that VSTAT(t) can return any value v for which the distribution $B(t, v)$ (outcomes of t independent Bernoulli variables with bias v) is close to $B(t, E[h])$ in total variation distance [FGR$^+$12]. In most cases $p > 1/t$ and then τ also corresponds to returning the expectation of a function to within the standard deviation error of averaging the function over t samples. However, it is important to note that within this constraint on the error, the oracle can return any value, possibly in an adversarial way.

In this paper, we also define the following generalization[1] of the VSTAT oracle to multi-valued functions.

Definition 2.3 (MVSTAT oracle)**.** *Let D be the input distribution over the domain X, $t, L > 0$ be integers. For any multi-valued function $h : X \to \{0, 1, \ldots, L-1\}$ and any set $\mathcal{S}$ of subsets of*

[1] For simplicity, this definition generalizes VSTAT only for Boolean query functions.

$\{0, \ldots, L-1\}$, *MVSTAT*(L,t) *returns a vector* $v \in \mathbb{R}^L$ *satisfying for every* $Z \in \mathcal{S}$

$$\left| \sum_{\ell \in Z} v_l - p_Z \right| \leq \max \left\{ \frac{1}{t}, \sqrt{\frac{p_Z(1-p_Z)}{t}} \right\},$$

where $p_Z = \Pr_D[h(x) \in Z]$. *The query cost of such a query is* $|\mathcal{S}|$.

We note that VSTAT(t) is equivalent to MVSTAT$(2,t)$ (the latter only allows Boolean queries but that is not an essential difference) and any query to MVSTAT(L,t) can be easily answered using L queries to VSTAT$(4 \cdot Lt)$ (see Thm. 7.2 for a proof). The additional strength of this oracle comes from allowing the sets in $\mathcal{S}$ to depend on the unknown distribution D and, in particular, be fixed but unknown to the algorithm. This is useful for ensuring that potential functions of our discrete power iteration algorithm for planted SAT behave in the same way as if the algorithm were executed on true samples (see Section 8.4). Another useful way to think of L-valued oracles in the context of vector-based algorithms is as a vector of L Boolean functions which are non-zero on disjoint parts of the domain. This view also allows to extend MVSTAT to bounded-range (non-Boolean) functions.

An important property of every one of these oracles is that it can be easily simulated using t samples (in the case of VSTAT/MVSTAT the success probability is a positive constant but it can be amplified to $1-\delta$ using $O(t\log(1/\delta))$ samples). The goal of our generalization of oracles to $L > 2$ was to show that even nearly optimal sample complexity can be achieved by a statistical algorithm using an oracle for which a nearly matching lower bound applies.

3 Results

We state our upper and lower bounds for the planted satisfiability problem. Identical upper and lower bounds apply to Goldreich's planted k-CSPs with r being the degree of lowest-degree non-zero Fourier coefficient of P. For brevity, we omit the repetitive definitions and statements in this section. In Section 6 we give the extension of our lower bounds to this problem and also make the connections between the two problems explicit.

3.1 Lower bounds

We begin with lower bounds for *any* statistical algorithm. For a clause distribution Q let $\mathcal{B}(\mathcal{D}_Q, U_k)$ denote the decision problem of distinguishing whether the input distribution is one of the planted distributions or is uniform.

Theorem 3.1. *For an assignment* $\sigma \in \{\pm 1\}^n$, *let* Q_σ *be a distribution over* k*-clauses of complexity* r *and* $\mathcal{D}_Q$ *be this family of distributions. Assume that the input distribution* D *is* U_k *with probability* $1/2$ *and with the remaining probability* $1/2$ *it is* Q_σ *for a uniform random* $\sigma \in \{\pm 1\}^n$. *Then any (randomized) statistical algorithm that decides correctly whether* $D \in \mathcal{D}_Q$ *or* $D = U_k$ *with probability at least* $2/3$ *(over the choice of* D *and randomness of the algorithm) needs either*

1. *m calls to the 1-MSTAT(L) oracle with* $m \cdot L \geq c_1 \left(\frac{n}{\log n}\right)^r$ *for a constant* $c_1 = \Omega_k(1)$, *OR*

2. *q queries to MVSTAT* $\left(L, \frac{c_2}{L} \cdot \frac{n^r}{(\log q)^r}\right)$ *for a constant* $c_2 = \Omega_k(1)$ *and any* $q \geq L$.

The first part of the theorem exhibits the trade-off between the number of queries m and the number of values the query can take L. It might be helpful to think of the latter as evaluating L disjoint functions on a random sample, a task that would have complexity growing with L. The second part of the theorem is a superpolynomial lower bound (in n, for any fixed r) if the parameter t (recall the oracle is allowed only an error equal to the standard deviation of averaging over t random samples) is less than $n^r/(\log n)^{2r}$.

3.2 Algorithms

We next turn to our algorithmic results, motivated by two considerations. First, the $O(n^{r/2})$-clause algorithm implicit in [BQ09] does not appear to lead to a non-trivial statistical algorithm. Second, much of the literature on upper bounds for planted problems uses spectral methods, and so we aim to implement such spectral algorithms statistically.

The algorithm we present is statistical and nearly matches the lower bound. It can be viewed as a discrete rounding of the power iteration algorithm for a suitable matrix constructed from the clauses of the input.

Theorem 3.2. *Let $\mathcal{Z}_Q$ be a planted satisfiability problem with clause distribution Q having distribution complexity r. Then there exists an algorithm to solve $\mathcal{Z}_Q$ using $O(n^{r/2}\log n)$ random clauses and time linear in this number. This algorithm can be implemented statistically in any of the following ways.*

1. *Using $O(n^{r/2}\log^2 n)$ calls to 1-MSTAT$(n^{\lceil r/2\rceil})$;*
2. *For even r: using $O(\log n)$ calls to MVSTAT$(n^{r/2}, n^{r/2}\log\log n)$;*
3. *For odd r: using $O(\log n)$ calls to MVSTAT$(O(n^{\lceil r/2\rceil}), O(n^{r/2}\log n))$;*

Thus for any r, the upper bound matches the lower bound up to logarithmic factors for sample size parameter $t = n^{r/2}$, with $L = n^{\lceil r/2\rceil}$ being only slightly higher in the odd case than the $L = n^{r/2}$ that the lower bound implies for such t. The algorithm is a discretized variant of the algorithm based on power iteration with subsampling from [FPV14]. The upper bound holds for the problem of finding the planted assignment exactly, except in the case $r = 1$. Here $\Omega(n\log n)$ clauses are required for complete identification since that many clauses are needed for each variable to appear at least once in the formula. In this case $O(n^{1/2}\log n)$ samples suffice to find an assignment with non-trivial correlation with the planted assignment, i.e. one that agrees with the planted assignment on $n/2 + t\sqrt{n}$ variables for an arbitrary constant t.

3.3 Statistical dimension for decision problems

For a domain X, let $\mathcal{D}$ be a set of distributions over X and let D be a distribution over X which is not in $\mathcal{D}$. For $t > 0$, the *distributional decision problem* $\mathcal{B}(\mathcal{D}, D)$ using t samples is to decide, given access to t random samples from an arbitrary unknown distribution $D' \in \mathcal{D}\cup\{D\}$, whether $D' \in \mathcal{D}$ or $D' = D$. Lower bounds on the complexity of statistical algorithms use the notion of *statistical dimension* introduced in [FGR$^+$12], based on ideas from [BFJ$^+$94, Fel12].

To prove our bounds we introduce a new, stronger notion of statistical dimension which directly examines a certain norm of the operator that discriminates between expectations taken relative to different distributions. Formally, for a distribution $D' \in \mathcal{D}$ and a reference distribution D we examine the (linear) operator that maps a function $h : X \to \mathbb{R}$ to $\mathbb{E}_{D'}[h] - \mathbb{E}_D[h]$. Our goal is to obtain bounds on a certain norm of this operator extended to a set of distributions. Specifically, the *discrimination norm* of a set of distributions $\mathcal{D}'$ relative to a distribution D is denoted by $\kappa_2(\mathcal{D}', D)$ and defined as follows:

$$\kappa_2(\mathcal{D}', D) \doteq \max_{h, \|h\|_D = 1} \left\{ \mathop{\mathbb{E}}_{D' \sim \mathcal{D}'} \left[\left| \mathop{\mathbb{E}}_{D'}[h] - \mathop{\mathbb{E}}_{D}[h] \right| \right] \right\},$$

where the norm of h over D is $\|h\|_D = \sqrt{\mathbb{E}_D[h^2(x)]}$ and $D' \sim \mathcal{D}'$ refers to choosing D' randomly and uniformly from the set $\mathcal{D}'$. A statistical algorithm can get an estimate of the expectation of a query h and use the value to determine whether the input distribution is the reference distribution or one of the distributions in $\mathcal{D}'$. Intuitively, the norm measures how well this can be done on average over distributions in $\mathcal{D}'$. In particular, we will show that if $\kappa_2(\mathcal{D}', D) = \kappa$ then a single query to VSTAT$(1/(3\kappa^2))$ cannot be used to distinguish all distributions in $\mathcal{D}'$ from D.

Our concept of statistical dimension is essentially the same as in [FGR+12] but uses $\kappa_2(\mathcal{D}', D)$ instead of average correlations.

Definition 3.3. *For $\kappa > 0$, domain X and a decision problem $\mathcal{B}(\mathcal{D}, D)$, let d be the largest integer such that there exists a finite set of distributions $\mathcal{D}_D \subseteq \mathcal{D}$ with the following property: for any subset $\mathcal{D}' \subseteq \mathcal{D}_D$, where $|\mathcal{D}'| \geq |\mathcal{D}_D|/d$, $\kappa_2(\mathcal{D}', D) \leq \kappa$. The **statistical dimension** with discrimination norm κ of $\mathcal{B}(\mathcal{D}, D)$ is d and denoted by* SDN$(\mathcal{B}(\mathcal{D}, D), \kappa)$.

The dimension is equal to (at least) d if there exists a reference distribution D and a "hard" subset of distributions $\mathcal{D}_D$, such that no large subset of $\mathcal{D}_D$ has discrimination norm larger than κ (and, consequently, cannot be distinguished from D using a single query to VSTAT$(1/(3\kappa^2))$). Here large subset means at least $1/d$ fraction of distributions in $\mathcal{D}_D$. We remark that this statistical dimension can be easily extended to general search problems as in [FGR+12] (the extension can be found in an earlier version of this work [FPV13, v5]). A detailed treatment and additional approaches to proving statistical query lower bounds for search problems can be found in a subsequent work of Feldman [Fel16].

The statistical dimension with discrimination norm κ of a problem over distributions gives a lower bound on the complexity of any statistical algorithm.

Theorem 3.4. *Let X be a domain and $\mathcal{B}(\mathcal{D}, D)$ be a decision problem over a class of distributions $\mathcal{D}$ on X and reference distribution D. For $\kappa > 0$, let $d =$ SDN$(\mathcal{B}(\mathcal{D}, D), \kappa)$ and let $L \geq 2$ be an integer.*

- *Any randomized statistical algorithm that solves $\mathcal{B}(\mathcal{D}, D)$ with probability $\geq 2/3$ over the randomness in the algorithm requires $\Omega(d/L)$ calls to MVSTAT$(L, 1/(12 \cdot \kappa^2 \cdot L))$.*
- *Any randomized statistical algorithm that solves $\mathcal{B}(\mathcal{D}, D)$ with probability $\geq 2/3$ over the randomness in the algorithm requires at least m calls to 1-MSTAT(L) for $m = \Omega\left(\min\left\{d, 1/\kappa^2\right\}/L\right)$.*

Further, the lower bound also holds when the input distribution D' is chosen randomly as follows: $D' = D$ with probability $1/2$ and D' equals a random and uniform element of $\mathcal{D}_D$ with probability $1/2$, where $\mathcal{D}_D$ is the set of distributions for which the value of d is attained.

We prove this theorem in a slightly more general form in Section 7. Our proof relies on techniques from [FGR+12] and simulations of MVSTAT and 1-MSTAT using VSTAT and 1-STAT, respectively.

In our setting the domain X_k is the set of all clauses of k ordered literals (without variable repetition); the class of distributions $\mathcal{D}_Q$ is the set of all distributions Q_σ where σ ranges over all 2^n assignments; the distribution D is the uniform distribution over X_k referred to as U_k.

In the Section 5 we prove the following bound on the statistical dimension with discrimination norm of planted satisfiability.

Theorem 3.5. *For any distribution Q over k-clauses of distributional complexity r, there exists a constant $c > 0$ (that depends on Q) such that for any $q \geq 1$,*

$$\mathrm{SDN}\left(\mathcal{B}(\mathcal{D}_Q, U_k), \frac{c(\log q)^{r/2}}{n^{r/2}}\right) \geq q.$$

For an appropriate choice of $q = n^{\theta(\log n)}$ we get, $\mathrm{SDN}(\mathcal{B}(\mathcal{D}_Q, U_k), \frac{(\log n)^r}{n^{r/2}}) = n^{\Omega_k(\log n)}$. Similarly, for any constant $\epsilon > 0$, we get $\mathrm{SDN}(\mathcal{B}(\mathcal{D}_Q, U_k), n^{r/2-\epsilon}) = 2^{n^{\Omega_k(1)}}$. By using this bound in Theorem 3.4 we obtain our main lower bounds in Theorem 3.1.

In Section 6.1 we prove the same lower bound for the generalized planted k-CSP problem. Our proof is based on a reduction showing that any statistical query for an instance of the k-CSP problem of complexity r can be converted to a query for a planted k-SAT instance of distribution complexity r. The reduction ensures that the resulting query is essentially as informative in distinguishing the planted distribution from the reference one as the original query. As a result it reduces a bound on κ_2 of a planted k-CSP problem to an almost equivalent bound on κ_2 of the corresponding planted k-SAT problem.

3.4 Corollaries and applications

3.4.1 Quiet plantings

Finding distributions of planted k-SAT instances that are algorithmically intractable has been a pursuit of researchers in both computer science and physics. It was recognized in [BHL$^+$02, JMS05] that uniform planted k-SAT is easy algorithmically due to the bias towards true literals, and so they proposed distributions in which true and false literals under the planted assignment appear in equal proportion. Such distributions have complexity $r \geq 2$ in our terminology. These distributions have been termed 'quiet plantings' since evidence of the planting is suppressed.

Further refinement of the analysis of quiet plantings was given in [KMZ14], in which the authors analyze belief propagation equations and give predicted densities at which quiet plantings transition from intractable to tractable. Their criteria for a quiet planting is exactly the equation that characterizes distribution complexity $r \geq 2$, and the conditions under which the tractability density diverges to infinity corresponds to distribution complexity $r \geq 3$.

The distribution complexity parameter defined here generalizes quiet plantings to an entire hierarchy of quietness. In particular, there are distributions of satisfiable k-SAT instances with distribution complexity as high as $k-1$ ($r = k$ can be achieved using XOR constraints but these instances are solvable by Gaussian elimination). Our main results show that for distributions with complexity $r \geq 3$, the number of clauses required to recover the planted assignment is super-linear (for statistical algorithms with $L \leq n^{r/2}$).

For examples of such distributions, consider weighting functions $Q(y)$ that depend only on the number of true literals in a clause under the planted assignment σ. We will write $Q(j)$ for the value of Q on any clause with exactly j true literals. Then setting $Q(0) = 0, Q(1) = 3/32, Q(2) =$

$1/16, Q(3) = 1/32, Q(4) = 1/8$ gives a distribution over satisfiable 4-SAT instances with distribution complexity $r = 3$, and an algorithmic threshold at $\tilde{\Theta}(n^{3/2})$ clauses. Similar constructions for higher k yield distributions of increasing complexity with algorithmic thresholds as high as $\tilde{\Theta}(n^{(k-1)/2})$. These instances are the most 'quiet' proposed and can serve as strong tests of industrial SAT solvers as well as the underlying hard instances in cryptographic applications. Note that in these applications it is important that a hard SAT instance can be obtained from an easy to sample planted assignment σ. Our lower bounds apply to the uniformly chosen σ and therefore satisfy this condition.

3.4.2 Feige's Hypothesis

As a second application of our main result, we show that Feige's 3-SAT hypothesis [Fei02] holds for the class of statistical algorithms. A refutation algorithm takes a k-SAT formula Φ as an input and returns either SAT or UNSAT. The algorithm must satisfy the following:

1. If Φ is satisfiable, the algorithm always returns SAT.

2. If Φ is drawn uniformly at random from all k-SAT formulas of n variables and m clauses, where m/n is above the satisfiability threshold (the clause density at which the formula become unsatisfiable with high probability), then the algorithm must return UNSAT with probability at least $2/3$ (or some other arbitrary constant).

As with planted satisfiability, the larger m is the easier refutation becomes, and so the challenge becomes finding efficient refutation algorithms that succeed on the sparsest possible instances. Efficient 3-SAT refutation algorithms are known for $m = \Omega(n^{3/2})$ [COGL04, FO04]. Feige hypothesized 1) that no polynomial-time algorithm can refute formulas with $m \leq \Delta n$ clauses for any constant Δ and 2) for every $\epsilon > 0$ and large enough constant Δ, there is no polynomial-time algorithm that answers UNSAT on most 3-SAT formulas but answers SAT on all formulas that have assignments satisfying $(1 - \epsilon)$-fraction of constraints. Hypothesis 2 is strictly weaker than hypothesis 1. Based on these hypotheses he derived hardness-of-approximation results for several fundamental combinatorial optimization problems.

To apply our bounds we need to first define a distributional version of the problem.

Definition 3.6. *In the distributional k-SAT refutation problem the input formula is obtained by sampling m i.i.d. clauses from some unknown distribution D over clauses. An algorithm successfully solves the distributional problem if:*

1. *The algorithm returns SAT for every distribution supported on simultaneously satisfiable clauses.*

2. *The algorithm returns UNSAT with probability at least $2/3$ when clauses are sampled from the uniform distribution and m/n is above the satisfiability threshold.*

Proposition 3.7. *The original refutation problem and distributional refutation problem are equivalent: a refutation algorithm for the original problem solves the distributional version and vice versa.*

Proof. The first direction is immediate: assume that we have a refutation algorithm A for a fixed formula. We run the refutation algorithm on the m clauses sampled from the input distribution and output the algorithm's answer. By definition, if the input distribution is uniform then the sampled clauses will give a random formula from this distribution. So A will return UNSAT with probability at least $2/3$. If the clauses in the support of the input distribution can be satisfied then the formula sampled from it will be necessarily satisfiable and A must return SAT.

In the other direction, we again run the distributional refutation algorithm A on the m clauses of Φ and output its answer (each clause is used as a new sample consecutively). If Φ was sampled from the uniform distribution above the satisfiability threshold, then the samples we produced are distributed according to the uniform distribution. Therefore, with probability at least $2/3$ A returns UNSAT. If Φ is satisfiable then consider the distribution D_Φ which is uniform over the m clauses of Φ. Φ has non-zero probability to be the outcome of m i.i.d. clauses sampled from D_Φ. Therefore A must output SAT on it since otherwise it would violate its guarantees. Therefore the output of our algorithm will be SAT for Φ. □

In the distributional setting, an immediate consequence of Theorem 3.1 is that Feige's hypothesis holds for the class of statistical algorithms.

Theorem 3.8. *Any (randomized) statistical algorithm that solves the distributional k-SAT refutation problem requires:*

1. *m calls to the 1-MSTAT(L) oracle with $m \cdot L \geq c_1 \left(\frac{n}{\log n}\right)^k$ for a constant $c_1 = \Omega_k(1)$.*
2. *q queries to MVSTAT$\left(L, \frac{c_2}{L} \cdot \frac{n^k}{(\log q)^k}\right)$ for a constant $c_2 = \Omega_k(1)$ and any $q \geq L$.*

Proof. The decision problem in Theorem 3.1 is a special case of the distributional refutation problem. Specifically, say there is such a refutation algorithm. Let Q be a fully satisfiable clause distribution with distribution complexity k. Then consider a distribution D so that either $D = U_k$ or $D = Q_\sigma \in \mathcal{D}_Q$ for a uniformly chosen $\sigma \in \{\pm 1\}^n$. Then run the refutation algorithm on D. If $D \in \mathcal{D}_Q$, then the algorithm must output SAT, and so we conclude $D \in \mathcal{D}_Q$. If $D = U_k$, then with probability $2/3$ the algorithm must output UNSAT in which case we conclude that $D = U_k$. This gives an algorithm for distinguishing U_k from $\mathcal{D}_Q$ with probability at least $2/3$, a contradiction to Theorem 3.1. □

If $r \geq 3$ and $L \leq n^{r/2}$, the lower bound on the number of clauses m is $\tilde{\Omega}(n^{r/2})$ and is much stronger than Δn conjectured by Feige. Such a stronger bound is useful for some hardness of learning results based on Feige's conjecture [DLSS13]. For $k = 3$, the $\tilde{\Omega}(n^{3/2})$ lower bound on m essentially matches the known upper bounds [COGL04, FO04].

We note that the only distributions with $r = k$ are noisy k-XOR-SAT distributions. Such distributions generate satisfiable formulas only when the noise rate is 0 and then formulas are refutable via Gaussian elimination. Therefore if one excludes the easy (noiseless) k-XOR-SAT distribution then we obtain only the stronger form of Feige's conjecture ($\epsilon > 0$) with $r = k = 3$.

3.4.3 Hardness of approximation

We note finally that optimal inapproximability results can be derived from Theorem 3.1 as well, including the fact that pairwise independent predicates (as studied in [AM09]) are approximation-resistant for the class of statistical algorithms.

Our work provides a means to generate candidate distributions of hard instances for approximation algorithms for CSP's: find a distribution Q on $\{\pm 1\}^k$ supported only on vectors that satisfy the CSP predicate with high distribution complexity (as in the example of 4-SAT above). Then statistical algorithms cannot efficiently distinguish the planted distribution (all constraints satisfied) from the uniformly random distribution (eg. $(1-2^{-k})$-fraction of constraints satisfied in the case of k-SAT).

4 Convex Programs and SQ Algorithms for Solving CSPs

In this section we show how our lower bounds for planted k-SAT together with general statistical query algorithms for solving stochastic convex programs from [FGV15] imply lower bounds on convex programs that can be used to solve planted k-SAT (analogous results also hold for Goldreich's k-CSP but we omit them for brevity). At a high level we observe that a convex relaxation can be viewed as a reduction from our planted constraint satisfaction problem to a stochastic convex optimization problem. Existence of such a reduction together with a statistical query algorithm for the corresponding stochastic convex program would violate the lower bounds that we prove. Hence, as a contrapositive, we rule out existence of several types of convex relaxations for the planted CSPs.

4.1 LP/SDP relaxations for k-CSPs

We first describe several standard ways in which Boolean constraint satisfaction problems[2] are relaxed to an LP or an SDP.

The classic SDP for a constraint satisfaction problem is the MAX-CUT SDP of Goemans and Williamson [GW95]. In this program the goal is to maximize $\sum_{i,j\in[n]}[e_{ij}(1-x_{i,j})]$, where $e_{ij}\in\mathbb{R}$ is the indicator of an edge presence in the graph and x, viewed as an $n\times n$ matrix, is constrained to be in the PSD cone with some normalization.

More generally, the canonical LP relaxation of a k-CSP with m constraints results in a program of the following type (see [O'D11] for a textbook version or [Rag08, BKS13, OW14] for some applications):

$$\text{maximize} \sum_{i\in[m]}\left(\sum_{y\in\{\pm1\}^k, y \text{ satisfies } R_i} x_{V_i,y}\right),$$

subject to $\bar{x}\in K$. Here, R_i denotes the k-ary Boolean predicate of the i-th constraint, V_i denotes the k-tuple of variables of i-th constraint and $x_{V_i,y}$ is the variable that tells whether variables in V_i are assigned values y (its constrained to be in $[0,1]$ and interpreted as probability). The set K is an $O_k(n^k)$-dimensional convex set that makes sure that $x_{V_i,y}$'s are consistent in a natural sense (with additional PSD cone constraints in the case of SDPs).

Such relaxations are a special case of even more general relaxations that are based on a linearization of the Boolean constraints (for example [KMR17]). Specifically, a linearization maps each assignment $x\in\{\pm1\}^n$ to a vector $v_x\in\mathbb{R}^N$ and each relevant k-ary Boolean constraint R to a vector $w_R\in\mathbb{R}^N$. Further, for every x and C, $C(x)=\langle v_x, w_C\rangle$. For a convex body K in $\mathbb{R}^N$ that includes $\{v_x \mid x\in\{\pm1\}^n\}$, given a set $C_1,\ldots,C_m$ of Boolean predicates on x one considers the program $\max_{v\in K}\sum_{i\in[m]}\langle v, w_{C_i}\rangle$. Note that condition $\{v_x \mid x\in\{\pm1\}^n\}\subset K$ ensures that the optimum of this program is always at least as large as the optimum of the original problem. In

[2] As usual in this literature, constraint satisfaction also refers to the problem of maximizing the number of satisfied constraints.

order to be useful, such relaxation also needs to ensure that the value of the solution to the relaxed program allows to distinguish between instances with high and low values of the optimum in the original program (for some appropriate values of "high" and "low").

Here we consider an even more general class of convex relaxations. First, we allow mapping Boolean constraints to general convex objective functions. That is, a Boolean function C over $\{\pm 1\}^n$ is mapped to a convex function f_C over a convex body $K \in \mathbb{R}^N$. We will not need an explicit mapping between the Boolean input and vectors in K but will only require that the value of the optimum of the resulting objectives allows us to distinguish between instances with high and low values of the optimum in the original program. Also note that going beyond linear objectives implies that we will be minimizing (and not maximizing) the resulting objective.

Our lower bounds apply to distributional CSPs in which constraints are sampled i.i.d. from some distribution D and they show that even estimating the value of the expected objective: $\max_{\sigma \in \{\pm 1\}^n} \mathbb{E}_{C \sim D}[\sigma(C)]$ is hard. We remark that, given $m = \Omega(n/\epsilon^2)$ samples, for every $x \in \{\pm 1\}^n$, the value of the objective based on m i.i.d. samples is within ϵ of the value of the expected objective (with high probability). Applying a convex relaxation to such a CSP leads to the following convex program: $\min_{x \in K} \mathbb{E}_{C \sim D}[f_C(x)]$, where K is a fixed convex N-dimensional set (that is not dependent the distribution D) and for every C, $f_C(x)$ is a bounded convex function over K. Such programs are referred to as *stochastic convex programs* and are well-studied in machine learning and optimization (e.g. [NJLS09, SSSS09]).

4.2 Statistical Query Algorithms for Stochastic Convex Optimization

We now describe several results from [FGV15] giving upper bounds on solving various stochastic convex programs by statistical query algorithms. Bounds for a number of additional types of convex programs are given in [FGV15] and can be applied in this context in a similar way. We start by defining the problem of distribution-independent stochastic convex optimization formally.

Definition 4.1. *For a convex set K, a set $\mathcal{F}$ of convex functions over K and $\epsilon > 0$ we denote by Opt$(K, \mathcal{F}, \epsilon)$ the problem of finding, for every distribution D over $\mathcal{F}$, x^* such that $f_D(x^*) \leq \min_{x \in K} f_D(x) + \epsilon$, where $f_D(x) \doteq \mathbb{E}_{f \sim D}[f(x)]$.*

Center-of-gravity: For general convex functions with range scaled to $[-1, 1]$, Feldman *et al.* [FGV15] describe two statistical query algorithms that both use VSTAT$(O(N^2/\epsilon^2))$ to find an ϵ-approximate solution to the stochastic convex program. The first algorithm is based on the random walk approach from [KV06a, LV06]. The second algorithm is based on the classic center-of-gravity method [Lev65] and requires fewer queries.

Theorem 4.2. *[[FGV15]] Let $K \subseteq \mathbb{R}^N$ be a convex body and let $\mathcal{F}$ be the set of all convex functions over K such that for all $x \in K, |f(x)| \leq 1$. Then there is an algorithm that solves Opt$(K, \mathcal{F}, \epsilon)$ using $O(N^2 \log(1/\epsilon))$ queries to VSTAT$(O(N^2/\epsilon^2))$.*

The theorem above ignores computational considerations since those do not play any role in our information-theoretic lower bounds. An efficient version of this algorithm is also given in [FGV15].

Mirror descent: In most practical cases the expected convex objective is optimized using simpler methods such as gradient-descent based algorithms. It is easy to see that such methods fit naturally into the statistical query framework. For example, gradient descent relies solely on knowing $\nabla f_D(x_t)$

approximately, where f_D is the optimized function and x_t is the solution at step t. By linearity of expectation, we know that $\nabla \mathbb{E}_D[f(x_t)] = \mathbb{E}_D[\nabla f(x_t)]$. This means that we can approximate $\nabla \mathbb{E}_D[f(x_t)]$ using queries to VSTAT with sufficiently large parameter. In particular, as shown in [FGV15], the classic mirror-descent method [NY83] can be implemented using a polynomial number of queries to $\mathrm{VSTAT}(O((L \cdot R \cdot \log(N)/\epsilon)^2))$ to ϵ-approximately solve any convex program whenever K is contained in the ℓ_p (for any $p \in [1,2]$) ball of radius R and the ℓ_q (for $q = 1/(1-1/p)$) norm of ∇f is bounded by L. Note that in this case the dependence of the accuracy parameter of VSTAT on the dimension is just logarithmic. We denote $\mathcal{B}_p^N(R) \doteq \{x \mid \|x\|_p \leq R\}$.

Theorem 4.3. *Let $p \in [1,2]$, $L, R > 0$, and $K \subseteq \mathcal{B}_p^N(R)$ be a convex body. Let $\mathcal{F}$ be the set of all functions f that satisfy, for all $x \in K$, $\|\nabla f(x)\|_q \leq L$ for $q = 1 - (1/p)$. Then there is an algorithm that solves Opt$(K, \mathcal{F}, \epsilon)$ using $O\left(N \log N \cdot (LR/\epsilon)^2\right)$ queries to VSTAT$(O((\log N \cdot LR/\epsilon)^2))$.*

4.3 Corollaries for Planted k-CSPs

Now observe that a convex relaxation (of the type that we defined) is just a mapping from Boolean constraints to convex functions in some class of functions $\mathcal{F}$ over a convex body K. Such mapping allows to implement a statistical query oracle for the distribution over convex functions given a statistical query oracle for the input distribution over Boolean constraints. In particular, it allows us to run a statistical query algorithm for $\mathrm{Opt}(K, \mathcal{F}, \epsilon)$ on the stochastic convex program that corresponds to the input distribution over k-clauses. Now assume that the value of the solution for the stochastic convex program corresponding to a planted k-CSP is smaller than the value of the solution for the the stochastic convex program corresponding to the uniform distribution over constraints by at least ϵ. Then a statistical query algorithm for $\mathrm{Opt}(K, \mathcal{F}, \epsilon)$ solves the decision version of our planted k-CSP. Hence, if $\mathrm{Opt}(K, \mathcal{F}, \epsilon)$ can be solved using some number of queries to a statistical oracle that violates our lower bound then a convex relaxation that satisfies these properties cannot exist. We now make these statements formally.

Theorem 4.4. *Let Q be a distribution over k-clauses of complexity r. Assume that there exists a mapping that maps each k-clause $C \in X_k$ to a convex function $f_C : K \to [-1, 1]$ over some convex N-dimensional set K that for some $\epsilon > 0$ and α satisfies:*

- $\Pr_{\sigma \in \{\pm 1\}^n} \left[\min_{x \in K} \left\{\mathbb{E}_{C \sim Q_\sigma}[f_C(x)]\right\} \leq \alpha\right] \geq 1/2$*;*
- $\min_{x \in K}\{\mathbb{E}_{C \sim U_k}[f_C(x)]\} > \alpha + \epsilon$*.*

Then for every $q \geq 1$, solving Opt$(K, \mathcal{F}, \epsilon)$ using VSTAT$\left(\frac{n^r}{(\log q)^r}\right)$ requires $\Omega(q)$ queries.

The first condition on the value of the solution requires that for most σ's the value of the minimum of the expected objective is at most α. The planted instances are usually the instances which have a higher number of satisfied clauses so this is a weakening of the condition that a convex relaxation should only decrease the value of the minimum. The second condition requires that the value of the minimum of the expected objective on the uniform distribution is at least $\alpha + \epsilon$. For comparison, the standard condition on a convex relaxation requires that the value of the solution obtained on m clauses randomly sampled from the uniform distribution is large. Our condition is weaker since for every m and $x^* = \arg\min_{x \in K}\{\mathbb{E}_{C \sim U_k}[f_C(x)]\}$,

$$\mathop{\mathbb{E}}_{C_1,\ldots,C_m \sim U_k}\left[\min_{x \in K}\left\{\frac{1}{m}\sum_i f_{C_i}(x)\right\}\right] \leq \mathop{\mathbb{E}}_{C_1,\ldots,C_m \sim U_k}\left[\frac{1}{m}\sum_i f_{C_i}(x^*)\right] = \min_{x \in K}\{\mathop{\mathbb{E}}_{C \sim U_k}[f_C(x)]\}.$$

Now Theorem 4.4 can be combined with upper bounds on the complexity of solving stochastic convex programs we gave above to obtain lower bounds on the parameters of convex relaxations that can be used to solve planted satisfiability problems. For example the algorithm described in Theorem 4.2 implies Corollary 1.4. Using Theorem 4.3 we we can exclude convex relaxations even in exponentially high dimension as long as the convex set is bounded in ℓ_p norm and functions satisfy a Lipschitz condition. For simplicity we take these constraints to be 1 (a more general statement can be obtained easily by rescaling).

Corollary 4.5. *Let Q be a distribution over k-clauses of complexity r. For $p \in [1,2]$, let $K \subseteq \mathcal{B}_p^N(1)$ be convex and compact set and $\mathcal{F}_p = \{f(\cdot) \mid \forall x \in K, \|\nabla f(x)\|_q \leq 1\}$. Assume that there exists a mapping that maps each k-clause $C \in X_k$ to a convex function $f_C \in \mathcal{F}_p$. Further assume that for some $\epsilon > 0$ and $\alpha \in \mathbb{R}$,*

$$\Pr_{\sigma \in \{\pm 1\}^n}\left[\min_{x \in K}\left\{\mathop{\mathbb{E}}_{C \sim Q_\sigma}[f_C(x)]\right\} \leq \alpha\right] \geq 1/2.$$

and

$$\min_{x \in K}\left\{\mathop{\mathbb{E}}_{C \sim U_k}[f_C(x)]\right\} > \alpha + \epsilon.$$

Then $N = 2^{\tilde{\Omega}_k\left(n \cdot \epsilon^{2/r}\right)}$.

For instances of planted satisfiability there is a constant gap between the fraction of clauses that can be satisfied in a formula sampled from U_k and the fraction of clauses that can be satisfied in a formula sampled from Q_σ. Thus, for convex relaxations that satisfy the conditions of Corollary 4.5 the lower bounds imply a large integrality gap.

Note that these corollaries give concrete lower bounds on the dimension and other structural properties of convex programs that can be used to solve an average-case k-CSP without any assumptions about how the convex program is solved. In particular, it does not need to be solved via a statistical algorithm or even computationally efficiently. As far as we know, this approach to obtaining lower bounds for convex relaxations from convex optimization algorithms and statistical query lower bounds is new.

Remark 4.6. *We observe that standard lift-and-project procedures (Sherali-Adams, Lovász-Schrijver, Lasserre) for strengthening LP/SDP formulations do not affect the analysis above. While these procedures add a large number of auxiliary variables and constraints the resulting program is still a convex optimization problem in the same dimension (although implementation of the separation oracle becomes more computationally intensive). Hence the use of such procedures does not necessarily affect the bounds on the number of queries and tolerance we gave above.*

At a more conceptual level, the primary difference between the commonly considered hierarchies of LP/SDP relaxations and our approach is as follows. The expected objective value of the stochastic convex programs corresponding to these hierarchies of relaxations captures the expected objective of the original Boolean k-CSP. Yet, solving stochastic convex programs corresponding to these relaxations for all distributions requires $\Omega(n^{r/2})$ samples (information theoretically). Lower bounds against such relaxations effectively prove that this number of samples is necessary even for the uniform distribution over the clauses: given fewer samples the optimum of the objective based on the given random samples will have a much lower value than the optimum of the expected objective (a phenomenon that is referred to as overfitting). In contrast, our approach rules out relaxations for which the resulting stochastic convex program can be solved by a statistical query algorithm using q queries to VSTAT $\left(\frac{n^r}{(\log q)^r}\right)$. In particular there is no overfitting. However, such relaxations end up not being sufficiently expressive: the optimum of the expected objective of the relaxation does not differentiate between the planted distributions and the uniform one. This difference makes our lower bounds incomparable and, in a way, complementary to existing work on lower bounds for specific hierarchies of convex relaxations.

5 Statistical Dimension of Planted Satisfiability

In this section, we prove our lower bound on the statistical dimension with discrimination norm of the planted satisfiability problem (Theorem 3.5). Recall that the theorem states that for any distribution Q over k-clauses of distributional complexity r, there exists a constant $c > 0$ such that for any $q \geq 1$,

$$\mathrm{SDN}\left(\mathcal{B}(\mathcal{D}_Q, U_k), \frac{c(\log q)^{r/2}}{n^{r/2}}\right) \geq q.$$

Proof overview: We first show that the discrimination operator corresponding to Q applied to a function $h : X_k \to \mathbb{R}$ can be decomposed into a linear combination of discrimination operators for ℓ-XOR-SAT. Namely, we show that

$$\mathop{\mathbb{E}}_{Q_\sigma}[h] - \mathop{\mathbb{E}}_{U_k}[h] = -2^k \sum_{S \subseteq [k]} \hat{Q}(S) \cdot (\mathop{\mathbb{E}}_{Z_{\ell,\sigma}}[h_S] - \mathop{\mathbb{E}}_{U_\ell}[h_S]),$$

where $Z_{\ell,\sigma}$ is the ℓ-XOR-SAT distribution over ℓ-clauses with planted assignment σ, and h_S is a projection of h to X_ℓ defined below.

The two key properties of this decomposition are: (i) the coefficients obtained in the decomposition are exactly $\hat{Q}(S)$'s, which determine the distribution complexity of Q, and (ii) $\|h_S\|_{U_\ell}$ is upper-bounded by $\|h\|_{U_k}$. This step implies that the discrimination norm for the problem defined by Q is upper bounded (up to constant factors) by the discrimination norm for $r(Q)$-XOR-SAT.

In the second step of the proof we bound the discrimination norm for the $r(Q)$-XOR-SAT problem. Our analysis is based on the observation that $\mathbb{E}_{Z_{\ell,\sigma}}[h_S] - \mathbb{E}_{U_\ell}[h_S]$ is a degree-ℓ polynomial as a function of σ. We exploit known concentration properties of degree-ℓ polynomials to show that the function cannot have high expectation over a large subset of assignments. This gives the desired bound on the discrimination norm for the $r(Q)$-XOR-SAT problem.

We now give the formal details of the proof. For a distribution Q_σ and query function $h : X_k \to \mathbb{R}$, we denote by $\Delta(\sigma, h) = \mathbb{E}_{Q_\sigma}[h] - \mathbb{E}_{U_k}[h]$. We start by introducing some notation:

Definition 5.1. *For $\ell \in [k]$,*

- *Let Z_ℓ be the ℓ-XOR-SAT distribution over $\{\pm 1\}^\ell$, that is a distribution such that $Z_\ell(i) = 1/2^{\ell-1}$ if i is odd and 0 otherwise.*
- *For a clause $C \in X_k$ and $S \subseteq [k]$ of size ℓ, let $C_{|S}$ denote a clause in X_ℓ consisting of literals of C at positions with indices in S (in the order of indices in S).*
- *For $h : X_k \to \mathbb{R}$, $S \subseteq [k]$ of size ℓ and $C_\ell \in X_\ell$, let*

$$h_S(C_\ell) = \frac{|X_\ell|}{|X_k|} \sum_{C \in X_k,\ C_{|S} = C_\ell} h(C).$$

- *For $g : X_\ell \to \mathbb{R}$, let $\Gamma_\ell(\sigma, g) = \mathbb{E}_{Z_{\ell,\sigma}}[g] - \mathbb{E}_{U_\ell}[g]$.*

Recall the discrete Fourier expansion of a function $Q : \{\pm 1\}^k \to \mathbb{R}$:

$$Q(x) = \sum_{S \subseteq [k]} \hat{Q}(S)\chi_S(x),$$

where $\chi_S(x) = \prod_{i \in S} x_i$ is a parity or Walsh basis function, and the Fourier coefficient of the set S is defined as:

$$\hat{Q}(S) = \frac{1}{2^k} \sum_{y \in \{\pm 1\}^k} Q(y)\chi_S(y)$$

We show that $\Delta(\sigma, h)$ (as a function of h) can be decomposed into a linear combination of $\Gamma_\ell(\sigma, h_S)$.

Lemma 5.2. *For every σ in $\{\pm 1\}^n$ and $h : X_k \to \mathbb{R}$,*

$$\Delta(\sigma, h) = -2^k \sum_{S \subseteq [k], S \neq \emptyset} \hat{Q}(S) \cdot \Gamma_\ell(\sigma, h_S).$$

Proof. Recall that for a clause C we denote by $\sigma(C)$ the vector in $\{\pm 1\}^k$ that gives evaluation of the literals in C on σ with -1 corresponding to TRUE and 1 to FALSE. Also by our definitions, $Q_\sigma(C) = \frac{2^k \cdot Q(\sigma(C))}{|X_k|}$. Now, using ℓ to denote $|S|$,

$$\begin{aligned}
\mathop{\mathbb{E}}_{Q_\sigma}[h] &= \sum_{C \in X_k} h(C) \cdot Q_\sigma(C) = \frac{2^k}{|X_k|} \sum_{C \in X_k} h(C) \cdot Q(\sigma(C)) \\
&= \frac{2^k}{|X_k|} \sum_{S \subseteq [k]} \hat{Q}(S) \sum_{C \in X_k} \chi_S(\sigma(C)) \cdot h(C) \\
&= \frac{2^k}{|X_k|} \sum_{S \subseteq [k]} \hat{Q}(S) \sum_{C_\ell \in X_\ell} \sum_{C \in X_k, C_{|S} = C_\ell} \chi_S(\sigma(C)) \cdot h(C) \qquad (2)
\end{aligned}$$

Note that if $C_{|S} = C_\ell$ then for $\ell \geq 1$,

$$\chi_S(\sigma(C)) = \chi_{[\ell]}(\sigma(C_\ell)) = 1 - 2^\ell \cdot Z_\ell(\sigma(C_\ell))$$

and for $\ell = 0$, $\chi_\emptyset(\sigma(C)) = 1$. Therefore, for $\ell \geq 1$ and $\ell = 0$ respectively,

$$\sum_{C \in X_k, C_{|S} = C_\ell} \chi_S(\sigma(C)) \cdot h(C) = (1 - 2^\ell \cdot Z_\ell(\sigma(C_\ell))) \cdot \sum_{C \in X_k, C_{|S} = C_\ell} h(C) \text{ and}$$

$$\frac{2^k}{|X_k|} \sum_{C \in X_k} [\hat{Q}(\emptyset) h(C)] = 2^k \cdot \hat{Q}(\emptyset) \cdot \mathop{\mathbb{E}}_{U_k}[h(C)] = \mathop{\mathbb{E}}_{U_k}[h(C)],$$

where $\hat{Q}(\emptyset) = 2^{-k}$ follows from Q being a distribution over $\{\pm 1\}^k$. Plugging this into eq.(2) we obtain

$$\begin{aligned}
\Delta(\sigma, h) &= \mathop{\mathbb{E}}_{Q_\sigma}[h] - \mathop{\mathbb{E}}_{U_k}[h] \\
&= \frac{2^k}{|X_k|} \sum_{S \subseteq [k], S \neq \emptyset} \hat{Q}(S) \sum_{C_\ell \in X_\ell} \left[(1 - 2^\ell \cdot Z_\ell(\sigma(C_\ell))) \cdot \sum_{C \in X_k, C_{|S} = C_\ell} h(C) \right] \\
&= \sum_{S \subseteq [k], S \neq \emptyset} \frac{2^k}{|X_\ell|} \hat{Q}(S) \sum_{C_\ell \in X_\ell} \left[(1 - 2^\ell \cdot Z_\ell(\sigma(C_\ell))) \cdot h_S(C_\ell) \right] \\
&= 2^k \sum_{S \subseteq [k], S \neq \emptyset} \hat{Q}(S) \left(\mathop{\mathbb{E}}_{U_\ell}[h_S] - \mathop{\mathbb{E}}_{Z_{\ell,\sigma}}[h_S] \right) \\
&= -2^k \sum_{S \subseteq [k], S \neq \emptyset} \hat{Q}(S) \cdot \Gamma_\ell(\sigma, h_S),
\end{aligned}$$

where we used that, by definition of $Z_{\ell,\sigma}$, $\frac{1}{|X_\ell|} \cdot 2^\ell \cdot Z_\ell(\sigma(C_\ell)) = Z_{\ell,\sigma}(C_\ell)$. □

We now analyze $\Gamma_\ell(\sigma, h_S)$. For a clause C let $I(C)$ denote the set of indices of variables in the clause C and let $\overline{\#}(C)$ denote the number of negated variables is C. Then, by definition,

$$Z_{\ell,\sigma}(C) = \frac{Z_\ell(\sigma(C))}{|X_\ell|} = \frac{1 - (-1)^{\overline{\#}(C)} \cdot \chi_{I(C)}(\sigma)}{|X_\ell|}.$$

This implies that $\Gamma_\ell(\sigma, h_S)$ can be represented as a linear combination of parities of length ℓ.

Lemma 5.3. *For $g : X_\ell \to \mathbb{R}$,*

$$\Gamma_\ell(\sigma, g) = -\frac{1}{|X_\ell|} \sum_{A \subseteq [n], |A| = \ell} \left(\sum_{C_\ell \in X_\ell, I(C_\ell) = A} g(C_\ell) \cdot (-1)^{\overline{\#}(C_\ell)} \right) \cdot \chi_A(\sigma).$$

Proof.

$$\begin{aligned}
\Gamma_\ell(\sigma, g) &= \mathop{\mathbb{E}}_{Z_{\ell,\sigma}}[g] - \mathop{\mathbb{E}}_{U_\ell}[g] \\
&= -\frac{1}{|X_\ell|} \sum_{C_\ell \in X_\ell} g(C_\ell) \cdot (-1)^{\overline{\#}(C_\ell)} \cdot \chi_{I(C_\ell)}(\sigma) \\
&= -\frac{1}{|X_\ell|} \sum_{A \subseteq [n], |A| = \ell} \left(\sum_{C_\ell \in X_\ell, I(C_\ell) = A} g(C_\ell) \cdot (-1)^{\overline{\#}(C_\ell)} \right) \cdot \chi_A(\sigma).
\end{aligned}$$

□

For $\mathcal{S} \subseteq \{\pm 1\}^n$ we now bound $\mathbb{E}_{\sigma \sim \mathcal{S}}[|\Gamma_\ell(\sigma, g)|]$ by exploiting its concentration properties as a degree-ℓ polynomial. To do this, we will need the following concentration bound for polynomials on $\{\pm 1\}^n$. It can be easily derived from the hypercontractivity results of Bonami and Beckner [Bon70, Bec75] as done for example in [Jan97, DFKO07].

Lemma 5.4. *Let $p(x)$ be a degree ℓ polynomial over $\{\pm 1\}^n$. Then there is constant c such that for all $t > 0$,*

$$\Pr_{x \sim \{\pm 1\}^n} [|p(x)| \geq t \|p\|_2] \leq 2 \cdot \exp(-c\ell \cdot t^{2/\ell}),$$

where $\|p\|_2$ is defined as $(\mathbb{E}_{x \sim \{\pm 1\}^n}[p(x)^2])^{1/2}$.

In addition we will use the following simple way to convert strong concentration to a bound on expectation over subsets of assignments.

Lemma 5.5. *Let $p(x)$ be a degree $\ell \geq 1$ polynomial over $\{\pm 1\}^n$, let $\mathcal{S} \subseteq \{\pm 1\}^n$ be a set of assignments for which $d = 2^n/|\mathcal{S}| \geq e^\ell$. Then $\mathbb{E}_{\sigma \sim \mathcal{S}}[|p(\sigma)|] \leq 2(\ln d/(c\ell))^{\ell/2} \cdot \|p\|_2$, where c is the constant from Lemma 5.4.*

Proof. Let $c_0 = \ell \cdot c$. By Lemma 5.4 we have that for any $t > 0$,

$$\Pr_{x \sim \{\pm 1\}^n} [|p(x)| \geq t \|p\|_2] \leq 2 \cdot \exp(-c_0 \cdot t^{2/\ell}).$$

The set $\mathcal{S}$ contains $1/d$ fraction of points in $\{\pm 1\}^n$ and therefore

$$\Pr_{x \sim \mathcal{S}} [|p(x)| \geq t \|p\|_2] \leq 2 \cdot d \cdot \exp(-c_0 \cdot t^{2/\ell}).$$

For any random variable Y and value $a \in \mathbb{R}$,

$$\mathbb{E}[Y] \leq a + \int_a^\infty \Pr[Y \geq t] dt.$$

Therefore, for $Y = |p(\sigma)|/\|p\|_2$ and $a = (\ln d/c_0)^{\ell/2}$ we obtain

$$\begin{aligned}
\frac{\mathbb{E}_{\sigma \sim \mathcal{S}}[|p(\sigma)|]}{\|p\|_2} &\leq (\ln d/c_0)^{\ell/2} + \int_{(\ln d/c_0)^{\ell/2}}^\infty d \cdot e^{-c_0 t^{2/\ell}} dt = (\ln d/c_0)^{\ell/2} + \frac{\ell \cdot d}{2 \cdot c_0^{\ell/2}} \cdot \int_{\ln d}^\infty e^{-z} z^{\ell/2-1} dz \\
&= (\ln d/c_0)^{\ell/2} + \frac{\ell \cdot d}{2 \cdot c_0^{\ell/2}} \cdot \left(-e^{-z} z^{\ell/2-1} \right) \Big|_{\ln d}^\infty + (\ell/2 - 1) \int_{\ln d}^\infty e^{-z} z^{\ell/2-2} dz = \ldots \\
&\leq (\ln d/c_0)^{\ell/2} + \frac{\ell \cdot d}{2 \cdot c_0^{\ell/2}} \sum_{\ell'=1/2}^{\lceil \ell/2 \rceil - 1} \left(-\frac{\lceil \ell/2 \rceil !}{\ell' !} e^{-z} z^{\ell'} \right) \Big|_{\ln d}^\infty \\
&= (\ln d/c_0)^{\ell/2} + \frac{1}{2 \cdot c_0^{\ell/2}} \sum_{\ell'=0}^{\lceil \ell/2 \rceil - 1} \frac{\lceil \ell/2 \rceil !}{\ell' !} (\ln d)^{\ell'} \leq 2 (\ln d/c_0)^{\ell/2},
\end{aligned}$$

where we used the condition $d \geq e^\ell$ to obtain the last inequality. □

We can now use the fact that $\Gamma_\ell(\sigma, g)$ is a degree-ℓ polynomial of σ to prove the following lemma:

Lemma 5.6. *Let $\mathcal{S} \subseteq \{\pm 1\}^n$ be a set of assignments for which $d = 2^n/|\mathcal{S}|$. Then*

$$\mathop{\mathbb{E}}_{\sigma \sim \mathcal{S}} [|\Gamma_\ell(\sigma, g)|] = O_\ell \left((\ln d)^{\ell/2} \cdot \|g\|_2 / \sqrt{|X_\ell|} \right),$$

where $\|g\|_2 = \sqrt{\mathbb{E}_{U_\ell}[g(C_\ell)^2]}$.

Proof. By Lemma 5.5 we get that

$$\mathop{\mathbb{E}}_{\sigma\sim\mathcal{S}}[|\Gamma_\ell(\sigma,g)|] \le 2(\ln d/(c\ell))^{\ell/2}\cdot\|\Gamma_{\ell,g}\|_2,$$

where $\Gamma_{\ell,g}(\sigma)\equiv\Gamma_\ell(\sigma,g)$. Now, by Parseval's identity and Lemma 5.3 we get that

$$\begin{aligned}\mathop{\mathbb{E}}_{\sigma\sim\{\pm1\}^n}\left[\Gamma_{\ell,g}(\sigma)^2\right] &= \sum_{A\subseteq[n]}\widehat{\Gamma_{\ell,g}}(A)^2\\ &= \frac{1}{|X_\ell|^2}\sum_{A\subseteq[n],|A|=\ell}\left(\sum_{C_\ell\in X_\ell,I(C_\ell)=A} g(C_\ell)\cdot(-1)^{\overline{\#}(C_\ell)}\right)^2\\ &\le \frac{1}{|X_\ell|^2}\sum_{A\subseteq[n],|A|=\ell}|\{C_\ell\mid I(C_\ell)=A\}|\cdot\left(\sum_{C_\ell\in X_\ell,I(C_\ell)=A} g(C_\ell)^2\right)\\ &= \frac{2^\ell\ell!}{|X_\ell|^2}\sum_{C_\ell\in X_\ell} g(C_\ell)^2 = \frac{2^\ell\ell!}{|X_\ell|}\mathop{\mathbb{E}}_{U_\ell}[g(C_\ell)^2].\end{aligned}$$

□

We are now ready to bound the discrimination norm.

Lemma 5.7. *Let Q be a clause distribution of the distributional complexity $r = r(Q)$, let $\mathcal{D}'\subseteq\{Q_\sigma\}_{\sigma\in\{\pm1\}^n}$ be a set of distributions over clauses and $d = 2^n/|\mathcal{D}'|$. Then $\kappa_2(\mathcal{D}',U_k) = O_k\left((\ln d/n)^{r/2}\right)$.*

Proof. Let $\mathcal{S} = \{\sigma\mid Q_\sigma\in\mathcal{D}'\}$ and let $h : X_k\to\mathbb{R}$ be any function such that $\mathbb{E}_{U_k}[h^2] = 1$. Let ℓ denote $|S|$. Using Lemma 5.2 and the definition of r,

$$|\Delta(\sigma,h)| = 2^k\cdot\left|\sum_{S\subseteq[k]\setminus\{0\}}\hat{Q}(S)\cdot\Gamma_\ell(\sigma,h_S)\right| \le 2^k\cdot\sum_{S\subseteq[k],\ell=|S|\ge r}\left|\hat{Q}(S)\right|\cdot|\Gamma_\ell(\sigma,h_S)|.$$

Hence, by Lemma 5.6 we get that,

$$\mathop{\mathbb{E}}_{\sigma\sim\mathcal{S}}[|\Delta(\sigma,h)|] \le 2^k\cdot\sum_{S\subseteq[k],\ |S|\ge r}\left|\hat{Q}(S)\right|\cdot\mathop{\mathbb{E}}_{\sigma\sim\mathcal{S}}[|\Gamma_\ell(\sigma,h_S)|] = O_k\left(\sum_{S\subseteq[k],\ |S|\ge r}\frac{(\ln d)^{\ell/2}\cdot\|h_S\|_2}{\sqrt{|X_\ell|}}\right) \quad (3)$$

By the definition of h_S,

$$\begin{aligned}\|h_S\|_2^2 &= \mathop{\mathbb{E}}_{U_\ell}[h_S(C_\ell)^2]\\ &= \frac{|X_\ell|^2}{|X_k|^2}\cdot\mathop{\mathbb{E}}_{U_\ell}\left[\left(\sum_{C\in X_k,\ C_{|S}=C_\ell} h(C)\right)^2\right]\\ &\le \frac{|X_\ell|^2}{|X_k|^2}\cdot\mathop{\mathbb{E}}_{U_\ell}\left[\frac{|X_k|}{|X_\ell|}\cdot\left(\sum_{C\in X_k,\ C_{|S}=C_\ell} h(C)^2\right)\right]\\ &= \mathop{\mathbb{E}}_{U_k}[h(C)^2] = \|h\|_2^2 = 1,\end{aligned}$$

where we used Cauchy-Schwartz inequality together with the fact that for any C_ℓ,

$$\left|\{C \in X_k \mid C_{|S} = C_\ell\}\right| = \frac{|X_k|}{|X_\ell|}.$$

By plugging this into eq.(3) and using the fact that $\ln d < n$ we get,

$$\mathop{\mathbb{E}}_{\sigma\sim\mathcal{S}}[|\Delta(\sigma,h)|] = O_k\left(\sum_{\ell\geq r}\frac{(\ln d)^{\ell/2}}{\sqrt{2^\ell \cdot n!/(n-\ell)!}}\right) = O_k\left(\frac{(\ln d)^{r/2}}{n^{r/2}}\right).$$

By the definition of $\kappa_2(\mathcal{D}', U_k)$ we obtain the claim. □

We are now ready to finish the proof of our bound on SDN.

Proof. (of Theorem 3.5) Our reference distribution is the uniform distribution U_k and the set of distributions $\mathcal{D} = \mathcal{D}_Q = \{Q_\sigma\}_{\sigma\in\{\pm1\}^n}$ is the set of distributions for all possible assignments. Let $\mathcal{D}' \subseteq \mathcal{D}$ be a set of distributions of size $|\mathcal{D}|/q$ and $\mathcal{S} = \{\sigma \mid Q_\sigma \in \mathcal{D}'\}$. Then, by Lemma 5.7, we get

$$\kappa_2(\mathcal{D}', U_k) = O_k\left(\frac{(\ln q)^{r/2}}{n^{r/2}}\right).$$

By the definition of SDN, this implies the claim. □

6 Planted k-CSPs

While the focus of our presentation is on planted satisfiability problems, the techniques can be applied to other models of planted constraint satisfaction. Here we describe describe how to apply our techniques to prove essentially identical lower bounds for the planted k-CSP problem. Recall that our generalization of this planted k-CSP problem is defined by a function $P : \{\pm1\}^k \to [-1,1]$ and we are given access to samples from a distribution P_σ, where σ is a planted assignment in $\{\pm1\}^n$. A random sample from this distribution is a randomly and uniformly chosen ordered k-tuple of variables (without repetition) $x_{i_1}, \ldots, x_{i_k}$ together with a randomly and independently chosen value $b \in \{-1,1\}$ such that $\mathbb{E}[b] = P(\sigma_{i_1}, \ldots, \sigma_{i_k})$ (or $\Pr[b = 1] = (1 + P(\sigma_{i_1}, \ldots, \sigma_{i_k}))/2$). This captures the important special case when P is a Boolean predicate.

Before going into the proof of the lower bound for this model we show two additional connections between this model and our planted satisfiability model. First we show that planted satisfiability can be easily reduced to the planted k-CSP above while preserving the complexity parameter (we remark that the reduction will always produce a non-boolean P and hence requires our generalization). The second connection is that both of these models can be seen as special cases of a more general model of planted constraint satisfaction introduced by Abbe and Montanari [AM15].

To describe the first reduction we start with some notation. Let Y_k denote the set of all k-tuples of variables without repetition and let $X'_k = Y_k \times \{-1,1\}$. For a function $P : \{\pm1\}^k \to \mathbb{R}$ we use $r(P)$ to denote the degree of the lowest-degree non-zero Fourier coefficient of P and refer to it as the complexity of P. For a clause $C = (l_1, \ldots, l_k) \in X_k$ we denote by $v(C)$ the k-tuple of variables in C (in the same order). For $j \in [k]$ let s_j be the sign of literal l_j (with 1 meaning not negated and -1 meaning negated) and let $s(C) = s_1, \ldots, s_k$. We use $\mathbf{1}_k$ to denote the k-dimensional vector $(1,1,\ldots,1)$.

Lemma 6.1. *There exists an algorithm that for every distribution Q over $\{\pm 1\}^k$ of complexity r, and any $\sigma \in \{\pm 1\}^n$, given a random sample distributed according to Q_σ outputs a random sample distributed according to P_σ, where $P \equiv Q - 2^{-k}$. Further, $r(Q) = r(P)$.*

Proof. Given a random clause C the algorithm outputs the tuple of variables $v(C)$ together with a bit b chosen according to the following rule. With probability 1/2: if $s(C) = \mathbf{1}_k$ then output 1, otherwise -1; with probability $1/2 - 2^{-k-1}$ output 1 and -1 with probability 2^{-k-1}.

Let us analyze the resulting distribution. First we note that the output distribution is uniform over Y_k. This follows from the fact that for every $u \in Y_k$, $\sum_{v(C)=u} Q_\sigma(C) = \sum_{y \in \{\pm 1\}^k} Q(y) = 1$. We now evaluate the expectation of the bit b produced by our reduction as a function of $\sigma(u)$ (the values assigned by σ to variables in u). From the definition of Q_σ, for every $u \in Y_k$ and $z \in \{\pm 1\}^k$,

$$\Pr_{C \sim Q_\sigma}[s(C) = z \mid v(C) = u] = Q(\sigma(C)) = Q(\sigma(u) \circ z), \tag{4}$$

where we use $\circ$ to denote the element-wise product of two vectors. In particular, $\Pr_{C \sim Q_\sigma}[s(C) = \mathbf{1}_k \mid v(C) = u] = Q(\sigma(u))$. This means that

$$\mathbb{E}[b] = \frac{1}{2}(Q(\sigma(u)) - (1 - Q(\sigma(u)) + \left(\frac{1}{2} - 2^{-k-1}\right) - 2^{-k-1} = Q(\sigma(u)) - 2^{-k}.$$

This means that the reduction produces a random sample from P_σ for $P(y) \equiv Q(y) - 2^{-k}$. Note that $\hat{P}(\emptyset) = 0$ and hence this reduction satisfies $r(Q) = r(P)$.

□

We now show how both of these models can be seen as special cases of the model in [AM15]. The model is specified by a collection of distributions $\{\Phi(\cdot \mid y)\}_{y \in \{\pm 1\}^k}$ over some output alphabet Z. For a planted assignment $\sigma \in \{\pm 1\}^n$ (their model allows a more general alphabet for each variable but $\{\pm 1\}$ suffices to subsume the models discussed in this paper) the planted distribution Φ_σ is defined as follows. A random sample from this distribution is a randomly and uniformly chosen ordered k-tuple of variables $u \in Y_k$ together with value z chosen randomly and independently according to $\Phi(\cdot \mid \sigma(u))$. We first observe that for any $P : \{\pm 1\}^k \to [-1, 1]$, setting $Z = \{\pm 1\}$ and having $\Phi(b \mid y) = (1 + b \cdot P(y))/2$ recovers exactly the generalized Goldreich's planted k-CSP for function P.

To recover the planted satisfiability model for distribution Q, we let $Z = \{\pm 1\}^k$ and then define $\Phi(z \mid y) = Q(y \circ z)$. Here the output alphabet represents the negation signs of variables. A k-tuple of variables $u \in Y_k$ with k negation signs $z \in \{\pm 1\}^k$ uniquely describes a clause $C \in X_k$ such that $v(C) = u$ and $s(C) = z$. Further, by Eqn. (4), we get that Φ_σ for Φ defined as above is exactly Q_σ. It is not hard to see that the techniques in this work can also be applied to characterize the SQ complexity of solving planted k-CSPs in this more general model.

6.1 Lower Bounds for Planted k-CSPs

We prove the analogue of Theorem 3.5 for the planted k-CPS, which, in turn, immediately implies that the lower bounds stated in Theorem 3.1 apply to this problem verbatim. We first note that the reduction in Lemma 6.1 implies the desired lower bound for all functions P such that $P \equiv Q - 2^{-k}$ for some distribution Q over $\{\pm 1\}^k$. Unfortunately, this is not sufficient to obtain a lower bound for all functions $P : \{\pm 1\}^{-k} \to [-1, 1]$. Indeed, this does not give a lower bound for any Boolean P. At the same time, we show that the reduction in Lemma 6.1 can be used to reduce bounds on the discrimination norm of the planted k-CSP problem to the bounds on the discrimination norm for planted satisfiability that we gave in Section 5[3]. We are not aware of similar reductions in the literature and our technique might be useful for relating the complexity of other problems for which standard reductions are not known.

We now give the formal details. Let $P : \{\pm 1\}^k \to [-1,1]$ be a function on k-bits. Let $\mathcal{D}_P$ denote the set of all distributions P_σ, where $\sigma \in \{\pm 1\}^n$ and U'_k be the uniform distribution over $X'_k = Y_k \times \{-1,1\}$. Let $\mathcal{B}(\mathcal{D}_P, U'_k)$ denote the decision problem in which given samples from an unknown input distribution $D \in \mathcal{D}_P \cup \{U'_k\}$ the goal is to output 1 if $D \in \mathcal{D}_P$ and 0 if $D = U'_k$. Our goal is to prove the following results.

Theorem 6.2. *For any function $P : \{\pm 1\}^k \to [-1,1]$ of complexity r, there exist a constant $c > 0$ (that depends on P) such that for any $q \geq 1$,*

$$\mathrm{SDN}\left(\mathcal{B}(\mathcal{D}_P, U'_k), \frac{c(\log q)^{r/2}}{n^{r/2}}\right) \geq q.$$

As in the case of Theorem 3.5, it suffices to prove the following analogue of Lemma 5.7.

Lemma 6.3. *Let $P : \{\pm 1\}^k \to [-1,1]$ be any function of complexity $r = r(P)$, let $\mathcal{D}' \subseteq \{P_\sigma\}_{\sigma \in \{\pm 1\}^n}$ be a set of distributions over clauses and $d = 2^n/|\mathcal{D}'|$. Then $\kappa_2(\mathcal{D}', U'_k) = O_k\left((\ln d/n)^{r/2}\right)$.*

Proof. We first note that this bound does not say anything non-trivial for $r = 0$ (and, indeed, the label distribution is biased in this case and can be distinguished from U'_k using a constant number of samples). Therefore, from now on we assume that $r \geq 1$. Let $\mathcal{S} = \{\sigma \mid P_\sigma \in \mathcal{D}'\}$ and let $h' : X'_k \to \mathbb{R}$ be any function such that $\|h'\|_{U'_k} = 1$ and $\mathbb{E}_{\sigma\sim\mathcal{S}}\left[\left|\mathbb{E}_{P_\sigma}[h'] - \mathbb{E}_{U'_k}[h']\right|\right] = \kappa_2(\mathcal{D}', U'_k)$. We define a function h on X_k as follows. If for $C \in X_k$, $s(C) = \mathbf{1}_k$ then $h(C) = h'(v(C), 1)$, otherwise $h(C) = h'(v(C), -1)$. We now claim that for every $\sigma \in \{\pm 1\}^n$,

$$\mathop{\mathbb{E}}_{P_\sigma}[h'] - \mathop{\mathbb{E}}_{U'_k}[h'] = 2^{2k-1} \cdot \left(\mathop{\mathbb{E}}_{Q_\sigma}[h] - \mathop{\mathbb{E}}_{U_k}[h]\right), \tag{5}$$

where $Q \equiv (P+1)/2^k$. Note that Q defined in this way is a distribution since for all $y \in \{\pm 1\}^k$, $Q(y) \geq 0$ and $\sum_{y\in\{\pm 1\}^k} Q(y) = 2^k \cdot \hat{P}(\emptyset) + 1 = 1$.

Distributions Q_σ,P_σ U_k and U'_k are uniform over k-tuples of variables and therefore to prove eq. (5), it suffices to prove that for every $u \in Y_k$,

$$\begin{aligned}\mathop{\mathbb{E}}_{(v,b)\sim P_\sigma}[h'(v,b) \mid v = u] - \mathop{\mathbb{E}}_{(v,b)\sim U'_k}[h'(v,b) \mid v = u]& \\ = 2^{2k-1} \cdot \left(\mathop{\mathbb{E}}_{C\sim Q_\sigma}[h(C) \mid v(C) = u] - \mathop{\mathbb{E}}_{C\sim U_k}[h(C) \mid v(C) = u]\right)&. \end{aligned} \tag{6}$$

The left hand side of this equality is equal to

$$h'(u,1)\cdot\frac{1+P(\sigma(u))}{2} + h'(u,-1)\cdot\frac{1-P(\sigma(u))}{2} - \frac{1}{2}\cdot h'(u,1) + \frac{1}{2}\cdot h'(u,-1) = \frac{P(\sigma(u))\cdot(h'(u,1) - h'(u,-1))}{2}.$$

[3] A direct proof of this bound can be found in an earlier version of this work [FPV13, v5].

By equation (4), the right side of eq. 6 is equal to

$$\begin{aligned}
&2^{2k-1}\cdot 2^{-k}\cdot \sum_{C,v(C)=u} h(C)\cdot(Q(C)-1)) = \\
&\quad = 2^{k-1}\cdot\left(h'(u,1)\cdot\frac{P(\sigma(u))}{2^k}+\sum_{C,v(C)=u,s(C)\neq \mathbf{1}_k} h'(u,-1)\cdot\frac{P(\sigma(u)\circ s(C))}{2^k}\right)\\
&\quad = \frac{1}{2}\cdot\big(h'(u,1)\cdot P(\sigma(u)) - h'(u,-1)\cdot P(\sigma(u))\big) = \frac{P(\sigma(u))\cdot(h'(u,1)-h'(u,-1))}{2},
\end{aligned}$$

where we used the fact that $\sum_{y\in\{\pm1\}^k} P(y) = 2^k\cdot\hat{P}(\emptyset)=0$ to obtain the equality of the second line to the third one.

Now all we need to bound $\kappa_2(\mathcal{D}',U'_k)$ is an upper bound on $\|h\|_{U_k}$. First, note that by our assumption,

$$\mathop{\mathbb{E}}_{U'_k}[h'^2] = \frac{1}{|Y_k|}\cdot\sum_{u\in Y_k}\frac{1}{2}\left(h'(u,1)^2+h'(u,-1)^2\right)=1. \tag{7}$$

For every $v\in Y_k$,

$$\mathop{\mathbb{E}}_{C\sim U_k}[(h(C))^2 \mid v(C)=u] = \frac{1}{2^k}\cdot\Big(h'(u,1)^2+(2^k-1)h'(u,-1)^2\Big)\geq\frac{1}{2^k}\left(h'(u,1)^2+h'(u,-1)^2\right).$$

Using eq. (7) we get that,

$$\|h\|^2_{U_k}\geq\frac{1}{|Y_k|}\cdot\frac{1}{2^k}\cdot\sum_{u\in Y_k}\left(h'(u,1)^2+h'(u,-1)^2\right)=\frac{1}{2^{k-1}}.$$

Using this bound on the norm and eq. (5) we can now bound $\kappa_2(\mathcal{D}',U'_k)$ as follows. Let $\mathcal{D}'_Q \doteq \{Q_\sigma \mid \sigma\in\mathcal{S}\}$.

$$\begin{aligned}
\kappa_2(\mathcal{D}',U'_k) &= \mathop{\mathbb{E}}_{\sigma\sim\mathcal{S}}\left[\left|\mathop{\mathbb{E}}_{P_\sigma}[h']-\mathop{\mathbb{E}}_{U'_k}[h']\right|\right] = 2^{2k-1}\cdot\mathop{\mathbb{E}}_{\sigma\sim\mathcal{S}}\left[\left|\mathop{\mathbb{E}}_{Q_\sigma}[h]-\mathop{\mathbb{E}}_{U_k}[h]\right|\right]\\
&\leq 2^{2k-1}\cdot\frac{\kappa_2(\mathcal{D}'_Q,U_k)}{\|h\|_{U_k}}\leq 2^{2k-1+k/2-1/2}\cdot\kappa_2(\mathcal{D}'_Q,U_k)=O_k\left((\ln d/n)^{r/2}\right),
\end{aligned}$$

where we used Lemma 5.7 to obtain the last bound. □

7 Lower Bounds using Statistical Dimension

7.1 Lower bound for VSTAT

We first prove an analogue of lower-bound for VSTAT from [FGR+12] but using the statistical dimension based on discrimination norm instead of the average correlation. It is not hard to see that discrimination norm is upper-bounded by the square root of average correlation and therefore our result subsumes the one in [FGR+12].

Theorem 7.1. *Let X be a domain and $\mathcal{B}(\mathcal{D}, D)$ be a decision problem over a class of distributions $\mathcal{D}$ on X and reference distribution D. Let $d = \mathrm{SDN}(\mathcal{B}(\mathcal{D}, D), \kappa)$ and let $\mathcal{D}_D$ be a set of distributions for which the value d is attained. Consider the following average-case version of the $\mathcal{B}(\mathcal{D}, D)$ problem: the input distribution D' equals D with probability $1/2$ and D' equals a random uniform element of $\mathcal{D}_D$ with probability $1/2$. Any randomized statistical algorithm that solves $\mathcal{B}(\mathcal{D}, D)$ with success probability $\gamma > 1/2$ over the choice of D' and randomness of the algorithm requires at least $(2\gamma - 1)d$ calls to VSTAT$(1/(3\kappa^2))$.*

Proof. We prove our lower bound for any deterministic statistical algorithm and the claim for randomized algorithms follows from the fact that the success probability of a randomized algorithm is just the expectation of its success probability for a random fixing of its coins.

Let $\mathcal{A}$ be a deterministic statistical algorithm that uses q queries to VSTAT$(1/(3\kappa^2))$ to solve $\mathcal{B}(\mathcal{D}, D)$ with probability γ over a random choice of an input distribution described in the statement. Following an approach from [Fel12], we simulate $\mathcal{A}$ by answering any query $h : X \to [0, 1]$ of $\mathcal{A}$ with value $\mathbb{E}_D[h(x)]$. Let $h_1, h_2, \ldots, h_q$ be the queries asked by $\mathcal{A}$ in this simulation and let b be the output of $\mathcal{A}$. $\mathcal{A}$ is successful with probability $\gamma > 1/2$ and therefore $b = 0$, that is $\mathcal{A}$ will certainly decide that the input distribution equals to D.

Let the set $\mathcal{D}^+ \subseteq \mathcal{D}_D$ be the set of distributions on which $\mathcal{A}$ is successful (that is outputs $b = 1$) and we denote these distributions by $\{D_1, D_2, \ldots, D_m\}$. We recall that, crucially, for $\mathcal{A}$ to be considered successful it needs to be successful for any valid responses of VSTAT to $\mathcal{A}$'s queries. We note that the success probability of $\mathcal{A}$ is $\frac{1}{2} + \frac{1}{2}\frac{m}{|\mathcal{D}_D|}$ and therefore $m \geq (2\gamma - 1)|\mathcal{D}_D|$.

For every $k \leq q$, let A_k be the set of all distributions D_i such that

$$\left|\mathop{\mathbb{E}}_{D}[h_k(x)] - \mathop{\mathbb{E}}_{D_i}[h_k(x)]\right| > \tau_{i,k} \doteq \max\left\{\frac{1}{t}, \sqrt{\frac{p_{i,k}(1-p_{i,k})}{t}}\right\},$$

where we use t to denote $1/(3\kappa^2)$ and $p_{i,k}$ to denote $\mathbb{E}_{D_i}[h_k(x)]$. To prove the desired bound we first prove the following two claims:

1. $\sum_{k \leq q} |A_k| \geq m$;
2. for every k, $|A_k| \leq |\mathcal{D}_D|/d$.

Combining these two implies that $q \geq d \cdot m/|\mathcal{D}_D|$ and therefore $q \geq (2\gamma - 1)d$ giving the desired lower bound.

In the rest of the proof for conciseness we drop the subscript D from inner products and norms. To prove the first claim we assume, for the sake of contradiction, that there exists $D_i \notin \cup_{k \leq q} A_k$. Then for every $k \leq q$, $|\mathbb{E}_D[h_k(x)] - \mathbb{E}_{D_i}[h_k(x)]| \leq \tau_{i,k}$. This implies that the replies of our simulation $\mathbb{E}_D[h_k(x)]$ are within $\tau_{i,k}$ of $\mathbb{E}_{D_i}[h_k(x)]$, in other words are valid responses. However we know that for these responses $\mathcal{A}$ outputs $b = 0$ contradicting the condition that $D_i \in \mathcal{D}^+$.

To prove the second claim, suppose that for some $k \in [d]$, $|A_k| > |\mathcal{D}_D|/d$. Let $p_k = \mathbb{E}_D[h_k(x)]$ and assume that $p_k \leq 1/2$ (when $p_k > 1/2$ we just replace h_k by $1 - h_k$ in the analysis below). We will next show upper and lower bounds on the following quantity

$$\Phi = \sum_{D_i \in A_k} \left[\left|\mathop{\mathbb{E}}_{D}[h_k(x)] - \mathop{\mathbb{E}}_{D_i}[h_k(x)]\right|\right] = \sum_{D_i \in A_k} |p_k - p_{i,k}|. \tag{8}$$

By our assumption for $D_i \in A_k$, $|p_{i,k} - p_k| > \tau_{i,k} = \max\{1/t, \sqrt{p_{i,k}(1-p_{i,k})/t}\}$. If $p_{i,k} \geq 2p_k/3$ then

$$|p_k - p_{i,k}| > \sqrt{\frac{p_{i,k}(1-p_{i,k})}{t}} \geq \sqrt{\frac{\frac{2}{3}p_k \cdot \frac{1}{2}}{t}} = \sqrt{\frac{p_k}{3t}}.$$

Otherwise (when $p_{i,k} < 2p_k/3$), $p_k - p_{i,k} > p_k - 2p_k/3 = p_k/3$. We also know that $|p_{i,k} - p_k| > \tau_{i,k} \geq 1/t$ and therefore $|p_{i,k} - p_k| > \sqrt{\frac{p_k}{3t}}$. Substituting this into eq. (8) we get that

$$\Phi > |A_k| \cdot \sqrt{\frac{p_k}{3t}} = |A_k| \cdot \sqrt{p_k} \cdot \kappa. \tag{9}$$

Now, by the definition of discrimination norm and its linearity we have that

$$\sum_{D_i \in A_k} \left[\left| \mathop{\mathbb{E}}_{D}[h_k(x)] - \mathop{\mathbb{E}}_{D_i}[h_k(x)] \right| \right] = |A_k| \cdot \mathop{\mathbb{E}}_{D' \sim A_k} \left[\left| \mathop{\mathbb{E}}_{D}[h_k(x)] - \mathop{\mathbb{E}}_{D'}[h_k(x)] \right| \right] \leq |A_k| \cdot \kappa_2(A_k, D) \cdot \|h_k\|_2.$$

We note that, h_k is a $[0,1]$-valued function and therefore $\|h_k\|^2 = \mathbb{E}_D[h_k(x)^2] \leq \mathbb{E}_D[h_k(x)] = p_k$. Also by definition of SDN, $\kappa_2(A_k, D) \leq \kappa$. Therefore $\Phi \leq |A_k| \cdot \kappa \cdot \sqrt{p_k}$. This contradicts the bound on Φ in eq. (9) and hence finishes the proof of our claim. □

7.2 Lower bounds for MVSTAT and 1-MSTAT

We now describe the extension of our lower bound to MVSTAT and 1-MSTAT(L) oracles. For simplicity we state them for the worst case search problems but all these results are based on a direct simulation of an oracle using a VSTAT oracle and therefore they equivalently apply to the average-case versions of the problem defined in Theorem 7.1.

Given the lower bound VSTAT we can obtain our lower bound for MVSTAT via the following simple simulation. For conciseness we use L_0 to denote $\{0, 1, \ldots, L-1\}$.

Theorem 7.2. *Let D be the input distribution over the domain X, $t, L > 0$ be integers. For any multi-valued function $h : X \to L_0$ and any set $\mathcal{S}$ of subsets of L_0, L queries to VSTAT$(4L \cdot t)$ can be used to give a valid answer to query h with set $\mathcal{S}$ to MVSTAT(L, t).*

Proof. For $i \in L_0$ we define $h_i(x)$ as $h_i(x) = 1$ if $h(x) = i$ and 0 otherwise. Let v_i be the response of VSTAT($4L \cdot t$) on query h_i. For any $Z \subseteq L_0$,

$$\begin{aligned}
\left|\sum_{\ell \in Z} v_\ell - p_Z\right| &\leq \sum_{\ell \in Z} |v_i - p_i| \\
&\leq \sum_{\ell \in Z} \max\left\{\frac{1}{4Lt}, \sqrt{\frac{p_i(1-p_i)}{4Lt}}\right\} \\
&\leq \frac{|Z|}{4Lt} + \sum_{\ell \in Z} \sqrt{\frac{p_i(1-p_i)}{4Lt}} \\
&\leq \frac{|Z|}{4Lt} + \sqrt{|Z|} \cdot \sqrt{\frac{\sum_{\ell \in Z} p_i(1-p_i)}{4Lt}} \\
&\leq \frac{|Z|}{4Lt} + \sqrt{|Z|} \cdot \sqrt{\frac{p_Z(1-p_Z)}{4Lt}} \\
&\leq \frac{1}{4t} + \sqrt{\frac{p_Z(1-p_Z)}{4t}} \\
&\leq \max\left\{\frac{1}{t}, \sqrt{\frac{p_Z(1-p_Z)}{t}}\right\},
\end{aligned}$$

where $p_Z = \Pr_D[h(x) \in Z]$. □

We now describe our lower bound for 1-MSTAT(L) oracle.

Theorem 7.3. *Let $\mathcal{B}(\mathcal{D}, D)$ be a decision problem. For $\kappa > 0$, let $d = \mathrm{SDN}(\mathcal{B}(\mathcal{D}, D), \kappa)$. Any (possibly randomized) statistical algorithm that solves $\mathcal{B}(\mathcal{D}, D)$ with probability $\gamma > 1/2$ requires at least m calls to 1-MSTAT(L) for*

$$m = \Omega\left(\frac{1}{L} \min\left\{d(2\gamma - 1), \frac{\gamma^2}{\kappa^2}\right\}\right) .$$

In particular, any algorithm with success probability of at least $2/3$ requires at least $\Omega\left(\frac{1}{L} \cdot \min\{d, 1/\kappa^2\}\right)$ samples from 1-MSTAT(L).

The proof of this result is based on the following simulation of 1-MSTAT(L) using VSTAT.

Theorem 7.4. *Let $\mathcal{Z}$ be a search problem and let $\mathcal{A}$ be a (possibly randomized) statistical algorithm that solves $\mathcal{Z}$ with probability at least γ using m samples from 1-MSTAT(L). For any $\delta \in (0, 1/2]$, there exists a statistical algorithm $\mathcal{A}'$ that uses at most $O(m \cdot L)$ queries to VSTAT($L \cdot m/\delta^2$) and solves $\mathcal{Z}$ with probability at least $\gamma - \delta$.*

A special case of this theorem for $L = 2$ is proved in [FGR$^+$12]. Their result is easy to generalize to the statement of Theorem 7.4 but is it fairly technical. Instead we describe a simple way to simulate m samples of 1-MSTAT(L) using $O(mL)$ samples from 1-STAT. This simulation (together with the simulation of 1-STAT from [FGR$^+$12]) imply Theorem 7.4. It also allows to easily relate the powers of these oracles. The simulation is based on the following lemma (proof by Jan Vondrak).

Lemma 7.5. *Let D be the input distribution over X and let $h : X \to L_0$ be any function. Then using $L + 1$ samples from 1-STAT it is possible to output a random variable $Y \in L_0 \cup \{\perp\}$, such that*

- $\Pr[Y \neq \perp] \geq 1/(2e)$,
- *for every* $i \in L_0$, $\Pr[Y = i \mid Y \neq \perp] = p_i$.

Proof. Y is defined as follows. For every $i \in L_0$ ask a sample for h_i from 1-STAT and let B_i be equal to the outcome with probability 1/2 and 0 with probability 1/2 (independently). If the number of B_i's that are equal to 1 is different from 1 then $Y = \perp$. Otherwise let j be the index such that $B_j = 1$. Ask a sample for h_j from 1-STAT and let B'_j be the outcome with probability 1/2 and 0 with probability 1/2. If $B'_j = 0$ let $Y = j$, otherwise $Y = \perp$. From the definition of Y, we obtain that for every $i \in L_0$,

$$\Pr[Y = i] = \frac{p_i}{2} \cdot \prod_{k \neq i}(1 - \frac{p_k}{2}) \cdot (1 - \frac{p_i}{2}) = \frac{p_i}{2} \cdot \prod_{k \in L_0}(1 - \frac{p_k}{2}).$$

This implies that for every $i \in L_0$, $\Pr[Y = i \mid Y \neq \perp] = p_i$. Also

$$\Pr[Y \neq \perp] = \sum_{i \in L_0} \frac{p_i}{2} \cdot \prod_{i \in L_0}(1 - \frac{p_i}{2}) \geq \frac{1}{2} \prod_{k \in L_0} e^{-p_i} = e^{-1}/2,$$

where we used that for $a \in [0, 1/2]$, $(1 - a) \leq e^{-2a}$. □

Given this lemma we can simulate 1-MSTAT(L) by sampling Y until $Y \neq \perp$. It is easy to see that simulating m samples from 1-MSTAT(L) will require at most $4e \cdot m(L + 1)$ with probability at least $1 - \delta$ for δ exponentially small in m.

We now combine Theorems 7.1 and 7.4 to obtain the claimed lower bound for statistical algorithms using MVSTAT.

Proof of Theorem 7.3. Assuming the existence of a statistical algorithm using less than m samples we apply Theorem 7.4 for $\delta = \gamma/2 - 1/4$ to simulate the algorithm using VSTAT. The bound on m ensures that the resulting algorithm uses less than $\Omega(d(2\gamma - 1))$ queries to VSTAT($\frac{1}{3\kappa^2}$) and has success probability of at least $\gamma/2 + 1/4$. By substituting these parameters into Theorem 7.1 we obtain a contradiction. □

Finally we state an immediate corollary of Theorems 7.1, 7.2 and 7.3 that applies to general search problems and generalizes Theorem 3.4.

Theorem 7.6. *Let $\mathcal{B}(\mathcal{D}, D)$ be a decision problem. For $\kappa > 0$, let $d = \mathrm{SDN}(\mathcal{B}(\mathcal{D}, D), \kappa)$ and let $L \geq 2$ be an integer. Any randomized statistical algorithm that solves $\mathcal{B}(\mathcal{D}, D)$ with probability $\geq 2/3$ requires either*

- *$\Omega(d/L)$ calls to MVSTAT($L, 1/(12 \cdot \kappa^2 \cdot L)$);*
- *at least m calls to 1-MSTAT(L) for $m = \Omega\left(\min\left\{d, 1/\kappa^2\right\}/L\right)$.*

8 Algorithmic Bounds

In this section we prove Theorem 3.2. The algorithm is a variant of the subsampled power iteration from [FPV14] that can be implemented statistically. We describe the algorithm for the planted satisfiability model, but it can be adapted to solve Goldreich's planted k-CSP by considering only the k-tuples of variables that the predicate P evaluates to 1 on the planted assignment σ.

8.1 Set-up

Lemma 1 from [FPV14] states that subsampling r literals from a distribution Q_σ on k-clauses with distribution complexity r and planted assignment σ induces a parity distribution over clauses of length r, that is a distribution over r-clauses with planting function $Q^\delta : \{\pm1\}^r \to \mathbb{R}^+$ of the form $Q^\delta(x) = \delta/2^r$ for $|x|$ even, $Q^\delta(x) = (2-\delta)/2^r$ for $|x|$ odd, for some $\delta \in [0,2]$, $\delta \neq 1$, where $|x|$ is the number of $+1$'s in the vector x. The set of r literals to subsample from each clause is given by the set $S \subset \{1,\dots,k\}$ with $\hat{Q}(S) \neq 0$.

From here on the distribution on clauses will be given by Q_σ^δ, for $\delta \neq 1$ and planted assignment σ. For ease of analysis, we define $Q_{\sigma,p}$ as the distribution over k-clause formulas in which each possible k-clause with an even number of true literals under σ appears independently in $Q_{\sigma,p}$ with probability δp, and each clause with and odd number of true literals appears independently with probability $(2-\delta)p$, for an overall clause density p. We will be concerned with $p = \tilde{\Theta}(n^{-k/2})$. Note that it suffices to solve the algorithmic problem for this distribution instead of that of selecting exactly $m = \tilde{\Theta}(n^{k/2})$ clauses independently at random. In particular, with probability $1 - \exp(-\Theta(n))$, a sample from $Q_{\sigma,p}$ will contain at most $2p \cdot \frac{2^k n!}{(n-k)!} = O(n^k p)$ clauses.

We present statistical algorithms to recover the partition of the n variables into positive and negative literals. We will recover the partition, which gives σ up to a sign change.

The algorithm proceeds by constructing a biadjacency matrix M of size $N_1 \times N_2$ with $N_1 = 2^{\lceil k/2 \rceil} \frac{n!}{(n-\lceil k/2 \rceil)!}$, $N_2 = 2^{\lfloor k/2 \rfloor} \frac{n!}{(n-\lfloor k/2 \rfloor)!}$. We set $N = \sqrt{N_1 N_2}$. For even k, we have $N_1 = N_2 = N$ and thus M is a square matrix. The rows of the matrix are indexed by ordered subsets $S_1, \dots, S_{N_1}$ of $\lceil k/2 \rceil$ literals and columns by subsets $T_1, \dots, T_{N_2}$ of $\lfloor k/2 \rfloor$ literals. For a formula $\mathcal{F}$, we construct a matrix $\hat{M}(\mathcal{F})$ as follows. For each k-clause $(l_1, l_2, \dots, l_k)$ in $\mathcal{F}$, we put a 1 in the entry of $\hat{M}$ whose row is indexed by the set $(l_1, \dots, l_{\lceil k/2 \rceil})$ and column by the set $(l_{\lceil k/2 \rceil + 1}, \dots, l_k)$.

Let $M_{\sigma,p}$ denote the distribution on random $N_1 \times N_2$ matrices induced by drawing a random formula according to $Q_{\sigma,p}$ and forming the associated matrix $M(Q_{\sigma,p})$ as above.

For k even, let $u \in \{\pm1\}^N$ be the vector with a $+1$ entry in every coordinate indexed by subsets containing an even number of true literals under σ, and a -1 entry for every odd subset. For k odd, define the analogous vectors $u_y \in \{\pm1\}^{N_1}$ and $u_x \in \{\pm1\}^{N_2}$, again with $+1$'s for even subsets and -1 for odd subsets.

The algorithm will apply a modified power iteration procedure with rounding to find u or u_x (up to a change of sign). From these vectors the partition σ into true and false literals can be determined by solving a system of linear equations.

For even k, the discrete power iteration begins by sampling a random vector $x^0 \in \{\pm1\}^N$ and multiplying by a sample of $M_{\sigma,p}$. We then randomly round each coordinate of $M_{\sigma,p}x^0$ to ±1 to get x^1, and then repeat, drawing a fresh sample from $M_{\sigma,p}$ at each step. The rounding at each is probabilistic and depends on the value of each coordinate and the maximum value of all the coordinates. The number of clauses used by the algorithm is the sum of the number of clauses used in each sampled matrix, or in other words, if we use T samples from $M_{\sigma,p}$, our original formula needs density $T \cdot p$. [4]

For odd k, we begin with a random $x^0 \in \{\pm1\}^{N_2}$ and a random sample $M_{\sigma,p}$, then form y^0 by deterministically rounding $M_{\sigma,p}x^0$ to a vector with entries $-1, 0,$ or $+1$. Then we form x^1 by taking a fresh sample of $M_{\sigma,p}$ and perform a randomized ±1 rounding of $M_{\sigma,p}^T y^0$, and repeat. There is a

[4] Obtaining T (nearly) independent samples of $M_{\sigma,p}$ from one sample of $M_{\sigma,Tp}$ is a subtle issue and is addressed in [FPV14]; for the purposes of the implementation by statistical oracles this is irrelevant and so we avoid the discussion here.

final rounding step to find a ±1 vector that matches u or u_x.

In Section 8.4 we will prove that this algorithm can be implemented *statistically* in any of the following ways:

1. Using $O(n^{r/2}\log^2 n)$ calls to 1-MSTAT$(n^{\lceil r/2\rceil})$;

2. For even r: using $O(\log n)$ calls to MVSTAT$(n^{r/2}, n^{r/2}\log\log n)$;

3. For odd r: using $O(\log n)$ calls to MVSTAT$(O(n^{\lceil r/2\rceil}), O(n^{r/2}\log n))$.

8.2 Algorithm Discrete-Power-Iterate (even k).

1. Pick $x^0 \in \{\pm 1\}^N$ uniformly at random. For $i = 1, \dots \log N$, repeat the following:
   (a) Draw a sample matrix $M \sim M_{\sigma,p}$.
   (b) Let $x = Mx^{i-1}$.
   (c) Randomly round each coordinate of x to ±1 to get x^i as follows: let

$$x^i_j = \begin{cases} \mathsf{sign}(x_j) \text{ with probability } \frac{1}{2} + \frac{|x_j|}{2\max_j |x_j|} \\ -\mathsf{sign}(x_j) \text{ otherwise.} \end{cases}$$

2. Let $x = Mx^{\log N}$ and set $u^* = \mathsf{sign}(x)$ by rounding each coordinate to its sign.

3. Output the solution by solving the system of parity equations defined by u^*.

Lemma 8.1. *If $p = \frac{K \log N}{(\delta-1)^2 N}$ for a sufficiently large constant K, then with probability $1 - o(1)$ the above algorithm returns the planted assignment.*

The main idea of the analysis is to keep track of the random process $(u \cdot x^i)$. It starts at $\Theta(\sqrt{N})$ with the initial randomly chosen vector x^0, and then after an initial phase, doubles on every successive step whp until it reaches $N/9$.

We will use the following Chernoff bound several times (see eg. Corollary A.1.14 in [AS11]).

Proposition 8.2. *Let $X = \sum_{i=1}^m \xi_i Y_i$ and $Y = \sum_{i=1}^m Y_i$, where the Y_i's are independent Bernoulli random variables and the ξ_i's are fixed ±1 constants. Then*

$$\Pr[|X - \mathbb{E}[X]| \geq \alpha\, \mathbb{E}[Y]] \leq e^{-\alpha^2\, \mathbb{E}[Y]/3}.$$

Proposition 8.3. *If $|x^i \cdot u| = \beta N \geq \sqrt{N}\log\log N$, then with probability $1 - O(1/N\beta^2)$,*

$$|x^{i+1} \cdot u| \geq \min\left\{\frac{N}{9}, 2|x^i \cdot u|\right\}.$$

Proof. We assume WLOG that $\delta > 1$ and $x^0 \cdot u > 0$ in what follows. Let $U^+ = \{i : u_i = +1\}$, $U^- = \{i : u_i = -1\}$, $X^+ = \{i : x_i = +1\}$, and $X^- = \{i : x_i = +1\}$. For a given $j \in [N]$, let $A_j = \{i : \text{sets } i \text{ and } j \text{ share no variables}\}$. We have $|A_j| = N^*$ for all j.

Let $z = Mx^i$. Note that the coordinates $z_1, \dots z_N$ are independent and if $j \in U^+$,

$$z_j \sim Z_{++} + Z_{-+} - Z_{+-} - Z_{--} - p(|X^+| - |X^-|)$$

where

$$\begin{aligned}
Z_{++} &\sim Bin(|U^+ \cap X^+ \cap A_j|, \delta p), \\
Z_{-+} &\sim Bin(|U^- \cap X^+ \cap A_j|, (2-\delta)p), \\
Z_{+-} &\sim Bin(|U^+ \cap X^- \cap A_j|, \delta p), \\
Z_{--} &\sim Bin(|U^- \cap X^- \cap A_j|, (2-\delta)p).
\end{aligned}$$

We can write a similar expression if $j \in U^-$, with the probabilities swapped. For $j \in U^+$ we calculate,

$$\begin{aligned}
\mathbb{E}[z_j] &= \delta p|U^+ \cap X^+ \cap A_j| + (2-\delta)p|U^- \cap X^+ \cap A_j| - \delta p|U^+ \cap X^- \cap A_j| - (2-\delta)p|U^- \cap X^- \cap A_j| \\
&\quad - p\left(|X^+| - |X^-|\right) \\
&= \delta p|U^+ \cap X^+| + (2-\delta)p|U^- \cap X^+| - \delta p|U^+ \cap X^-| \\
&\quad - (2-\delta)p|U^- \cap X^-| - p\left(|X^+| - |X^-|\right) + O((N - N^*)p) \\
&= (\delta - 1)p(u \cdot x) + O(n^{k/2-1}p).
\end{aligned}$$

For $j \in U^-$ we get $\mathbb{E}[z_j] = (1-\delta)p(u \cdot x) + O(n^{k/2-1}p)$.

To apply Proposition 8.2, note that there are N entries in each row of M, half of which have probability δp of being 1, and the other half with probability $(2-\delta)p$ of being a 1, so $\mathbb{E}[Y] = Np$. Using the proposition with $\alpha = (\delta-1)/26$ and union bound, we have that with probability $1 - o(N^{-1})$,

$$\begin{aligned}
\max_j |z_j| &\leq (\delta - 1)p \cdot (u \cdot x) + \frac{(\delta-1)Np}{26} + O(n^{k/2-1}p) \\
&\leq (\delta - 1)p \cdot (u \cdot x) + \frac{(\delta-1)Np}{25}.
\end{aligned} \tag{10}$$

For each ordered set of $k/2$ literals indexed by $j \in U^+$, there is a set indexed by $j' \in U^-$ that is identical except the first literal in j' is the negation of the first literal in j. Note that $A_j = A_{j'}$, and so we can calculate:

$$\mathbb{E}[z_j] - \mathbb{E}[z_{j'}] = 2(\delta-1)p\left[|U^+ \cap X^+ \cap A_j| + |U^- \cap X^- \cap A_j| - |U^- \cap X^+ \cap A_j| - |U^+ \cap X^- \cap A_j|\right]$$

which is simply $2(\delta-1)p$ times the dot product of u and x restricted to the coordinates A_j. Summing over all $j \in [N]$ we get

$$\begin{aligned}
\mathbb{E}[u \cdot z] &= |A_j|(\delta-1)p(u \cdot x) \\
&= N^*(\delta-1)p(u \cdot x) \\
&= N(\delta-1)p(u \cdot x)(1 + o(1)).
\end{aligned}$$

Applying Proposition 8.2 to $(u \cdot z)$ (with $\mathbb{E}[Y] = N^2 p$, and $\alpha = \frac{(\delta-1)(u \cdot x)}{2N}$), we get

$$\begin{aligned}
\Pr[(u \cdot z) < N(\delta-1)p(u \cdot x)/2] &\leq \exp\left[-\frac{N^2 p(\delta-1)^2(u \cdot x)^2}{12N^2}\right] \\
&= \exp\left[-\frac{K \log N (u \cdot x)^2}{12N}\right] = o\left(\frac{1}{N}\right).
\end{aligned} \tag{11}$$

Now we round z to a ± 1 vector x' as above. Let Z be the number of j's so that $x'_j = u_j$. Then, conditioning on $u \cdot z$ and $\max |z_j|$ as above,

$$\begin{aligned}\mathbb{E}[Z] &= \sum_{j=1}^{N} \left(\frac{1}{2} + \frac{u_j z_j}{2 \max |z_j|} \right) \\ &= \frac{N}{2} + \frac{u \cdot z}{2 \max |z_j|} \\ &\geq \frac{N}{2} + \frac{N(\delta - 1)p(u \cdot x)}{4((\delta - 1)p(u \cdot x) + \frac{(\delta-1)Np}{25})}.\end{aligned}$$

If $(\delta - 1)p(u \cdot x) \leq \frac{(\delta-1)Np}{25}$, we have

$$\mathbb{E}[Z] \geq \frac{N}{2} + \frac{N(\delta - 1)p(u \cdot x)}{8(\delta - 1)Np/25} \geq \frac{N}{2} + 3(u \cdot x).$$

If $(\delta - 1)p(u \cdot x) \geq \frac{(\delta-1)Np}{25}$, we have

$$\mathbb{E}[Z] \geq \frac{N}{2} + \frac{N(\delta - 1)p(u \cdot x)}{8(\delta - 1)p(u \cdot x)} = \frac{5N}{8}.$$

Note that the variance of Z is at most $N/4$. From Chebyshev's inequality, with probability $1 - O(N/(u \cdot x)^2)$, $Z \geq \min \left\{ \frac{N}{2} + (u \cdot x), \frac{5N}{9} \right\}$, which completes the proof of Proposition 8.3. □

Finishing: We consider two phases. When $|u \cdot x| < \sqrt{N} \log \log N$, with probability at least 1/2, $|u \cdot x^{i+1}| \geq \max\{\sqrt{N}/10, 2|u \cdot x^i|\}$. This follows from Berry-Esseen bounds in the Central Limit Theorem: Z is the sum of N independent $0, 1$ random variables with different probabilities, and we know at least $9N/10$ have a probability between $2/5$ and $3/5$ (comparing a typical $|z_i|$ with $\max |z_i|$). This shows the variance of Z is at least $N/5$ when $u \cdot x$ is this small.

Now call a step 'good' if $|u \cdot x^{i+1}| \geq \max\{\sqrt{N}/10, 2|u \cdot x^i|\}$. Then in $\log N$ steps whp there is at least one run of at least $\log \log N$ good steps, and after any such run we have $|u \cdot x| \geq \sqrt{N} \log \log N$ with certainty, completing the first phase.

Once we have $|u \cdot x| \geq \sqrt{N} \log \log N$, then according to Proposition 8.3, after $O(\log N)$ steps the value of $|x^u \cdot u|$ successively doubles with error probabilities that are geometrically decreasing, and so whp at the end we have a vector $x \in \{\pm 1\}^N$ so that $|u \cdot x| \geq \frac{N}{9}$. In the positive case, when we multiply x once more by $M \sim M_{\sigma,p}$, we have for $i : u_i = 1$, $\mathbb{E}[(Mx)_i] \geq (\delta - 1)pN/9$. Using Proposition 8.2 (with $\mathbb{E}[Y] = Np$ and $\alpha = (\delta - 1)/10$),

$$\Pr[(Mx)_i \leq 0] \leq e^{-cNp} = o(N^{-2})$$

Similarly, if $u_i = -1$, $\Pr[(Mx)_i \geq 0] = o(N^{-2})$, and thus whp rounding to the sign of x will give us u exactly. The same holds in the negative case where we will get $-u$ exactly.

8.3 Algorithm Discrete-Power-Iterate (odd k)

1. Pick $x^0 \in \{\pm 1\}^{N_2}$ uniformly at random. For $i = 1, \ldots \log N$, repeat the following:

   (a) Draw a sample matrix $M \sim M_{\sigma,p}$.

(b) Let $\overline{y}^i = Mx^{i-1}$; round $\overline{y}^i$ to a vector y^i with entries $0, +1$, or -1, according to the sign of the coordinates.

(c) Draw another sample $M \sim M_{\sigma,p}$.

(d) Let $\overline{x}^i = M^T y^i$. Randomly round each coordinate of $\overline{x}^i$ to ± 1 as follows to get x^i:

$$x^i_j = \begin{cases} \mathsf{sign}(\overline{x}_j) \text{ with probability } \frac{1}{2} + \frac{|\overline{x}_j|}{2\max_j |\overline{x}_j|} \\ -\mathsf{sign}(\overline{x}_j) \text{ otherwise.} \end{cases}$$

2. Set $u^* = \mathsf{sign}(x^{\log N})$ by rounding each coordinate to its sign.

3. Output the solution by solving the system of parity equations defined by u^*.

Lemma 8.4. *Set $p = \frac{K \log N}{(\delta-1)^2 N}$. Then whp, the algorithm returns the planted assignment.*

We will keep track of the inner products $x^i \cdot u_x$ and $y^i \cdot u_y$ as the algorithm iterates.

Proposition 8.5. *If $|x^i \cdot u_x| = \beta N_2 \geq \sqrt{N_2}/\log\log N$, then with probability $1 - o(1/\log N)$,*

1. $|y^{i+1} \cdot u_y| \geq N\beta \log N$

2. $\|y^{i+1}\|_1 = N^2 p(1 + o(1))$.

Proof. Let $x \in \{\pm 1\}^{N_2}$ and $M \sim M_{\sigma,p}$. Let $y = Mx$. We will assume $\delta > 1$ and $x \cdot u_x > 0$ for simplicity.

If $j \in U_y^+$, then

$$\Pr[y_j \geq 1] = \delta p |X^+ \cap U_x^+| + (2-\delta)p|X^+ \cap U_x^-| + O((N_2 - N_2^*)p) + O(p^2 N_2),$$
and
$$\Pr[y_j \leq -1] = \delta p |X^- \cap U_x^+| + (2-\delta)p|X^- \cap U_x^-| + O((N_2 - N_2^*)p) + O(p^2 N_2),$$

and similarly for $j \in U_y^-$:

$$\Pr[y_j \geq 1] = \delta p |X^+ \cap U_x^-| + (2-\delta)p|X^+ \cap U_x^+| + O((N_2 - N_2^*)p) + O(p^2 N_2),$$
and
$$\Pr[y_j \leq -1] = \delta p |X^- \cap U_x^-| + (2-\delta)p|X^- \cap U_x^+| + O((N_2 - N_2^*)p) + O(p^2 N_2).$$

Rounding y by the sign of each coordinate gives a $0, +1, -1$ vector y'. Let Y^+ be the set of $+1$ coordinates of y', and Y^- the set of -1 coordinates. An application of Proposition 8.2 with $\mathbb{E}[Y] = N^2 p$ and $\alpha = 1/\log N$ immediately gives $\|y'\|_1 = N^2 p(1+o(1))$ with probability $1 - o(N^{-1})$.

We can write

$$y' \cdot u_y = |Y^+ \cap U_y^+| + |Y^- \cap U_y^-| - |Y^+ \cap U_y^-| - |Y^- \cap U_y^+|,$$

and compute

$$\begin{aligned}\mathbb{E}[y' \cdot u_y] &= \frac{N_1^*}{2}\left[(2\delta - 2)(|X^+ \cap U_x^+| + |X^- \cap U_x^-| - |X^+ \cap U_x^-| - |X^- \cap U_x^+|)\right] + O(N_1 N_2 p^2) \\ &= N_1 p(\delta - 1)(x \cdot u_x)(1 + o(1)).\end{aligned}$$

Another application of Proposition 8.2 with $\mathbb{E}[Y] = N_1 N_2 p$ and $\alpha = \frac{(\delta-1)(x\cdot u_x)}{2N_2}$ shows that with probability $1 - o(N^{-2})$,

$$y' \cdot u_y \geq N_1 p(\delta - 1)(x \cdot u_x)/2 = \frac{N\beta C \log N}{2(\delta-1)} \geq N\beta \log N. \tag{12}$$

□

Proposition 8.6. *If $|y^i \cdot u_y| = \gamma N \geq \sqrt{N_1} \log N / \log\log N$ with $\|y\|_1 = N^2 p(1 + o(1))$, then with probability $1 - o(1/\log N)$,*

$$|x^i \cdot u_x| \geq \min\left\{\frac{N_2}{9}, \frac{N_2 c\gamma}{\sqrt{\log N}}\right\}.$$

for some constant $c = c(\delta, K)$.

Proof. For $j \in U_x^+$ as above we calculate,

$$\begin{aligned}\mathbb{E}[x_j] &= \delta p|U_y^+ \cap Y^+| + (2-\delta)p|U_y^- \cap Y^+| - \delta p|U_y^+ \cap Y^-| \\ &\quad - (2-\delta)p|U_y^- \cap Y^-| - p\left(|Y^+| - |Y^-|\right) + O((N_1 - N_1^*)p) \\ &= (\delta - 1)p(u_y \cdot y) + O((N_1 - N_1^*)p) \\ &= (\delta - 1)p(u_y \cdot y) + O(N_2 p)\end{aligned}$$

And for $j \in U_x^-$, $\mathbb{E}[x_j] = -(\delta-1)p(u_y \cdot y) + O(N_2 p)$. We also have $\mathbb{E}[u_x \cdot x] = (\delta - 1)N_2^* p(u_y \cdot y)$.

Proposition 8.2 with $\mathbb{E}[Y] = N_1 p$ and $\alpha = \frac{N_2 p}{\sqrt{\log N}}$ shows that with probability $1 - o(N^{-1})$,

$$\max_j |x_j| \leq |\delta - 1|p(u_y \cdot y) + \frac{N^2 p^2}{\sqrt{\log N}},$$

and applied with $\mathbb{E}[Y] = N_1 N_2^2 p^2$ and $\alpha = \frac{1}{\sqrt{N_2 \log N}}$ shows that with probability $1 - o(N^{-1})$,

$$\begin{aligned}(u_x \cdot x) &\geq (\delta-1)N_2 p(u_y \cdot y) - \frac{N^2 p^2 \sqrt{N_2}}{\sqrt{\log N}} \\ &= (\delta - 1)N_2 p(u_y \cdot y)(1 + o(1))\end{aligned}$$

for $(u_y \cdot y) \geq \sqrt{N_1} \log N / \log\log N$. Again we randomly round to a vector x^*, and if Z is the number of of coordinates on which x^* and u_x agree,

$$\mathbb{E}[Z] = \frac{N_2}{2} + \frac{u_x \cdot x}{2\max|x_j|} \geq \frac{N_2}{2} + \frac{N_2(\delta-1)p(u_y \cdot y)}{4((\delta-1)p(u_y \cdot y) + \frac{N^2p^2}{\sqrt{\log N}})}.$$

If $(\delta - 1)p(u_y \cdot y) \leq \frac{N^2p^2}{\sqrt{\log N}}$, we have

$$\mathbb{E}[Z] \geq \frac{N_2}{2} + \frac{N_2(\delta-1)p(u_y \cdot y)}{8N^2p^2/\sqrt{\log N}} = \frac{N_2}{2} + \frac{N_2\gamma(\delta-1)^3}{8K\sqrt{\log N}}.$$

If $(\delta - 1)p(u \cdot x) \geq \frac{N^2p^2}{\sqrt{\log N}}$, we have

$$\mathbb{E}[Z] \geq \frac{N_2}{2} + \frac{N_2(\delta-1)p(u \cdot x)}{8(\delta-1)p(u \cdot x)} = \frac{5N_2}{8}.$$

Another application of Proposition 8.2 with $\mathbb{E}[Y] = \mathbb{E}[Z]$ and $\alpha = \frac{(\delta-1)^3\gamma}{100K\sqrt{\log N}}$ shows that with probability $1 - o(1)$,

$$Z \geq \min\left\{\frac{N_2}{2} + \frac{N_2\gamma(\delta-1)^3}{9K\sqrt{\log N}}, \frac{5N_2}{9}\right\},$$

which shows that $x^* \cdot u_x \geq \min\left\{\frac{N_2 c\gamma}{\sqrt{\log N}}, \frac{N_2}{9}\right\}$ for some constant $c = c(\delta, K)$. □

Finishing: Choosing x^0 at random gives $|x^0 \cdot u_x| \geq \frac{\sqrt{N_2}}{\log\log N}$ whp. After a pair of iterations, Propositions 8.5 and 8.6 guarantee that whp the value of $x^i \cdot u_x$ rises by a factor of $\sqrt{\log N}/K$, so after at most $\log N$ steps we have a vector $x \in \{\pm 1\}^{N_2}$ with $|x \cdot u_x| \geq \frac{N_2}{9}$. One more iteration gives a vector y with $|u_y \cdot y| \geq \frac{N \log N}{9}$. Now consider $(M^T y)$. In the positive case (when $u_y \cdot y \geq \frac{N\log N}{9}$), we have for $i \in U_x^+$, $\mathbb{E}[(M^T y)_i] \geq (\delta-1)Np\log N/9$. Using Proposition 8.2, $\Pr[(M^T y)_i \leq 0] = o(N^{-2})$. Similarly, if $i \in U_x^-$, $\Pr[(M^T y)_i \geq 0] = o(N^{-2})$, and thus whp rounding to the sign of the vector will give us u_x exactly. The same holds in the negative case where we will get $-u_x$ exactly.

8.4 Implementing the algorithms with the statistical oracle

We complete the proof of Theorem 3.2 by showing how to implement the above algorithms with the statistical oracles 1-MSTAT and MVSTAT.

Lemma 8.7 (Even k)**.** *There is a randomized algorithm that makes $O(N\log^2 N)$ calls to the 1-MSTAT(N) oracle and returns the planted assignment with probability $1 - o(1)$. There is a randomized algorithm that makes $O(\log N)$ calls to the MVSTAT(t, L) oracle with $L = N$ and $t = N\log\log N$, and returns the planted assignment with probability $1 - o(1)$.*

Proof. We can run the above algorithm using the 1-MSTAT(N) oracle. Given a vector $x \in \{\pm 1\}^N$, we compute x', the next iteration, as follows: each $j \in [N]$ corresponds to a different value of the query functions h^+ and h^- defined as $h^+(X) = i$ if the clause $X = (i, j)$ for $j : x_j = +1$ and zero otherwise, and similarly $h^-(X) = i$ if $X = (i, j)$ for $j : x_j = -1$ and zero otherwise. For use in the implementation, we define the Boolean functions h_i^+ as $h_i^+(X) = 1$ iff $h^+(X) = i$. Let v_i^+, v_i^- denote the corresponding oracle's responses to the two queries, and $v_i = v_i^+ - v_i^-$. Now to compute x', for each coordinate we sum v_i over all samples and subtract $p\sum x_i$. We use $O(\log N)$ such iterations, and we use $O(N\log N)$ clauses per iteration (corresponding to $p = \frac{K\log N}{(\delta-1)^2 N}$).

To use the MVSTAT oracle, we note that for each query function v, we make $t = O(N\log N)$ calls to 1-MSTAT(N). We can replace each group of t calls with a one call to MVSTAT(N, t). Let the response be a vector p in $[0,1]^L$, with $L+2$ subsets, namely singleton subsets for each coordinate as well as for the subsets with positive parity and with negative parity on the unknown assignment σ. For each coordinate l, we set $v_l = \text{Binom}(1, p_l)$, the output of an independent random coin toss with bias p_l. The guarantees on MVSTAT imply that the result of this simulation are equivalent for our purposes to directly querying 1-MSTAT. Here we give a direct simulation with smaller t.

For $t = N\log\log N$, versions of equations (10) and (11) (properly scaled) hold due to the oracle's bound on $|v_i|$ and the bound on $\sum_V v_i$.

In particular, we can calculate that $\mathbb{E}[h_i^+ - h_i^- - \frac{1}{N}\sum_j x_j] = u_i \cdot \frac{(\delta-1)\beta}{N} + O\left(\frac{1}{N(N-N^*)}\right)$ where $u \cdot x = \beta N$. The oracle bounds then give $\max_i |v_i| \leq \frac{(\delta-1)\beta}{N} + \frac{2}{\sqrt{tN}} = \frac{(\delta-1)\beta}{N} + \frac{2}{N\sqrt{\log\log N}}$ since $t \gg (N - N^*)$. The oracle also guarantees that $|u \cdot v - (\delta-1)\beta| \leq \frac{1}{\sqrt{t}}$, and so for $\beta \geq \frac{2}{\sqrt{N\log\log N}}$, $u \cdot v \geq (\delta-1)\beta/2$.

Now we do the same randomized rounding as above, and we see that

$$\begin{aligned}\mathbb{E}[Z] &= \sum_{j=1}^{N}\left(\frac{1}{2}+\frac{u_j v_j}{2\max|v_j|}\right)\\ &= \frac{N}{2}+\frac{u\cdot v}{2\max|v_j|}\\ &\geq \frac{N}{2}+\frac{(\delta-1)\beta}{4(\frac{(\delta-1)\beta}{N}+\frac{2}{N\sqrt{\log\log N}})}.\end{aligned}$$

If $\frac{(\delta-1)\beta}{N} \leq \frac{2}{N\sqrt{\log\log N}}$, we have

$$\mathbb{E}[Z] \geq \frac{N}{2}+\frac{(\delta-1)\beta}{\frac{16}{N\sqrt{\log\log N}}} = \frac{N}{2}+\frac{\sqrt{\log\log N}(\delta-1)}{16}\beta N.$$

If $\frac{(\delta-1)\beta}{N} \geq \frac{2}{N\sqrt{\log\log N}}$, we have

$$\mathbb{E}[Z] \geq \frac{N}{2}+\frac{(\delta-1)\beta}{8(\delta-1)\beta/N} = \frac{5N}{8}.$$

The variance of Z is at most $N/4$, and with probability $1-o(1)$ we start with $|x^0\cdot u| \geq \sqrt{N}/\log\log\log N$. Then successive applications of Chebyshev's inequality as above show that whp after at most $\log N$ steps, we have $|x^i \cdot u| \geq \frac{5N}{8}$.

□

Lemma 8.8 (Odd k)**.** *There is a randomized algorithm that makes $O(n^{k/2}\log^2 n)$ calls to the 1-MSTAT(L) oracle for $L = N_1$, and returns the planted assignment with probability $1-o(1)$.*

Proof. We run the algorithm using 1-MSTAT, alternately querying N_1-valued functions and N_2-valued functions, each with $t = O(N\log N)$ samples per iteration. Since there are $O(\log N)$ iterations in all, this gives the claimed bound of $O(N\log^2 N)$ calls to 1-MSTAT(N_1).

To implement using MVSTAT, we do as described in proof for the even case. Evaluation of an L-valued query h with t samples via t calls to 1-MSTAT(L) is replaced by one call to MVSTAT(L, t) and this response is used to generate a 0/1 vector, each time with subsets corresponding to all singletons and the two subsets with different parities according to the planted assignment σ. This gives the bounds claimed in Theorem 3.2. To see that the algorithm converges as claimed, we note that Prop. 8.5 continues to hold, with a lower order correction term in Equation (12) for the difference when $y'\cdot u_y$ when y' is obtained by the above simulation. This difference is small as guaranteed by the MVSTAT oracle on the two subsets corresponding to the positive support and negative support of u_y. □

9 Discussion and open problems

By querying well-chosen sequences of functions, statistical query algorithms can be efficient and just as powerful as unconstrained algorithmic approaches, in spite of not being able to directly examine samples from an input distribution. As far as we know, there is only one counterexample, namely solving equations over finite fields, which can be done easily by Gaussian elimination but not with any efficient statistical query algorithm. Here we have given a unifying model of planted constraint satisfaction problems and characterized their SQ complexity. Our bounds correspond closely to known upper bounds for unconstrained algorithms.

Our work also gives a new technique for proving lower bounds on SQ algorithm that strengthens and generalizes previous techniques. It has already been crucial in getting tight lower bounds on SQ complexity of stochastic linear optimization and high-dimensional mean estimation [FGV15]. It also served as a step toward a characterization of the SQ complexity of solving general problems over distributions given in [Fel16].

We conclude with some candidate directions for future research.

1. A long-standing and intriguing question is to find an additional example (besides solving equations over finite fields) of a natural problem over distributions for which there exists an efficient algorithm that beats the lower bound for statistical algorithms, and disproves our conjecture.

2. Which additional problems can be addressed using the methods of this paper? One interesting candidate is the problem of detection in a stochastic block model with $k > 2$ blocks? There is currently a gap between the information-theoretic and algorithmic thresholds for the number of edges needed for detection, but the gap is only a factor of roughly $k/\log k$. A special case of this problem is planted k-coloring.

3. It would be interesting to better understand the relationship of our lower bounds for convex program relaxations to those known for hierarchies of LP and SDP relaxations. Does there exist a unifying approach?

Acknowledgments

We thank Amin Coja-Oghlan, Florent Krzakala, Ryan O'Donnell, Prasad Raghavendra, and Lenka Zdeborová for insightful comments and helpful discussions. We also thank Jan Vondrak for the proof idea of Lemma 7.5.

References

[ABBS14] Emmanuel Abbe, Afonso S Bandeira, Annina Bracher, and Amit Singer. Decoding binary node labels from censored edge measurements: Phase transition and efficient recovery. *IEEE Transactions on Network Science and Engineering*, 1(1):10–22, 2014.

[ABR12] Benny Applebaum, Andrej Bogdanov, and Alon Rosen. A dichotomy for local small-bias generators. In *Theory of Cryptography*, pages 600–617. Springer, 2012.

[ABW10] Benny Applebaum, Boaz Barak, and Avi Wigderson. Public-key cryptography from different assumptions. In *STOC*, pages 171–180. ACM, 2010.

[ACO08] Dimitris Achlioptas and Amin Coja-Oghlan. Algorithmic barriers from phase transitions. In *FOCS*, pages 793–802. IEEE, 2008.

[AJM05] Dimitris Achlioptas, Haixia Jia, and Cristopher Moore. Hiding satisfying assignments: Two are better than one. *J. Artif. Intell. Res.(JAIR)*, 24:623–639, 2005.

[AL16] Benny Applebaum and Shachar Lovett. Algebraic attacks against random local functions and their countermeasures. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 1087–1100. ACM, 2016.

[Ale11] Michael Alekhnovich. More on average case vs approximation complexity. *Computational Complexity*, 20(4):755–786, 2011.

[AM09] Per Austrin and Elchanan Mossel. Approximation resistant predicates from pairwise independence. *Computational Complexity*, 18(2):249–271, 2009.

[AM15] Emmanuel Abbe and Andrea Montanari. Conditional random fields, planted constraint satisfaction, and entropy concentration. *Theory of Computing*, 11:413–443, 2015.

[AOW15] Sarah R. Allen, Ryan O'Donnell, and David Witmer. How to refute a random CSP. In *FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 689–708, 2015.

[App13] Benny Applebaum. Pseudorandom generators with long stretch and low locality from random local one-way functions. *SIAM Journal on Computing*, 42(5):2008–2037, 2013.

[App16] Benny Applebaum. Cryptographic hardness of random local functions. *Computational complexity*, 25(3):667–722, 2016.

[AS11] Noga Alon and Joel H Spencer. *The Probabilistic Method*, volume 73. John Wiley & Sons, 2011.

[BBDV14] Jeremiah Blocki, Manuel Blum, Anupam Datta, and Santosh Vempala. Human computable passwords. *CoRR*, abs/1404.0024, 2014.

[BD98] Shai Ben-David and Eli Dichterman. Learning with restricted focus of attention. *J. Comput. Syst. Sci.*, 56(3):277–298, 1998.

[BDMN05] Avrim Blum, Cynthia Dwork, Frank McSherry, and Kobbi Nissim. Practical privacy: the SuLQ framework. In *PODS*, pages 128–138, 2005.

[Bec75] William Beckner. Inequalities in fourier analysis. *The Annals of Mathematics*, 102(1):159–182, 1975.

[BF15] Maria-Florina Balcan and Vitaly Feldman. Statistical active learning algorithms for noise tolerance and differential privacy. *Algorithmica*, 72(1):282–315, 2015.

[BFJ$^+$94] Avrim Blum, Merrick Furst, Jeffrey Jackson, Michael Kearns, Yishay Mansour, and Steven Rudich. Weakly learning DNF and characterizing statistical query learning using Fourier analysis. In *STOC*, pages 253–262, 1994.

[BFKV98] Avrim Blum, Alan Frieze, Ravi Kannan, and Santosh Vempala. A polynomial-time algorithm for learning noisy linear threshold functions. *Algorithmica*, 22(1-2):35–52, 1998.

[BHL$^+$02] 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.

[BKS13] Boaz Barak, Guy Kindler, and David Steurer. On the optimality of semidefinite relaxations for average-case and generalized constraint satisfaction. In *ITCS*, pages 197–214. ACM, 2013.

[Bon70] Aline Bonami. Étude des coefficients de fourier des fonctions de $l_p(g)$. In *Annales de l'institut Fourier*, volume 20, pages 335–402. Institut Fourier, 1970.

[Bop87] Ravi B Boppana. Eigenvalues and graph bisection: An average-case analysis. In *FOCS*, pages 280–285. IEEE, 1987.

[BQ09] Andrej Bogdanov and Youming Qiao. On the security of Goldreich's one-way function. In *RANDOM-APPROX*, pages 392–405. Springer, 2009.

[CEMT09] James Cook, Omid Etesami, Rachel Miller, and Luca Trevisan. Goldreich's one-way function candidate and myopic backtracking algorithms. In *Theory of Cryptography*, pages 521–538. Springer, 2009.

[CKL$^+$06] Cheng-Tao Chu, Sang Kyun Kim, Yi-An Lin, YuanYuan Yu, Gary Bradski, Andrew Y. Ng, and Kunle Olukotun. Map-reduce for machine learning on multicore. In *NIPS*, pages 281–288, 2006.

[CO06] Amin Coja-Oghlan. A spectral heuristic for bisecting random graphs. *Random Structures & Algorithms*, 29:3:351–398, 2006.

[COCF10] Amin Coja-Oghlan, Colin Cooper, and Alan Frieze. An efficient sparse regularity concept. *SIAM Journal on Discrete Mathematics*, 23(4):2000–2034, 2010.

[COGL04] 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.

[COGLS04] 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.

[CW04] Moses Charikar and Anthony Wirth. Maximizing quadratic programs: Extending Grothendieck's inequality. In *FOCS*, pages 54–60, 2004.

[DFKO07] Irit Dinur, Ehud Friedgut, Guy Kindler, and Ryan O'Donnell. On the Fourier tails of bounded functions over the discrete cube. *Israel Journal of Mathematics*, 160(1):389–412, 2007.

[DKMZ11] 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.

[DLR77] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

[DLSS13] 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.

[DV08] John Dunagan and Santosh Vempala. A simple polynomial-time rescaling algorithm for solving linear programs. *Math. Program.*, 114(1):101–114, 2008.

[Fei02] Uriel Feige. Relations between average case complexity and approximation complexity. In *STOC*, pages 534–543. ACM, 2002.

[Fel12] Vitaly Feldman. A complete characterization of statistical query learning with applications to evolvability. *Journal of Computer System Sciences*, 78(5):1444–1459, 2012.

[Fel16] Vitaly Feldman. A general characterization of the statistical query complexity. *CoRR*, abs/1608.02198, 2016. Extended abstract in COLT 2017.

[Fel17] Vitaly Feldman. Statistical query learning. In *Encyclopedia of Algorithms*, pages 2090–2095. 2017. Available at `http://vtaly.net/papers/Kearns93-2017.pdf`.

[FGK05] Joel Friedman, Andreas Goerdt, and Michael Krivelevich. Recognizing more unsatisfiable random k-SAT instances efficiently. *SIAM Journal on Computing*, 35(2):408–430, 2005.

[FGR$^+$12] Vitaly Feldman, Elena Grigorescu, Lev Reyzin, Santosh Vempala, and Ying Xiao. Statistical algorithms and a lower bound for detecting planted cliques. *arXiv, CoRR*, abs/1201.1214, 2012. Extended abstract in STOC 2013.

[FGV15] Vitaly Feldman, Cristobal Guzman, and Santosh Vempala. Statistical query algorithms for mean vector estimation and stochastic convex optimization. *CoRR*, abs/1512.09170, 2015. Extended abstract in SODA 2017.

[Fla03] Abraham Flaxman. A spectral technique for random satisfiable 3cnf formulas. In *SODA*, pages 357–363, 2003.

[FO04] Uriel Feige and Eran Ofek. Easily refutable subformulas of large random 3-CNF formulas. In *Automata, languages and programming*, pages 519–530. Springer, 2004.

[FPV13] Vitaly Feldman, Will Perkins, and Santosh Vempala. On the complexity of random satisfiability problems with planted solutions. *CoRR*, abs/1311.4821, 2013. Extended abstract in STOC 2015.

[FPV14] Vitaly Feldman, Will Perkins, and Santosh Vempala. Subsampled power iteration: a new algorithm for block models and planted CSP's. *CoRR*, abs/1407.2774, 2014. Extended abstract in NIPS 2015.

[FPV15] Vitaly Feldman, Will Perkins, and Santosh Vempala. On the complexity of random satisfiability problems with planted solutions. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, pages 77–86. ACM, 2015.

[GL03] Andreas Goerdt and André Lanka. Recognizing more random unsatisfiable 3-SAT instances efficiently. *Electronic Notes in Discrete Mathematics*, 16:21–46, 2003.

[Gol00] Oded Goldreich. Candidate one-way functions based on expander graphs. *IACR Cryptology ePrint Archive*, 2000:63, 2000.

[GS90] Alan E. Gelfand and Adrian F.M. Smith. Sampling based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85:398–409, 1990.

[GS14] David Gamarnik and Madhu Sudan. Performance of the survey propagation-guided decimation algorithm for the random NAE-K-SAT problem. *arXiv preprint arXiv:1402.0052*, 2014.

[GW95] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42:1115–1145, 1995.

[Han12] Lars Peter Hansen. Large sample properties of generalized method of moments estimators. *Econometrica*, 50:1029–1054, 2012.

[HPS09] Hiêp Hàn, Yury Person, and Mathias Schacht. Note on strong refutation algorithms for random k-SAT formulas. *Electronic Notes in Discrete Mathematics*, 35:157–162, 2009.

[IKOS08] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Cryptography with constant computational overhead. In *STOC*, pages 433–442. ACM, 2008.

[Jan97] Svante Janson. *Gaussian Hilbert spaces.* Cambridge University Press, 1997.

[JMS05] Haixia Jia, Cristopher Moore, and Doug Strain. Generating hard satisfiable formulas by hiding solutions deceptively. In *AAAI*, volume 20, page 384, 2005.

[Kea98] Michael Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM*, 45(6):983–1006, 1998.

[KJV83] Scott Kirkpatrick, D. Gelatt Jr., and Mario P. Vecchi. Optimization by simmulated annealing. *Science*, 220(4598):671–680, 1983.

[KMR17] Pravesh K. Kothari, Raghu Meka, and Prasad Raghavendra. Approximating rectangles by juntas and weakly-exponential lower bounds for LP relaxations of csps. In *STOC*, pages 590–603, 2017.

[KMRT$^+$07] Florent Krzakała, Andrea Montanari, Federico Ricci-Tersenghi, Guilhem Semerjian, and Lenka Zdeborová. Gibbs states and the set of solutions of random constraint satisfaction problems. *Proceedings of the National Academy of Sciences*, 104(25):10318–10323, 2007.

[KMZ14] 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.

[KV06a] A. T. Kalai and S. Vempala. Simulated annealing for convex optimization. *Math. Oper. Res.*, 31(2):253–266, 2006.

[KV06b] Michael Krivelevich and Dan Vilenchik. Solving random satisfiable 3cnf formulas in expected polynomial time. In *SODA*, pages 454–463. ACM, 2006.

[KZ09] Florent Krzakala and Lenka Zdeborová. Hiding quiet solutions in random constraint satisfaction problems. *Physical review letters*, 102(23):238701, 2009.

[Lev65] A.Yu. Levin. On an algorithm for the minimization of convex functions. *Sov. Math., Dokl.*, 6:268–290, 1965.

[LV06] László Lovász and Santosh Vempala. Fast algorithms for logconcave functions: Sampling, rounding, integration and optimization. In *FOCS*, pages 57–68, 2006.

[Mas14] Laurent Massoulié. Community detection thresholds and the weak Ramanujan property. In *STOC*, pages 1–10, 2014.

[McS01] Frank McSherry. Spectral partitioning of random graphs. In *FOCS*, pages 529–537, 2001.

[MNS13] Elchanan Mossel, Joe Neeman, and Allan Sly. A proof of the block model threshold conjecture. *arXiv preprint arXiv:1311.4115*, 2013.

[MNS15] Elchanan Mossel, Joe Neeman, and Allan Sly. Reconstruction and estimation in the planted partition model. *Probability Theory and Related Fields*, 162(3-4):431–461, 2015.

[MPRT16] Raffaele Marino, Giorgio Parisi, and Federico Ricci-Tersenghi. The backtracking survey propagation algorithm for solving random k-sat problems. *Nature communications*, 7:12996, 2016.

[MST06] Elchanan Mossel, Amir Shpilka, and Luca Trevisan. On ε-biased generators in NC0. *Random Structures & Algorithms*, 29(1):56–81, 2006.

[NJLS09] Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009.

[NY83] A.S. Nemirovsky and D.B. Yudin. *Problem Complexity and Method Efficiency in Optimization.* J. Wiley @ Sons, New York, 1983.

[O'D11] Ryan O'Donnell. Lecture 13. notes for 15-859 linear and semidefinite programming. Available at
`http://www.cs.cmu.edu/afs/cs.cmu.edu/academic/class/15859-f11/www/notes/lecture13.p`
2011.

[OW14] Ryan O'Donnell and David Witmer. Goldreich's PRG: Evidence for near-optimal polynomial stretch. In *Conference on Computational Complexity*, 2014.

[Rag08] Prasad Raghavendra. Optimal algorithms and inapproximability results for every CSP? In *STOC*, pages 245–254, 2008.

[SD15] Jacob Steinhardt and John C. Duchi. Minimax rates for memory-bounded sparse linear regression. In *COLT*, pages 1564–1587, 2015.

[SSSS09] Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Stochastic convex optimization. In *COLT*, 2009.

[SVW15] J. Steinhardt, G. Valiant, and S. Wager. Memory, communication, and statistical queries. *Electronic Colloquium on Computational Complexity (ECCC)*, 22:126, 2015.

[Tre08] Luca Trevisan. Checking the quasirandomness of graphs and hypergraphs. http://terrytao.wordpress.com/2008/02/15/luca-trevisan-checking-the-quasirandomness-of-graphs-and-hypergraphs/, February 2008.

[TW87] Martin A Tanner and Wing Hung Wong. The calculation of posterior distributions by data augmentation (with discussion). *Journal of the American Statistical Association*, 82:528–550, 1987.

[Č85] V. Černý. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45(1):41–51, January 1985.