Electronic Colloquium on Computational Complexity, Report No. 148 (2014)

On the Complexity of Random Satisfiability Problems with Planted Solutions

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

Abstract

We consider the problem of identifying a planted assignment given a random k-SAT formula consistent with the assignment. This problem exhibits a large algorithmic gap: while the planted solution can always 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, of $\tilde{\Omega}(n^{r/2})$ clauses for *statistical* algorithms [52, 36], matching the known upper bound (which, as we show, can be implemented using a statistical algorithm) [37]. 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 algorithms. It also points to a geometric *paring* phenomenon in the space of all planted assignments that might be of independent interest.

As a consequence, we prove that a strong form of Feige's refutation hypothesis [33] holds for statistical algorithms. Our bounds extend to the planted k-CSP model, defined by Goldreich as a candidate for one-way function [44], and therefore provide concrete evidence for the security of Goldreich's one-way function and the associated pseudorandom generator when used with a sufficiently hard predicate.

[*] IBM Research - Almaden. Part of the work done while visiting the Simons Institute, UC Berkeley

[†] Georgia Tech and the IMA. Supported in part by an NSF postdoctoral fellowship.

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

ISSN 1433-8092

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 random models with planted solutions more generally appear widely in several different forms including network clustering problems with planted partitions (the stochastic block model and its variants), random k-SAT with a planted assignment, and in Goldreich's proposed one-way function from cryptography [44].

It was noted in [11] 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 [50, 2, 57, 55]. Algorithms for planted 3-SAT with various relative proportions were given by Flaxman [38] and Coja-Oghlan *et al.* [24], 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 the 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 for special cases use $\tilde{O}(n^{r/2})$ clauses to identify the planted solution (with the exception of the k-XOR-SAT which can be solved using Gaussian elimination from $O(n)$ clauses but has distribution complexity k). In [37] 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 [44]. 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).

Bogdanov and Qiao [18] show that an SDP-based algorithm of Charikar and Wirth [21] 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 (the folklore birthday "paradox"-based reduction to $r = 2$ is described in [68]). Nearly optimal integrality gaps for LP and SDP hierarchies were recently given for this problem [68] (and references therein) for $\Omega(n^{r/2-\epsilon})$ evaluations of the predicate. The assumption that a decision version of this planted k-CSP is hard is stated as the DCSP hypothesis in [10], and has been used in complexity theory and cryptography [3, 44, 48, 6, 5]. Goldreich's PRG is shown to be an ϵ-biased generator in [65, 7], and lower bounds against DPLL-style algorithms are given in [27]. We note that despite the similarities between these two types of planted k-CSPs we are not aware of an equivalence between these two problems.

1.1 Our 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 algorithms, defined by Kearns in the context of PAC learning [52] and by Feldman *et al.* [36] 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. Such algorithms include most algorithmic approaches used in practice and in theory on a wide variety of problems, including Expectation Maximization (EM) [30], MCMC optimization [71, 41], (generalized) method of moments [47], simulated annealing [53, 74], first and second order methods for linear/convex optimization [32, 13], and many others (see [22, 15, 36] for proofs and other examples).

The simplest form of statistical algorithms are algorithms 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, the strongest form of our lower bounds is for algorithms that use VSTAT oracle defined in [36]. 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. *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 a strengthening of the standard oracle defined by Kearns [52] that uses the same tolerance τ for all query functions.

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 random and uniform k-SAT clauses. 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. *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 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 [37]. 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 the Ellipsoid algorithm that we discuss in Appendix A). 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 exists a natural strengthening of VSTAT and 1-STAT oracles that bridges this gap. Specifically, we extend the oracle 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).

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 statistical algorithm that can solve the problem described in Theorem 1 using less than $\tilde{O}(n^{r/2})$ calls to 1-MSTAT$(n^{r/2})$. We state the upper bound more formally:

Theorem 2. *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$(n^{\lceil r/2 \rceil})$ time linear in this number and identifies the planted assignment with probability $1 - o(1)$.*

We prove this bound by showing that the algorithm from [37] based on a subsampled power iteration can be implemented statistically. 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 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 [38] 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 [18] (with the "birthday paradox" as shown in [68]). This approach can also be implemented statistically, although a direct implementation using a generic statistical SDP solver such as the one we describe in Appendix A 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).

1.2 Corollaries and applications

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 [33]. 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 formulae 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: for even k [39, 26, 46] and $k = 3$ [43, 34], $n^{k/2}$ clauses suffice, while for odd $k \geq 5$, the current best algorithms require $n^{\lceil k/2 \rceil}$ clauses.

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 is that Feige's hypothesis holds for the class of statistical algorithms. The proof (see Thm. 7) 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 [11, 50, 57, 55] 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 [14].

Lower bounds for convex programs: Our lower bounds imply limitations of using convex programs to recover planted solutions: 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 $n^{\Omega(r)}$.

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 essentially follow from statistical implementations of existing algorithms for convex optimization. Details are given in Appendix A.

1.3 Overview of the technique

Our proof of the lower bound builds on the notion of statistical dimension given in [36] which itself is based on ideas developed in a line of work on statistical query learning [52, 17, 35].

Our primary technical contribution is a new, stronger notion of statistical dimension and its analysis for the planted k-CSP problems. The statistical dimension in [36] 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 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 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 [19, 12]).

We remark that while the k-XOR-SAT problem is superficially similar to learning of sparse parities from random uniform examples for which optimal statistical lower bounds are well-known and easy to derive, the problems and the techniques are very different. The primary 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 are relatively large. Further, 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 to capture properties of statistical algorithms. As the distance in this metric increases, the fraction of solutions that can be discarded goes up rapidly from exponentially small to a polynomial fraction. We call this a ‘paring’ transition as a large number of distributions become amenable to being separated and discarded as possible solutions.

We conjecture that our lower bounds hold for *all* algorithms, except in the case of strict constraints of low algebraic degree. Formally, the algebraic degree of a Boolean function over $\mathcal{Z}_2^k$ is the degree of the lowest degree polynomial over $\mathcal{Z}_2^k$ that represents f. For example, the parity function equals to $x_1 + x_2 + \cdots + x_k$ and therefore has algebraic degree 1. A function of algebraic degree d can be viewed as XOR of monomials of degree at most d. Therefore, as is well known, Gaussian elimination can be applied to the values of the function on all $\sum_{i \in [d]} \binom{k}{i}$ monomials to recover the coefficients of the monomials and thereby the function itself (e.g. [62]). Note that this requires at least $\sum_{i \in [d]} \binom{k}{i}$ noiseless constraints. We say that a clause distribution Q induces strict constraints of algebraic degree d if $f(x) = 2^{k-1}Q(x)$ is a $\{0,1\}$ valued function that has algebraic degree d.

Conjecture 1. *For a planted k-SAT problem with distribution complexity r that does not induce strict constraints of algebraic degree $\leq r/2$, $\tilde{\Theta}(n^{r/2})$ clauses are necessary and sufficient for a polynomial-time algorithm to recover the planted solution.*

This conjecture generalizes previous hardness assumptions and conjectures (e.g. [44, 10]) and predicts a precise threshold at which planted CSPs become tractable.

1.4 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 case $k = 2$ 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 [20], McSherry's general-purpose spectral algorithm [61], and Coja-Oghlan's algorithm that works for graphs of constant average degree [23]. Recently an intriguing threshold phenomenon was conjectured by Decelle, Krzakala, Moore, and Zdeborová [29]: 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 [63, 60, 64]. 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 [72] 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 [56, 1]: 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. Shattering has also been used to prove that certain algorithms fail at high enough densities [40].

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 both shattering and paring.

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 [29]; as the gap between impossibility and efficient recovery is only a constant factor there, the paring transition would need to be located more precisely.

1.5 Outline

- In Section 2, we define planted k-CSP problems, their notions of complexity and the associated computational problems. We then define the statistical oracles used by statistical algorithms discussed here.

- In Section 3 we give detailed statements of our main results.

- In Section 4, we define the new statistical dimension and give an overview of its application to our problem. In Section 7, we prove that the statistical dimension gives lower bounds on the complexity of statistical algorithms for the oracles discussed here.

- In Section 5, we give the proof of the bounds on the statistical dimension of planted satisfiability. In the Appendix B we show the extension of these bounds to Goldreich's k-CSP.

- In Section 6 we present corollaries and applications of our main theorems: 1) we give a hierarchy of hard satisfiable k-SAT distributions generalizing the notion of "quiet plantings" 2) we prove that Feige's 3-SAT hypothesis holds for statistical algorithms 3) we show that our lower bounds give a means for generating hard instances for CSP approximation algorithms.

- In Section 8, we describe an efficient statistical algorithm for planted satisfiability that nearly matches our lower bounds.

- Finally, in Appendix A we show that many standard convex optimization techniques, including LP's and SDP's can be implemented statistically.

2 Preliminaries

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 [38, 50, 2, 54, 57, 24, 55].

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 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 one 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 $P(\sigma_{i_1}, \ldots, \sigma_{i_k})$. 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 $P(\sigma_{i_1}, \ldots, \sigma_{i_k})$). 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

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

$\hat{Q}(S)$ 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 with $Q(0) = Q(2) = \delta/2^{k-1}$,

$Q(1) = Q(3) = (2-\delta)/2^{k-1}$, for $\delta \neq 1$. In this case, we have $\hat{Q}(1) = 0$ and $\hat{Q}(2) = 0$. The distribution complexity of Q is therefore $r = 3$. We will see that such parity-type distributions are in fact the hardest for 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 [52] in learning theory, Feldman *et al.* [36] defined statistical 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[1] 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-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 [36]. 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. The multiple 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 [36].

Definition 3 (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 [36]. 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 of the VSTAT oracle to multi-valued functions.

Definition 4 (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 $\{0, \ldots, L-1\}$, MVSTAT(L, t) returns a vector $v \in \mathbb{R}^L$ satisfying for every $Z \in \mathcal{S}$*

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

[1] For simplicity here we only give definitions relevant to Boolean functions and their generalizations.

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$(2, t)$ and any query to MVSTAT(L, t) can be easily answered using L queries to VSTAT$(4 \cdot Lt)$ (Thm. 10). 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 behave in the same way as expected when the algorithm is executed on true samples. 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 Main 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 proofs and statements in this section. In Appendix B we show the extension of our lower bounds to this problem and also make the similarity between the two problems explicit.

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. *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 $D \in \mathcal{D}_Q$ 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$\left(L, \frac{1}{L} \cdot \frac{n^r}{(\log q)^r}\right)$ for a constant $c_2 = \Omega_k(1)$ and any $q \geq 1$.*

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 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 $n^r/(\log n)^{2r}$.

We next turn to our algorithmic results, motivated by two considerations. First, the $O(n^{r/2})$-clause algorithm implicit in [18] 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.

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 4. *Let $\mathcal{Z}_Q$ be a planted satisfiability problem with clause distribution Q having distribution complexity r. Then $\mathcal{Z}_Q$ can be solved using $O(n^{r/2}\log n)$ random clauses and time linear in this number and 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 [37]. 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 on $n/2 + t\sqrt{n}$ variables for an arbitrary constant t.

4 Lower bounds via statistical dimension

Lower bounds on the complexity of statistical algorithms are based on the notion of a *statistical dimension* introduced in [36] on the basis of ideas from [17, 35]. To describe further details on the statistical dimension used in this work, we start with some definitions.

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 simplicity we will focus on decision problems in this section and present extensions to general search problems in Sec. 7. 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 unknown distribution $D' \in \mathcal{D} \cup \{D\}$, whether $D' \in \mathcal{D}$ or $D' = D$.

To prove our bounds we introduce a new, stronger notion 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 $\mathcal{D}'$.

Our concept of statistical dimension is essentially the same as in [36] but uses $\kappa_2(\mathcal{D}', D)$ instead of average correlations.

Definition 5. *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 $\mathrm{SDN}(\mathcal{B}(\mathcal{D}, D), \kappa)$.*

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

Theorem 5. *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 $d = \mathrm{SDN}(\mathcal{B}(\mathcal{D}, D), \kappa)$ 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 $\Omega(d/L)$ calls to MVSTAT$(L, 1/(12 \cdot \kappa^2 \cdot L))$.*
- *Any randomized statistical algorithm that solves $\mathcal{B}(\mathcal{D}, D)$ with probability $\geq 2/3$ 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 to 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 Appendix 7. Our proof relies on techniques from [36] 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 next section we prove the following bound on the statistical dimension with discrimination norm of planted satisfiability.

Theorem 6. *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 5 we obtain our main lower bounds in Theorem 3.

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 (stated in Theorem 6).

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 problem for every $S \subseteq [k]$ of size ℓ applied to $h_S : X_\ell \to \mathbb{R}$ which is a certain averaging projection of h to X_ℓ. Namely, if we denote by $\mathcal{Z}_\ell$ the ℓ-XOR-SAT clause distribution, then 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.

The two key properties of this decomposition are: (i) the coefficients are $\hat{Q}(S)$'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 6. *For $\ell \in [k]$,*

- *Let Z_ℓ be the ℓ-XOR-SAT distribution over $\{\pm1\}^\ell$, 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 $Q : \{\pm1\}^k \to \mathbb{R}$:

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

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:

$$\hat{Q}(S) = \frac{1}{2^k} \sum_{y\in\{\pm1\}^k} Q(y)\chi_S(y)$$

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 1. *For every σ in $\{\pm1\}^n$ and $h : X_k \to \mathbb{R}$,*

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

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 $\ell \geq 1$,

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

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

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

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

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

$$\begin{aligned}
\Delta(\sigma, h) &= \mathop{\mathbb{E}}_{Q_\sigma}[h] - \mathop{\mathbb{E}}_{U_k}[h] \\
&= \frac{2^k}{|X_k|} \sum_{S \subseteq [k] \setminus \{0\}} \hat{Q}(S) \sum_{C_\ell \in X_\ell} \left[(1 - Z_\ell(\sigma(C_\ell))) \cdot \sum_{C \in X_k, C_{|S} = C_\ell} h(C) \right] \\
&= \sum_{S \subseteq [k] \setminus \{0\}} \frac{2^k}{|X_\ell|} \hat{Q}(S) \sum_{C_\ell \in X_\ell} [(1 - Z_\ell(\sigma(C_\ell))) \cdot h_S(C_\ell)] \\
&= 2^k \sum_{S \subseteq [k] \setminus \{0\}} \hat{Q}(S) \left(\mathop{\mathbb{E}}_{U_\ell}[h_S] - \mathop{\mathbb{E}}_{Z_{\ell,\sigma}}[h_S] \right) \\
&= -2^k \sum_{S \subseteq [k] \setminus \{0\}} \hat{Q}(S) \cdot \Gamma_\ell(\sigma, h_S)
\end{aligned}$$

□

We now analyze $\Gamma_\ell(\sigma, h_S)$. For a clause C let $V(C)$ 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,

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

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

Lemma 2. *For $g : X_\ell \to \mathbb{R}$,*

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

Proof.

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

□

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 [19, 12] as done for example in [49, 31].

Lemma 3. *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 4. *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 3.*

Proof. Let $c_0 = \ell \cdot c$. By Lemma 3 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

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

where we used the condition $d \ge e^{\ell}$ 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. *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 4 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 2 we get that

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

□

We are now ready to bound the discrimination norm.

Lemma 6. *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 1 and the definition of r,

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

Hence, by Lemma 5 we get that,

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

By the definition of h_S,

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

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 6) Our reference distribution is the uniform distribution U_k and the set of distributions $\mathcal{D} = \mathcal{D}_Q = \{Q_\sigma\}_{\sigma \in \{\pm 1\}^n}$ 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 6, 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 Corollaries and applications

6.1 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 [11, 50] 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 correspond to distributions with distribution 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 [55], 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). Thus these distributions are intractable over a very wide range of densities.

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.

6.2 Feige's Hypothesis

As a second application of our main result, we show that Feige's 3-SAT hypothesis [33] 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 formulae of n variables and m clauses, where m/n is above the satisfiability threshold, 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 $m = \Omega(n^{3/2})$ [25, 34]. 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 formulae but answers SAT on all formulae 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 7. *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 1. *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 Theorems 1 and 3 is that Feige's hypothesis holds for the class of statistical algorithms.

Theorem 7. *Any (randomized) statistical algorithm that solves the distributional k-SAT refutation problem requires:*

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

2. *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 $c_1 = \Omega_k(1)$.*

Proof. The decision problem in Theorems 1 and 3 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. □

If $r \geq 3$ then 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 stronger conjecture is useful for some hardness of learning results based on Feige's conjecture [28]. For $k = 3$, $\tilde{\Omega}(n^{3/2})$ lower bound on m essentially matches the known upper bounds [25, 34].

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 $r = k = 3$. For the case when $\epsilon = 0$ one can easily obtain satisfiable distributions with $r = k - 1$ that are different from $(k-1)$-XOR-SAT and cannot be solved via Gaussian elimination. We note that Gaussian elimination can be foiled by addition of even small amount of noise. Therefore by mixing the k-XOR-SAT distribution over clauses with an $n^{-1/2}$-fraction of clauses from a satisfiable distribution having $r = k - 1$ as described above, we can obtain a satisfiable distribution for which the lower bound is $\tilde{\Omega}(n^{k/2})$ clauses yet it can no longer be solved using Gaussian elimination. This observation implies the weaker form of Feige's conjecture (the analysis of the lower bound for such a mixture follows easily from the definition of discrimination norm and is omitted).

6.3 Hardness of approximation

We note finally that optimal inapproximability results can be derived from Theorem 3 as well, including the fact that pairwise independent predicates (as studied in [9]) 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 any statistical algorithm 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).

7 Lower Bounds using Statistical Dimension

7.1 Lower bound for VSTAT

We first prove an analogue of lower-bound for VSTAT from [36] 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 [36].

Theorem 8. *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 $\mathrm{VSTAT}(1/(3\kappa^2))$ 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 [35], 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

$$\Phi = \sum_{D_i \in A_k} \left[\left| \mathop{\mathbb{E}}_{D}[h_k(x)] - \mathop{\mathbb{E}}_{D_i}[h_k(x)] \right| \right] = \sum_{D_i \in A_k} |p_k - p_{i,k}|. \tag{4}$$

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. (4) we get that

$$\Phi > |A_k| \cdot \sqrt{\frac{p_k}{3t}} = |A_k| \cdot \sqrt{p_k} \cdot \kappa. \tag{5}$$

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 $\|h_k\|^2 = p_k$. 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. (5) and hence finishes the proof of our claim. □

7.2 General Search Problems

We now show how the statistical dimension and lower bounds can be applied to a broad class of problems referred to as distributional search problems [36]. For a domain X, let $\mathcal{D}$ be a set of distributions over X let $\mathcal{F}$ be a set of solutions and $\mathcal{Z} : \mathcal{D} \to 2^{\mathcal{F}}$ be a map from a distribution $D \in \mathcal{D}$ to a subset of solutions $\mathcal{Z}(D) \subseteq \mathcal{F}$ that are defined to be valid solutions for D. For $t > 0$ the *distributional search problem* $\mathcal{Z}$ over $\mathcal{D}$ and $\mathcal{F}$ using t samples is to find a valid solution $f \in \mathcal{Z}(D)$ given access to t random samples from an unknown $D \in \mathcal{D}$. With slight abuse of notation, for a solution $f \in \mathcal{F}$, we denote by $\mathcal{Z}^{-1}(f)$ the set of distributions in $\mathcal{D}$ for which f is a valid solution.

The statistical dimension for general search problems effectively reduces a general search problem to a hard instance of decision problem.

Definition 8. *For $\kappa > 0$, $\eta > 0$, domain X and a search problem $\mathcal{Z}$ over a set of solutions $\mathcal{F}$ and a class of distributions $\mathcal{D}$ over X, let d be the largest integer such that there exists a* reference *distribution D over X and a finite set of distributions $\mathcal{D}_D \subseteq \mathcal{D}$ with the following property: for any solution $f \in \mathcal{F}$ the set $\mathcal{D}_f = \mathcal{D}_D \setminus \mathcal{Z}^{-1}(f)$ has size at least $(1-\eta) \cdot |\mathcal{D}_D|$ and for any subset $\mathcal{D}' \subseteq \mathcal{D}_f$, where $|\mathcal{D}'| \geq |\mathcal{D}_f|/d$, $\kappa_2(\mathcal{D}', D) \leq \kappa$. The **statistical dimension** with discrimination norm κ and error parameter η of $\mathcal{Z}$ is d and we denote it by* $\mathrm{SDN}(\mathcal{Z}, \kappa, \eta)$.

We now state and prove the generalization of Thm. 8 to search problems.

Theorem 9. *Let X be a domain and $\mathcal{Z}$ be a search problem over a set of solutions $\mathcal{F}$ and a class of distributions $\mathcal{D}$ over X. For $\kappa > 0$ and $\eta \in (0,1)$ let $d = \mathrm{SDN}(\mathcal{Z}, \kappa, \eta)$. Let D be the reference distribution and $\mathcal{D}_D$ be a set of distributions for which the value d is achieved. Any randomized statistical algorithm that, given access to VSTAT$(1/(3\kappa^2))$ oracle for a distribution chosen randomly and uniformly from $\mathcal{D}_D$, succeeds with probability $\gamma > \eta$ over the choice of the distribution and internal randomness requires at least $\frac{\gamma-\eta}{1-\eta}d$ calls to the oracle.*

Proof. As before it suffices to restrict our attention to deterministic algorithms. Let $\mathcal{A}$ be a deterministic statistical algorithm that uses q queries to VSTAT$(1/(3\kappa^2))$ to solve $\mathcal{Z}$ with probability γ over a random choice of a distribution from $\mathcal{D}_D$. 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 f be the output of $\mathcal{A}$.

By the definition of SDN, for $\mathcal{D}_f = \mathcal{D}_D \setminus \mathcal{Z}^{-1}(f)$ it holds that $|\mathcal{D}_f| \geq (1-\eta)|\mathcal{D}_D|$ and for every $\mathcal{D}' \subseteq \mathcal{D}_f$, either $\kappa_2(\mathcal{D}', D) < \kappa$ or $|\mathcal{D}'| \leq |\mathcal{D}_f|/d$. Let the set $\mathcal{D}^+ \subseteq \mathcal{D}_D$ be the set of distributions on which $\mathcal{A}$ is successful. Let $\mathcal{D}_f^+ = \mathcal{D}_f \cap \mathcal{D}^+$ and we denote these distributions by $\{D_1, D_2, \ldots, D_m\}$. We note that $\mathcal{D}_f^+ = \mathcal{D}^+ \setminus (\mathcal{D}_D \setminus \mathcal{D}_f)$ and therefore

$$m = |\mathcal{D}_f^+| \geq |\mathcal{D}^+| - |\mathcal{D}_D \setminus \mathcal{D}_f| \geq \gamma|\mathcal{D}_D| - |\mathcal{D}_D \setminus \mathcal{D}_f| = \frac{\gamma|\mathcal{D}_D| - |\mathcal{D}_D \setminus \mathcal{D}_f|}{|\mathcal{D}_D| - |\mathcal{D}_D \setminus \mathcal{D}_f|}|\mathcal{D}_f| \geq \frac{\gamma-\eta}{1-\eta}|\mathcal{D}_f|. \quad (6)$$

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, $|A_k| \leq |\mathcal{D}_f|/d$.

Combining these two implies that $q \geq d \cdot m/|\mathcal{D}_f|$. By inequality (6), $q \geq \frac{\gamma-\eta}{1-\eta} \cdot d$ giving the desired lower bound.

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 $\mathbb{E}_{D_i}[h_k(x)]$. By the definition of $\mathcal{A}$ and VSTAT(t), this implies that f is a valid solution for $\mathcal{Z}$ on D_i, contradicting the condition that $D_i \in \mathcal{D}_f^+ \subseteq \mathcal{D}_D \setminus Z^{-1}(f)$.

The proof of the second claim is identical to the proof of the analogous claim in Thm. 8 where we use $\mathcal{D}_f$ in place of $\mathcal{D}_D$ (note that in the decision problem $\mathcal{D}_D$ is precisely the set of distributions for which the output of $\mathcal{A}$ is not a valid solution). □

This generalization allows us to view decision problems as search problems where the solution set size parameter is $\eta = 1/2$ (to make this entirely formal we would need to view the set of input distributions as a multiset in which one half of the distributions is D and the other half is $\mathcal{D}$).

7.3 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 Theorems 8 and 9.

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 10. *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 $Z \subseteq L_0$,

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

where $p_Z = \Pr_D[h(x) \in Z]$. □

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

Theorem 11. *Let X be a domain and $\mathcal{Z}$ be a search problem over a set of solutions $\mathcal{F}$ and a class of distributions $\mathcal{D}$ over X. For $\kappa > 0$ and $\eta \in (0, 1)$, let $d = \mathrm{SDN}(\mathcal{Z}, \kappa, \eta)$. Any (possibly randomized) statistical algorithm that solves $\mathcal{Z}$ with probability γ requires at least m calls to 1-MSTAT(L) for*

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

In particular, if $\eta \leq 1/2$ then 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 12. *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 [36]. Their result is easy to generalize to the statement of Theorem 12 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 [36]) imply Theorem 12. 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. *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 9 and 12 to obtain the claimed lower bound for statistical algorithms using MVSTAT.

Proof of Theorem 11. Assuming the existence of a statistical algorithm using less than m samples we apply Theorem 12 for $\delta = (\gamma - \eta)/2$ to simulate the algorithm using VSTAT. The bound on m ensures that the resulting algorithm uses less than $\Omega\left(\frac{d(\gamma-\eta)}{(1-\eta)}\right)$ queries to VSTAT($\frac{1}{3\kappa^2}$) and has success probability of at least $(\gamma + \eta)/2$. By substituting these parameters into Theorem 9 we obtain a contradiction. □

Finally we state an immediate corollary of Theorems 9,10 and 11 that applies to general search problems and generalizes Theorem 5.

Theorem 13. *Let X be a domain and $\mathcal{Z}$ be a search problem over a set of solutions $\mathcal{F}$ and a class of distributions $\mathcal{D}$ over X. For $\kappa > 0$, let $d = \mathrm{SDN}(\mathcal{Z}, \kappa, 1/2)$ and let $L \geq 2$ be an integer. Any randomized statistical algorithm that solves $\mathcal{Z}$ 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 4. The algorithm is a variant of the subsampled power iteration from [37] 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 [37] 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 $Q^\delta : \{\pm 1\}^r \to \mathbb{R}^+$ 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 $S \subset \{1, \dots k\}$ so that $\hat{Q}(S) \neq 0$.

From here on the distribution on clauses will be given by Q^δ_σ, for $\delta \neq 1$ and planted assignment σ. For ease of analysis, we define $Q_{\sigma,p}$ as the distribution over k-clause formulae 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 generalized adjacency matrix M of size $N_1 \times N_2$ with $N_1 = 2^{\lceil k/2 \rceil} \frac{n!}{(n-\lceil k/2 \rceil)!}$, $N_2 = 2^{\lfloor k/2 \rfloor} \frac{n!}{(n-\lfloor k/2 \rfloor)!}$, and $N = \sqrt{N_1 N_2}$. 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 $\hat{M}(\mathcal{F})$ 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 $(l_{\lceil k/2 \rceil + 1}, \dots, l_k)$.

We define the distribution $M_{\sigma,p}$ 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 $u \in \{\pm 1\}^N$ 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, we define the analogous vectors $u_y \in \{\pm 1\}^{N_1}$ and $u_x \in \{\pm 1\}^{N_2}$, 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. From these vectors the partition into true and false literals can be determined.

For even k, the discrete power iteration begins by sampling a random vector $x^0 \in \{\pm 1\}^N$ 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. The rounding is probabilistic and depends on the value of each coordinate and the maximum value of all the coordinates. For odd k, we begin with a random $x^0 \in \{\pm 1\}^{N_2}$ 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 a randomized ± 1 rounding of $M^T_{\sigma,p} y^0$, and repeat. There is a final rounding step to find a ± 1 vector that matches u or u_x.

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$(O(n^{\lceil r/2\rceil}), O(n^{r/2}\log n))$; with $O(n^{\lceil r/2\rceil})$ subsets;
4. For odd r: using $O(\log n)$ calls to MVSTAT$(O(n^{\lfloor r/2\rfloor}), O(n^{\lceil r/2\rceil}\log n))$.

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

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

2. Let $x = Mx^{\log N}$ and set $u^* = \mathsf{sign}(x)$ by rounding each coordinate to its sign.
3. Output the solution of the parity equations defined by u^* using Gaussian elimination.

Lemma 8. *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 [4]).

Proposition 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 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 for simplicity 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 2, note that there are N entries in each row of M half with probability δp and 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 $1 - o(N^{-1})$,

$$\begin{aligned} \max_j |z_j| &\le (\delta-1)p \cdot (u \cdot x) + \frac{(\delta-1)Np}{26} + O(n^{k/2-1}p) && (7) \\ &\le (\delta-1)p \cdot (u \cdot x) + \frac{(\delta-1)Np}{25} && (8) \end{aligned}$$

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 if 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 the $2(\delta-1)p$ times the dot product of u and x restricted to the coordinates A_j. Summing over all $j \in [N]$ an using symmetry 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 2 to $(u \cdot z)$ (with $\mathbb{E}\, Y = N^2 p$, and $\alpha = \frac{(\delta-1)(u \cdot x)}{2N}$), we get

$$\begin{aligned} \Pr[(u \cdot z) < N(\delta-1)p(u \cdot x)/2] &\le \exp\left[-\frac{N^2 p(\delta-1)^2(u \cdot x)^2}{12N^2}\right] && (9) \\ &= \exp\left[-\frac{K \log N (u \cdot x)^2}{12N}\right] = o\left(\frac{1}{N}\right). && (10) \end{aligned}$$

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

$$\begin{aligned}
\mathbb{E}\, Z &\geq \frac{N}{2} + \frac{N(\delta - 1)p(u \cdot x)}{8(\delta - 1)Np/25} \\
&\geq \frac{N}{2} + 3(u \cdot x)
\end{aligned}$$

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

$$\begin{aligned}
\mathbb{E}\, Z &\geq \frac{N}{2} + \frac{N(\delta - 1)p(u \cdot x)}{8(\delta - 1)p(u \cdot x)} \\
&= \frac{5N}{8}
\end{aligned}$$

Note that the variance of Z is at most $N/4$. By Chebyshev, 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 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 $\log \log N$ good steps, and after any such run we will have with certainty $|u \cdot x| \geq \sqrt{N} \log \log N$, completing the first phase.

Once we have $|u \cdot x| \geq \sqrt{N} \log \log N$, then according to Proposition 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 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 $i = 1, \dots \log N$, 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 to get x^i as follows:
$$x^i_j = \begin{cases} \mathsf{sign}(\overline{x}_j) \text{ with probability } \frac{1}{2} + \frac{|\overline{x}_j|}{2\max_j |\overline{x}_j|} \\ -\mathsf{sign}(\overline{x}_j) \text{ otherwise.} \end{cases}$$
2. Set $u^* = \mathsf{sign}(x^{\log N})$ by rounding each coordinate to its sign.
3. Output the solution of the parity equations defined by u^* using Gaussian elimination.

Lemma 9. *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 4. *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. *$|y^{i+1} \cdot u_y| \geq N\beta \log N$ and*
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

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

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

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

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 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

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

Another application of Proposition 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 $1 - o(N^{-2})$,

$$\begin{aligned}y' \cdot u_y &\geq N_1 p(\delta-1)(x \cdot u_x)/2 && (11)\\ &= \frac{N\beta C \log N}{2(\delta-1)} && (12)\\ &\geq N\beta \log N && (13)\end{aligned}$$

□

Proposition 5. *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. As in the proof of Proposition 3.

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 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,

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

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

$$\begin{aligned}\mathbb{E}\, Z &\geq \frac{N_2}{2} + \frac{N_2(\delta-1)p(u_y \cdot y)}{8N^2p^2/\sqrt{\log N}} \\ &= \frac{N_2}{2} + \frac{N_2\gamma(\delta-1)^3}{8K\sqrt{\log N}}\end{aligned}$$

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

$$\begin{aligned}\mathbb{E}\, Z &\geq \frac{N_2}{2} + \frac{N_2(\delta-1)p(u \cdot x)}{8(\delta-1)p(u \cdot x)} \\ &= \frac{5N_2}{8}\end{aligned}$$

Another application of Proposition 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 4 and 5 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 $i\ inU_x^+$, $\mathbb{E}(M^T y)_i \geq (\delta-1)Np\log N/9$. Using Proposition 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 4 by showing how to implement the above algorithms with the statistical oracles 1-MSTAT and MVSTAT.

Lemma 10 (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 with directly querying 1-MSTAT. Here we give a direct simulation with smaller t.

For $t = N \log \log N$, versions of equations (7) and (9) (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 11 (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 $1 - o(1)$. There is a randomized algorithm that makes $O(\log N)$ calls to the MVSTAT($N_2, N_1 logN$) oracle and returns the planted assignment with probability $1 - o(1)$.*

Proof. We run the algorithm using 1-MSTAT, we alternately query 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 the first part of Theorem 4(4). To see that the algorithm converges as claimed we note that Prop. 4 continues to hold, with a lower order correction term in Equation (11) 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 subsets corresponding to the positive support and negative support of u_y.

To obtain the second implementation with smaller value of $t \cdot L$, we use a direct argument. For the step $M^T y = x$, we proceed exactly as above. The queries are of length N_2.

For the step $Mx = y$, we break the queries into $N_1/N_2 = \Theta(n)$ blocks of length N_2 each. For a block B_l we define the restricted inner product $(u_y \cdot v)_l := \sum_{j \in B_l} (u_y)_j \cdot v_j$. We have the following bounds:

$$
\begin{aligned}
|v_i| &\le \frac{(\delta-1)\beta}{N_1} + \min\left\{\frac{1}{t}, 2\sqrt{\frac{1}{N_1 t}}\right\} \\
&\le \frac{(\delta-1)\beta}{N_1} + \frac{1}{\sqrt{N_1 t}}
\end{aligned}
$$

and

$$
\begin{aligned}
(u_y \cdot v)_l &\ge \frac{(\delta-1)\beta}{N_1/N_2} - \min\left\{\frac{1}{t}, \sqrt{\frac{N_2/N_1}{t}}\right\} \\
&\le \frac{(\delta-1)\beta}{N_1/N_2} + O\left(\frac{1}{n^{(k+3)/k}\sqrt{\log N}}\right) \\
&= \frac{(\delta-1)\beta}{N_1/N_2}(1 + o(1))
\end{aligned}
$$

for $\beta \ge \frac{1}{\sqrt{N_2}\log\log N}$. And so summing over all blocks B_l, $(u_y \cdot v) = (\delta - 1)\beta(1 + o(1))$.

Now we randomly round v to a ± 1 vector $y$as follows: let $y_j = \text{sign}(v_j)$ with probability $\frac{1}{2} + \frac{|v_j|}{2\max_j |v_j|}$ and $-\text{sign}(v_j)$ otherwise. If Z is the number of coordinates on which y and u_y agree, we have $\mathbb{E}\, Z \ge \frac{N_1}{2} + \frac{(\delta-1)\beta}{2\left(\frac{(\delta-1)\beta}{N_1} + \frac{2}{N_1\sqrt{\log N}}\right)}$. For $\frac{(\delta-1)\beta}{N_1} \le \frac{2}{N_1\sqrt{\log N}}$, we have

$$
\mathbb{E}\, Z \ge \frac{N_1}{2} + \frac{(\delta-1)\beta}{\frac{8}{N_1\sqrt{\log N}}} = \frac{N_1}{2} + \frac{(\delta-1)\beta\sqrt{\log N}}{8} N_1
$$

And for $\frac{(\delta-1)\beta}{N_1} \geq \frac{2}{N_1\sqrt{\log N}}$, we have

$$\mathbb{E}\, Z \geq \frac{N_1}{2} + \frac{(\delta-1)\beta}{4(\delta-1)\beta/N_1} = \frac{3N_1}{4}$$

The proof is then completed as in the even case. □

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.

References

[1] Dimitris Achlioptas and Amin Coja-Oghlan. Algorithmic barriers from phase transitions. In *Foundations of Computer Science, 2008. FOCS'08. IEEE 49th Annual IEEE Symposium on*, pages 793–802. IEEE, 2008.

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

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

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

[5] Benny Applebaum. Pseudorandom generators with long stretch and low locality from random local one-way functions. In *Proceedings of the 44th symposium on Theory of Computing*, pages 805–816. ACM, 2012.

[6] Benny Applebaum, Boaz Barak, and Avi Wigderson. Public-key cryptography from different assumptions. In *Proceedings of the 42nd ACM symposium on Theory of computing*, pages 171–180. ACM, 2010.

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

[8] David Applegate and Ravi Kannan. Sampling and integration of near log-concave functions. In *STOC*, pages 156–163, 1991.

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

[10] Boaz Barak, Guy Kindler, and David Steurer. On the optimality of semidefinite relaxations for average-case and generalized constraint satisfaction. In *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*, pages 197–214. ACM, 2013.

[11] 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.

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

[13] Alexandre Belloni, Robert M. Freund, and Santosh Vempala. An efficient rescaled perceptron algorithm for conic systems. *Math. Oper. Res.*, 34(3):621–641, 2009.

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

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

[16] 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.

[17] 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 *Proceedings of STOC*, pages 253–262, 1994.

[18] Andrej Bogdanov and Youming Qiao. On the security of goldreich's one-way function. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 392–405. Springer, 2009.

[19] 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.

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

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

[22] 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 *Proceedings of NIPS*, pages 281–288, 2006.

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

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

[25] 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.

[26] 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.

[27] 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.

[28] 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.

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

[30] 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.

[31] 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.

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

[33] Uriel Feige. Relations between average case complexity and approximation complexity. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 534–543. ACM, 2002.

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

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

[36] Vitaly Feldman, Elena Grigorescu, Lev Reyzin, Santosh Vempala, and Ying Xiao. Statistical algorithms and a lower bound for planted clique. In *Proceedings of the 45th annual ACM symposium on Symposium on theory of computing*, pages 655–664. ACM, 2013.

[37] Vitaly Feldman, Will Perkins, and Santosh Vempala. Subsampled power iteration: a unified algorithm for block models and planted csp's. *arXiv preprint arXiv:1407.2774*, 2014.

[38] Abraham Flaxman. A spectral technique for random satisfiable 3cnf formulas. In *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 357–363. Society for Industrial and Applied Mathematics, 2003.

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

[40] 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.

[41] 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.

[42] 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.

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

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

[45] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization.* Springer, 1988.

[46] 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.

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

[48] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Cryptography with constant computational overhead. In *Proceedings of the 40th annual ACM symposium on Theory of computing*, pages 433–442. ACM, 2008.

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

[50] Haixia Jia, Cristopher Moore, and Doug Strain. Generating hard satisfiable formulas by hiding solutions deceptively. In *PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE*, volume 20, page 384. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2005.

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

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

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

[54] Michael Krivelevich and Dan Vilenchik. Solving random satisfiable 3cnf formulas in expected polynomial time. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 454–463. ACM, 2006.

[55] Florent Krzakala, Marc Mézard, and Lenka Zdeborová. Reweighted belief propagation and quiet planting for random k-sat. *arXiv preprint arXiv:1203.5521*, 2012.

[56] 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.

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

[58] László Lovász. *An algorithmic theory of numbers, graphs and convexity*, volume 50. SIAM, 1987.

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

[60] Laurent Massoulié. Community detection thresholds and the weak ramanujan property. In *STOC 2014: 46th Annual Symposium on the Theory of Computing*, pages 1–10, 2014.

[61] Frank McSherry. Spectral partitioning of random graphs. In *Foundations of Computer Science, 2001. Proceedings. 42nd IEEE Symposium on*, pages 529–537. IEEE, 2001.

[62] E. Mossel, R. O'Donnell, and R. Servedio. Learning functions of k relevant variables. *Journal of Computer and System Sciences*, 69(3):421–434, 2004.

[63] Elchanan Mossel, Joe Neeman, and Allan Sly. Stochastic block models and reconstruction. *arXiv preprint arXiv:1202.1499*, 2012.

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

[65] Elchanan Mossel, Amir Shpilka, and Luca Trevisan. On ε-biased generators in nc0. *Random Structures & Algorithms*, 29(1):56–81, 2006.

[66] 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.

[67] 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.pdf`, 2011.

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

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

[70] Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Stochastic convex optimization. In *COLT 2009 - The 22nd Conference on Learning Theory*, 2009.

[71] 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.

[72] 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.

[73] Paul Valiant. Distribution free evolvability of polynomial functions over all convex loss functions. In *ITCS*, pages 142–148, 2012.

[74] 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.

A Statistical Algorithms for Solving CSPs

In this section we provide high level details of statistical implementation of several algorithmic tools used in the context of constraint satisfaction problems.

We will focus on convex programs.

A.1 Canonical LPs/SDPs for k-CSPs

We first describe the standard way to relax Boolean constraint satisfaction problems to an LP or SDP. The classic SDP for a constraint satisfaction problem is the MAX-CUT SDP of Goemans and Williamson [42]. 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. This SDP is also applied to solve instances of MAX-CUT in which constraints are sampled from a distribution such as in the case of a stochastic block model. Standard concentration bounds imply that the maximum achieved on randomly sampled edges is very close to the maximum achieved for the expectation $\mathbb{E}_{e_{ij}\sim D}[e_{ij}(1-x_{i,j})]$, where e_{ij} is a randomly sampled edge from some distribution over edges D and x is as before. In fact, maximizing $\mathbb{E}_{e_{ij}\sim D}[e_{ij}(1-x_{i,j})]$ is the true objective of such SDPs for which it suffices to maximize it on the random samples.

More generally, the canonical LP relaxation of a k-CSP with m constraints results in the following program (see [67] for a textbook version or [69, 10, 68] for some applications):

$$\text{maximize} \mathop{\mathbb{E}}_{i\sim[m]}\left[\sum_{y\in\{\pm\}^k, y \text{ satisfies } R_i} x_{V_i,y}\right],$$

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 (with additional PSD cone constraints in the case of SDPs).

If the constraints (R,V) are sampled randomly from some distribution D then this gives the following convex program:

$$\max_{\bar{x}\in K} \mathop{\mathbb{E}}_{(R,V)\sim D}\left[\sum_{y\in\{\pm\}^k, y \text{ satisfies } R} x_{V,y}\right]. \tag{14}$$

Note that if the distribution D is uniform over some (arbitrarily chosen) set of m constraints then we get a convex program that is identical to the regular (non-distributional version). And in the other direction, solving a (non-distributional) instance obtained by i.i.d. sampling of constraints from D gives a solution with a value close to the optimal one for the distributional problem (this normally happens after seeing a linear in n number of constraints and hence applies to our setting). This distributional formulation allows using statistical algorithms for solving worst-case CSPs.

Our lower bounds apply to distributional CSPs in which constraints are sampled from a distribution D. As can be seen from the example and the general form, to solve such CSPs we need to solve the following class of convex programs $\min_{x \in K} \mathbb{E}_{w \sim D}[f(x,w))]$, where K is a fixed convex N-dimensional set (that is not dependent the distribution D) and $f(x,w)$ is a bounded convex function (in the examples above the objective was a linear function, its negation gives a convex function that we need to minimize). More generally, such formulations arise whenever the objective function in the convex relaxation is a sum of objectives for individual constraints and the domain does not depend on the Boolean constraints. Such programs are referred to as *stochastic convex programs* and are well-studied in machine learning and optimization (e.g. [66, 70]).

We now show that several standard algorithms for solving convex programs give statistical algorithms for problems of this type.

The Ellipsoid algorithm: We first outline a statistical algorithm based on the Ellipsoid algorithm with a weak evaluation oracle as shown by Lovász [58] (similar application of this result appears in [73]). For simplicity, we omit some technical conditions on K which hold for K's that arise in the common constraint satisfaction and combinatorial optimization relaxations.

Theorem 14. *For the convex program (14) there exists a statistical algorithm that for any distribution D over $\mathbb{R}^N$ and $\epsilon > 0$, with probability at least $2/3$, outputs $x \in \mathbb{R}^N$ such that $x \in K$ and $\mathbb{E}_D[f(x,w)] \leq OPT + \epsilon$, where OPT is the optimal value of the convex program. The algorithm runs in time $\mathrm{poly}(N, \frac{1}{\epsilon}, B)$ using queries to $VSTAT(\mathrm{poly}(N, \frac{1}{\epsilon}, B))$.*

Proof. We use fundamental results from [58, 45] that show that a weak membership oracle for a convex set can be used to obtain a "very weak" separation oracle that is sufficient for running the ellipsoid algorithm. This in turn implies that it is possible to weakly minimize a convex function over a convex set given a weak evaluation oracle.

More precisely, a ν-weak evaluation oracle for a function $F(x)$ is the oracle that for every $x \in \mathbb{R}^N$, returns value v such that $|v - F(x)| \leq \nu$. An ϵ-weak optimization algorithm for F over a convex K is the algorithm that returns x such that $F(x) \leq OPT + \epsilon$, where OPT is the minimal value of F on K. The results in [58](Theorem 2.2.15) imply that, under certain mild conditions on K, for every bounded F and $\epsilon > 0$, there exists a randomized $\mathrm{poly}(N, \frac{1}{\epsilon}, B)$-time ϵ-weak optimization algorithm for F that uses a ν-weak evaluation oracle for F, where $\nu \geq 1/\mathrm{poly}(N, \frac{1}{\epsilon}, B)$.

In our case $F(x) = \mathbb{E}_D[f(x,w)]$ and therefore ν-weak evaluation oracle for $F(x)$ can be obtained using the VSTAT oracle with $t = \frac{1}{\nu^2}$. □

Random Walks: Here we give a more efficient algorithm using sampling that effectively rules out LP/SDP relaxations formulated in dimension at most $(n/\log^2 n)^{r/2}$.

We note that a strong membership oracle is available for K, and by means of VSTAT, a ν-approximate evaluation oracle is available for the objective function.

Theorem 15. *For any convex program $\min \mathbb{E}_{w \sim D}[f(x,w)], x \in K$ in $\mathbb{R}^N$ given by a membership oracle for K with the guarantee that $rB_n \subseteq K \subseteq RB_n$, there is a statistical algorithm that with probability at least $2/3$ outputs a point $x \in \mathbb{R}^N$ s.t. $\mathbb{E}_D[f(x,w)] \leq OPT + \epsilon$ in time $\mathrm{poly}(n, \frac{1}{\epsilon}, B)$ using queries to $VSTAT(O(N^2/\epsilon^2))$.*

Proof. Let $F(x) = \mathbb{E}_D[f(x, w)]$. The basic idea is to sample from a distribution that has most of its measure on points with $F(x) \leq OPT + \epsilon$. To do this, we use the random walk approach as in [51, 59] with a minor extension. The algorithm performs a random walk whose stationary distribution is proportional to $g(x) = e^{-\alpha F(x)}$. Each step of the walk is a function evaluation, i.e., a call to VSTAT. Noting that $e^{-\alpha F(x)}$ is a logconcave function, the number of steps is $\mathrm{poly}(n, \log \alpha, \delta)$ to get a point from a distribution within total variation distance δ of the target distribution. Further, as shown in [51], a random point from the target distribution satisfies

$$\mathbb{E}[F(x)] \leq \min_{x \in K} F(x) + N/\alpha.$$

Thus, setting $\alpha = N/\epsilon$ suffices.

Now we turn to the extension, which arises because we can only evaluation $F(x)$ approximately through the oracle. We do this using VSTAT(N^2/ϵ^2) and assume that VSTAT is consistent in its answers (i.e., returns the same value on the same query and parameter t value). The value returned $\tilde{F}(x)$ satisfies $|F(x) - \tilde{F}(x)| \leq \epsilon/N$. The stationary distribution is now proportional to $\tilde{g}(x) = e^{-\alpha \tilde{F}(x)}$ and satisfies

$$\frac{\tilde{g}(x)}{g(x)} \leq e^{-\alpha(\tilde{F}(x) - F(x))} \leq e^{\alpha \frac{\epsilon}{N}} \leq e.$$

Moreover, $\tilde{F}(x)$ is approximately convex, i.e., for any $x, y \in K$ and any $\lambda \in [0, 1]$, we have

$$\tilde{F}(\lambda x + (1 - \lambda) y) \leq \lambda \tilde{F}(x) + (1 - \lambda) \tilde{F}(y) + \epsilon.$$

This, as first shown by Applegate and Kannan [8], increases the convergence time of the random walk by only a factor of $e^{2\epsilon}$. □

Gradient descent and other techniques: In most practical cases the objective function given in formulation (1) is optimized using simpler methods such as iterative first-order methods and simulated annealing. It is easy to see that such methods can be implemented using VSTAT oracle. For example gradient descent relies solely on knowing $\nabla F(x_m)$ approximately, where F is the optimized function and x_m is the solution at step m. By linearity of expectation we know that $\nabla \mathbb{E}_D[f(x_m, w)] = \mathbb{E}_D[\nabla f(x_m, w)]$. This means that using access to VSTAT we can approximate $\nabla \mathbb{E}_D[f(x_m, w)]$ with inverse-polynomial error. The same approach can be used to implement second-order methods and most other local search heuristics.

Corollaries for planted k-CSPs: The significance of these theorems for solving planted k-CSPs is the following. Consider a convex program (e.g. LP/SDP) in the above general form and dimension N. By our upper bounds such convex program can be solved to accuracy ϵ using a polynomial number of queries to VSTAT($O(N^2/\epsilon^2)$) (in the case of the Random Walk algorithm). For small enough constant ϵ the value of the solution for a planted k-CSP would be different from a solution for the uniform distribution over constraints, solving our decision problem. This would contradict our lower bound for any $N < c(n/\log^2 n)^{r/2}$.

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 affect the bounds on the number of queries and tolerance we gave above. In other words, for the purposes of our bounds only the number of variables used in the objective matters.

Note that this is a concrete lower bound on the dimension of the convex program (in the general form given above) that can be used to solve a distributional k-CSP that does not make any assumptions about how the convex program is solved. In particular, it does not need to be solved via a statistical algorithm or efficiently. We are not aware of this form of lower bounds against convex programs stated before.

It is important to note that the many standard SDPs for k-CSPs (e.g. [69, 10]) optimize over $n^{\theta(k)}$-dimensional space. The number of samples needed by VSTAT in Theorem 14 has polynomial dependence on the dimension and therefore our lower bound of $\tilde{\Omega}(n^r)$ cannot rule out the use of such programs (and indeed lift-and-project hierarchies imply that for any k-CSP there exists a convex program of dimension $n^{\theta(k)}$ whose optimum would give an optimal solution). Finally, we make the observation that if a decision problem between a uniform distribution and a planted one formulated as a convex program cannot be solved by a statistical algorithm then solution values for those two cases must be indistinguishable. This implies strong integrality gaps for any such convex program.

A.2 Satisfying most constraints

We now describe methods that can be used for finding feasible solutions when each Boolean constraint is mapped to a single linear/SDP constraint. Satisfying a constraint with probability $1-\epsilon$ implies that $1/\epsilon$ constraints randomly drawn from the distribution will be satisfied with probability $> 1/3$. We use B to denote the bit complexity of each of the elements in the support of D.

We will say that D is a feasible distribution if there exists $x \in \mathbb{R}^n \setminus \{0^n\}$ such that for every a in the support of D, $a \cdot x \geq 0$. The following result for solving most constraints coming from any distribution was proved by Dunagan and Vempala [32].

Theorem 16 ([32]). *There exists a statistical algorithm* `HS-DV` *that for any feasible distribution D over $\mathbb{R}^n$, $\epsilon > 0$* `HS-DV` *outputs $x \in \mathbb{R}^n \setminus \{0^n\}$ such that $\Pr_{a\sim D}[a \cdot x \geq 0] \geq 1-\epsilon$. The algorithm runs in time polynomial in n,$1/\epsilon$ and B and uses queries to VSTAT*$(\mathrm{poly}(n, 1/\epsilon, B))$.

We note that in [32] this algorithm is described as manipulating examples and not using statistical queries. However the implementation of the algorithm is based on variants of the Perceptron algorithm for which an implementation using statistical queries is well-known [16]. Roughly, a Perceptron algorithm updates its current solution given a counterexample. However the centroid of (positive or negative) counterexamples is still a (positive or negative) counterexample and can be found via n statistical queries. The complexity of the algorithm in [32] is also stated in terms of the radius of the largest ball that fits in the feasible region which implies our formulation.

Belloni *et al.* [13] extended the algorithm in [32] to SDPs (and other classes of conic programming). The result they obtain is similar to Theorem 16 but with vector x being restricted to represent a $d \times d$ PSD matrix (where $n = d^2$) and feasibility defined by existence of a PSD matrix satisfying the constraints in the support of D.

These algorithms can be easily adapted to finding solutions for non-homogeneous constraints (by adding another variable) and then to optimization by adding a fixed constraint to a distribution (say with weight 1/2) and using it to perform binary search for a solution.

B Extension to Goldreich's planted k-CSP

We now show essentially identical lower bounds for the planted k-CSP problem which is the basis of Goldreich's one-way function candidate. Specifically, we prove the analogue of Theorem 6 for this problem, which in turn automatically implies that the lower bounds stated in Theorem 3 apply to this problem verbatim.

Let $P : \{\pm 1\}^k \to \{0,1\}$ be a Boolean predicate on k-bits. 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. Let $\mathcal{D}_P$ denote the set of all distributions P_σ, where $\sigma \in \{\pm 1\}^k$ 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 $D = U'_k$.

Theorem 17. *For any predicate P of complexity r, there exist a constant $c > 0$ (that depends on P) such that for any $q \geq 1$,*

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

The proof follows exactly the same approach. For a distribution P_σ and query function $h : X'_k \to \mathbb{R}$, we denote by $\Delta(\sigma, h) = \mathbb{E}_{P_\sigma}[h] - \mathbb{E}_{U'_k}[h]$. Our goal is to first decompose $\Delta(\sigma, h)$ into a linear combination of the differences in expectations of h evaluated on XOR predicate distributions for σ. We will need the following notation

Definition 9. *For $\ell \in [k]$,*

- *Let Z_ℓ be the ℓ-XOR predicate over $\{\pm 1\}^\ell$.*
- *For $C \in Y_k$ and $S \subseteq [k]$ of size ℓ let $C_{|S}$ denote an ℓ-tuple of variables in Y_ℓ consisting of variables in 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, $b \in \{\pm 1\}$ and $C_\ell \in Y_\ell$, let*

$$h_S(C_\ell, b) = \frac{|X'_\ell|}{|X'_k|} \sum_{C \in Y_k,\ C_{|S} = C_\ell} h(C, b).$$

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

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 12. *For every σ in $\{\pm 1\}^n$ and $h : X'_k \to \mathbb{R}$,*

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

Proof. For a variable tuple C we denote by $\sigma(C)$ the vector in $\{\pm 1\}^k$ that gives evaluation of the variables in C on σ. Also by our definitions,

$$P_\sigma(C, b) = \frac{b \cdot P(\sigma(C)) + 1}{|X'_k|} = \frac{b \cdot P(\sigma(C))}{|X'_k|} + U'_k(C, b).$$

Now, using ℓ to denote $|S|$,

$$\begin{aligned}\Delta(\sigma,h) &= \mathop{\mathbb{E}}_{P_\sigma}[h] - \mathop{\mathbb{E}}_{U'_k}[h] = \sum_{(C,b)\in X'_k} h(C,b)\cdot(P_\sigma(C,b) - U'_k(C,b)) = \frac{1}{|X'_k|}\sum_{(C,b)\in X'_k} h(C,b)\cdot b\cdot P(\sigma(C)) \\ &= \frac{1}{|X'_k|}\sum_{S\subseteq[k]}\hat{P}(S)\sum_{(C,b)\in X'_k} h(C,b)\cdot b\cdot \chi_S(\sigma(C)) \\ &= \frac{1}{|X'_k|}\sum_{S\subseteq[k]}\hat{P}(S)\sum_{C_\ell\in Y_\ell}\sum_{(C,b)\in X'_k, C_{|S}=C_\ell} h(C,b)\cdot b\cdot\chi_S(\sigma(C)) \qquad (15)\end{aligned}$$

Note that if $C_{|S} = C_\ell$ then

$$\chi_S(\sigma(C)) = \chi_{[\ell]}(\sigma(C_\ell)) = Z_\ell(\sigma(C_\ell)).$$

Therefore,

$$\sum_{(C,b)\in X'_k, C_{|S}=C_\ell} h(C,b)\cdot b\cdot\chi_S(\sigma(C)) = Z_\ell(\sigma(C_\ell))\cdot\sum_{(C,b)\in X'_k, C_{|S}=C_\ell} h(C,b)\cdot b.$$

Plugging this into eq.(15) we obtain

$$\begin{aligned}\Delta(\sigma,h) &= \frac{1}{|X'_k|}\sum_{S\subseteq[k]}\hat{P}(S)\sum_{C_\ell\in Y_\ell} Z_\ell(\sigma(C_\ell))\cdot\sum_{(C,b)\in X'_k, C_{|S}=C_\ell} h(C,b)\cdot b \\ &= \sum_{S\subseteq[k]}\frac{\hat{P}(S)}{|X'_\ell|}\sum_{C_\ell\in Y_\ell}\left[Z_\ell(\sigma(C_\ell))\cdot\sum_{b\in\{\pm1\}} h_S(C_\ell,b)\cdot b\right] \\ &= \sum_{S\subseteq[k]}\frac{\hat{P}(S)}{|X'_\ell|}\sum_{(C_\ell,b)\in X'_\ell}\left[Z_\ell(\sigma(C_\ell))\cdot b\cdot h_S(C_\ell,b)\right] \\ &= \sum_{S\subseteq[k]}\hat{P}(S)\left(\mathop{\mathbb{E}}_{U'_\ell}[h_S] - \mathop{\mathbb{E}}_{Z_{\ell,\sigma}}[h_S]\right) \\ &= \sum_{S\subseteq[k]}\hat{P}(S)\cdot\Gamma_\ell(\sigma,h_S)\end{aligned}$$

□

We now show that in this version $\Gamma_\ell(\sigma,h_S)$ is also a degree ℓ polynomial. For a tuple of variables C let $V(C)$ denote the set of indices of variables in C. By definition, $Z_\ell(\sigma(C)) = \chi_{V(C)}(\sigma)$. This implies that $\Gamma_\ell(\sigma,h_S)$ can be represented as a linear combination of parities of length ℓ.

Lemma 13. *For $g : X'_\ell \to \mathbb{R}$,*

$$\Gamma_\ell(\sigma,g) = \frac{1}{|X'_\ell|}\sum_{A\subseteq[n],|A|=\ell}\left(\sum_{(C_\ell,b)\in X'_\ell, V(C_\ell)=A} g(C_\ell,b)\cdot b\right)\cdot\chi_A(\sigma).$$

Proof.

$$\begin{aligned}\Gamma_\ell(\sigma, g) &= \mathop{\mathbb{E}}_{Z_{\ell,\sigma}}[g] - \mathop{\mathbb{E}}_{U'_\ell}[g] = \frac{1}{|X'_\ell|} \sum_{(C_\ell,b)\in X'_\ell} [Z_\ell(\sigma(C_\ell)) \cdot b \cdot g(C_\ell, b)] \\ &= -\frac{1}{|X'_\ell|} \sum_{A\subseteq[n],|A|=\ell} \left(\sum_{(C_\ell,b)\in X'_\ell,\ V(C_\ell)=A} g(C_\ell, b) \cdot b \right) \cdot \chi_A(\sigma)\end{aligned}$$

□

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

Lemma 14. *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 $\|g\|_2 = \sqrt{\mathbb{E}_{U'_\ell}[g(C_\ell, b)^2]}$.

Proof. By Lemma 4 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 13 we get that

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

□

We proceed to bound the discrimination norm as before.

Lemma 15. *Let P be a predicate of complexity r, let $\mathcal{D}' \subseteq \{P_\sigma\}_{\sigma\in\{\pm1\}^n}$ be a set of distributions over variable k-tuples 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 12 and the definition of r,

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

Hence, by Lemma 14 we get that,

$$\mathop{\mathbb{E}}_{\sigma\sim\mathcal{S}}[|\Delta(\sigma,h)|] \le \sum_{S\subseteq[k],\ |S|\ge r} \left|\hat{P}(S)\right| \cdot \mathop{\mathbb{E}}_{\sigma\sim\mathcal{S}}[|\Gamma_\ell(\sigma,h_S)|] = O_k\left(\sum_{S\subseteq[k],\ |S|\ge r}\frac{(\ln d)^{\ell/2}\cdot\|h_S\|_2}{\sqrt{|X'_\ell|}}\right) \tag{16}$$

By the definition of h_S,

$$\begin{aligned}
\|h_S\|_2^2 &= \mathop{\mathbb{E}}_{U'_\ell}[h_S(C_\ell,b)^2] \\
&= \frac{|X'_\ell|^2}{|X'_k|^2}\cdot\mathop{\mathbb{E}}_{U'_\ell}\left[\left(\sum_{C\in Y_k,\ C_{|S}=C_\ell} h(C,b)\right)^2\right] \\
&\le \frac{|X'_\ell|^2}{|X'_k|^2}\cdot\mathop{\mathbb{E}}_{U'_\ell}\left[\frac{|X'_k|}{|X'_\ell|}\cdot\left(\sum_{C\in Y_k,\ C_{|S}=C_\ell} h(C,b)^2\right)\right] \\
&= \mathop{\mathbb{E}}_{U'_k}[h(C,b)^2] = \|h\|_2^2 = 1,
\end{aligned}$$

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

$$\left|\{C\in Y_k \mid C_{|S}=C_\ell\}\right| = \frac{|Y_k|}{|Y_\ell|} = \frac{|X'_k|}{|X'_\ell|}.$$

By plugging this into eq.(16) and using the fact that $\ln d < n$ we get,

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

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 17) Our reference distribution is the uniform distribution U'_k and the set of distributions $\mathcal{D}=\mathcal{D}_P=\{P_\sigma\}_{\sigma\in\{\pm1\}^n}$ 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 $\mathcal{S}=\{\sigma\mid P_\sigma\in\mathcal{D}'\}$. Then, by Lemma 15, 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. □