2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS) | 979-8-3503-1894-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/FOCS57990.2023.00026

Efficient Algorithms for Semirandom Planted CSPs at the Refutation Threshold

Venkatesan Guruswami
Department of EECS
UC Berkeley
Berkeley, USA
venkatg@berkeley.edu

Jun-Ting Hsieh
Computer Science Department
Carnegie Mellon University
Pittsburgh, USA
juntingh@cs.cmu.edu

Pravesh K. Kothari
Computer Science Department
Carnegie Mellon University
Pittsburgh, USA
praveshk@cs.cmu.edu

Peter Manohar
Computer Science Department
Carnegie Mellon University
Pittsburgh, USA
pmanohar@cs.cmu.edu

***Abstract*—We present an efficient algorithm to solve semirandom planted instances of any Boolean constraint satisfaction problem (CSP). The semirandom model is a hybrid between worst case and average case input models, where the input is generated by (1) choosing an arbitrary planted assignment x^*, (2) choosing an arbitrary clause structure, and (3) choosing literal negations for each clause from an arbitrary distribution "shifted by x^*" so that x^* satisfies each constraint. For an n variable semirandom planted instance of a k-arity CSP, our algorithm runs in polynomial time and outputs an assignment that satisfies all but a $o(1)$-fraction of constraints, provided that the instance has at least $\tilde{O}(n^{k/2})$ constraints. This matches, up to $\mathrm{polylog}(n)$ factors, the clause threshold for algorithms that solve *fully random* planted CSPs [23], as well as algorithms that refute *random and semirandom* CSPs [1], [4]. Our result shows that despite having worst case clause structure, the randomness in the literal patterns makes semirandom planted CSPs significantly easier than worst case, where analogous results require $O(n^k)$ constraints [7], [26].**

Perhaps surprisingly, our algorithm follows a significantly different conceptual framework when compared to the recent resolution of semirandom CSP refutation. This turns out to be inherent and, at a technical level, can be attributed to the need for *relative* spectral approximation of certain random matrices — reminiscent of the classical spectral sparsification — which ensures that an SDP can certify the *uniqueness* of the planted assignment. In contrast, in the refutation setting, it suffices to obtain a weaker guarantee of absolute upper bounds on the spectral norm of related matrices.

***Index Terms*—Semirandom CSPs, Expander Decomposition, Spectral Sparsification**

I. Introduction

Four decades of work in computational complexity has uncovered strong hardness results for constraint satisfaction problems (CSPs) such as k-SAT that leave only a little room for non-trivial efficient algorithms in the *worst-case*. Strong hardness of approximation [30] essentially rule out (unless $\mathsf{P} = \mathsf{NP}$) any improvement over simply returning a uniformly random assignment when the input instance is *sparse* (i.e., has $m = O(n)$ constraints on n variables). While there is a polynomial time approximation scheme (PTAS) [7] for maximally dense instances (e.g., with $m = O(n^k)$ constraints for k-SAT), under the exponential time hypothesis [32], we can already rule out polynomial time algorithms for $o(n^k)$ dense instances and more generally, $2^{n^{1-\delta}}$ time algorithms for any $\delta > 0$ for $o(n^{k-1})$ dense instances [26].

Search and refutation in the average-case. In sharp contrast, in well-studied *average-case* settings, there appears to be significant space for new algorithms and markedly better guarantees for CSPs. CSPs can be studied as two natural problems in such average-case settings: the problem of *refutation* — where we focus on efficiently finding witnesses of unsatisfiability for models largely supported on unsatisfiable instances, and the problem of *search* — where our goal is to find an assignment that the model guarantees is *planted* in the instance.

The refutation problem has been heavily investigated in the past two decades. For *fully random* k-CSPs with uniformly random clause structure (i.e., which variables appear in each clause) and "literal pattern" (i.e., which variables appear negated in each clause), there is a polynomial-time algorithm that, with high probability over the instance, certifies that the instance is unsatisfiable, provided that m is at least $\tilde{O}(n^{k/2})$ [4], [9], [18], [27], [44]. This threshold is far below the $\sim n^k$ hardness threshold of [26]. Furthermore, there is lower bounds in various restricted models [8], [13], [20], [24], [36], [39], [43] provide some evidence that this threshold might be tight for polynomial time algorithms.

The search problem for planted models of CSPs has also received a fair bit of attention. The setting naturally arises in the investigation of *local* one-way functions and pseudorandom generators in cryptography. Indeed, the security of the well-known one-way function proposed by Goldreich [28] (also conjectured to be a pseudorandom generator [6], [42]) is equivalent to the hardness of recovering a satisfying assignment planted (via a carefully chosen procedure) in a random CSP instance with an appropriate predicate. This has led to significant research on solving *fully random* planted CSPs [11], [15], [17], [23], [33]. Specifically, Feldman, Perkins and Vempala [23] showed that for *fully random* planted k-CSPs with planted assignment x^*, there is a polynomial-time algorithm that, with high probability over the instance, recovers the planted assignment x^* *exactly*, provided that the instance has at least $\tilde{O}(n^{k/2})$ constraints. That is, the refutation and search versions have the same clause threshold.

***Beyond* the average-case: semirandom instances** The phenomenal progress in average-case algorithm design notwith-

979-8-3503-1894-4/23/$31.00 ©2023 IEEE

DOI 10.1109/FOCS57990.2023.00026

standing, there is a nagging concern that the algorithms so developed rely too heavily on "brittle" properties of a specific random model. That is, our methods may have "overfitted" to the specific setting thus offering algorithms that only apply in a limited setting. Unfortunately, this fear turns out to be rather well-founded — natural spectral algorithms for refuting random k-CSPs and solving the natural planted variants break down under minor perturbations such as the introduction of a vanishingly small fraction of additional clauses.

Motivated by such concerns, Blum and Spencer [14] and later Feige and Kilian [21], [22] introduced *semirandom* models for optimization problems. In semirandom models, the instances are constructed by a combination of benign average-case and adversarial worst-case choices. Algorithms that succeed for such models are naturally "robust" to perturbations of the input instance.

For CSPs, a *semirandom* instance is generated by first choosing a "worst-case" clause structure and then choosing the literal negation patterns in each clause via some sufficiently random (and thus "benign") process. Recent work [1], [29], [31] has shown that in the case of refutation, there are indeed more resilient algorithms that succeed in refuting semirandom instances at the *same* $\tilde{O}(n^{k/2})$ threshold as the *fully random* case. These developments have added new general-purpose new spectral methods based on Kikuchi matrices [29], [48] to our algorithmic arsenal.

Semirandom planted problems. In this work, we make the first step in obtaining algorithms for the *search* variant of CSPs in the semirandom setting. Our main result gives an efficient algorithm for solving semirandom planted CSPs that succeeds in finding the planted assignment whenever the number of constraints exceeds $\tilde{O}(n^{k/2})$ — the *same* threshold at which polynomial time algorithms exist for the refutation problem for random (and semirandom) instances.

Theorem 1 (Main result, informal Theorem 2)**.** *There is an efficient algorithm that takes as input a k-CSP Ψ and outputs an assignment x with the following guarantee: if Ψ is a semirandom planted k-CSP with $m \geq \tilde{O}(n^{k/2})$ constraints, then with high probability over Ψ, the output x satisfies $1 - o(1)$-fraction of the constraints in Ψ.*

We note that in the semirandom setting, it is not possible to efficiently recover an assignment that satisfies *all* of the constraints without being able to do so even when $m = O(n)$,[1]. This is because it is easy to construct a semirandom instance ψ that is the "union" of two disjoint instances ψ_1 and ψ_2, where ψ_1 and ψ_2 use disjoint sets of $n/2$ variables, but ψ_1 only has $m_1 \sim O(n)$ clauses (and ψ_2, therefore, contains almost all of the $m \sim n^{k/2}$ clauses). Thus, the guarantee in Theorem 1 of satisfying a $1-o(1)$-fraction of constraints is qualitatively the best we can hope for.

Search vs. refutation. It is natural to compare Theorem 1 to the recent resolution of the problem of *refuting* semirandom CSPs [1], [29], [31]. For average-case optimization problems, techniques for refuting random instances can typically be adapted to solving the search problem in the related planted model. This can be formalized in the *proofs to algorithms* paradigm [10], [25] where spectral/SDP-based refutations can be transformed into "simple" (i.e., "captured" within the low-degree sum-of-squares proof system) efficient certificates of near-uniqueness of optimal solution — that is, every optimal solution is close to the planted assignment. Unfortunately, this intuition breaks down even in the simplest setting of semirandom 2-XOR where there can be multiple maximally far-off solutions that satisfy as many (or even more) constraints as the planted assignment. Such departure from uniqueness also breaks algorithms for recovery [23] that rely on the top eigenvector of a certain matrix built from the instance being correlated with the planted assignment. In the semirandom setting, one can build instances where the top eigenspace of such matrices is the span of the multiple optimal solutions and has dimension $\omega(1)$ (searching for a Boolean vector close to the subspace is, in general, hard in super-constant dimensional subspaces).

Our key insight. Our starting point is a new, efficiently checkable certificate of the unique identifiability of the planted solution for noisy planted k-XOR (i.e., where each equation in a satisfiable k-sparse linear system is corrupted independently with some fixed constant probability) whenever the constraint hypergraph satisfies a certain weak expansion property. For random graphs in case of 2-XOR (and generalizations to multiple community *stochastic block models*), such certificates (in the form of explicit dual solutions to a semidefinite program) were shown to exist by Abbe and Sandon [2].

Our certificate naturally yields an efficient algorithm for *exactly* recovering the planted assignment in noisy k-XOR instances whenever the constraint hypergraph satisfies a deterministic weak expansion property and has size exceeding the refutation threshold $\sim n^{k/2}$. Finally, we use expander decomposition procedures to decompose the input constraint hypergraph into pieces that satisfy the above condition. This is done in a manner that further allows us to find a good assignment via a consistent patching scheme to combine solutions across all the pieces in our decomposition.

A. Our semirandom planted model and results

Before formally stating our results, we define the semirandom planted model that we work with and explain some of the subtleties in the definition. Our model is the natural one that arises if we wish to enforce independent randomness (for each clause) in the literal negations, while still fixing a particular satisfying assignment.

Definition I.1 (k-ary Boolean CSPs)**.** A CSP instance Ψ with a k-ary predicate $P\colon \{-1,1\}^k \to \{0,1\}$ is a set of m constraints on variables $x_1, \ldots, x_n$ of the form $P(\ell(\vec{C})_1 x_{\vec{C}_1}, \ell(\vec{C})_2 x_{\vec{C}_2}, \ldots, \ell(\vec{C})_k x_{\vec{C}_k}) = 1$. Here, $\vec{C}$ ranges

[1] Achieving this would break a hardness assumption for the search problem analogous to Feige's random 3-SAT hypothesis for the refutation problem [20].

over a collection $\vec{\mathcal{H}}$ of *scopes*[2] (a.k.a. clause structure) of k-tuples of n variables and $\ell(\vec{C}) \in \{-1,1\}^k$ are "literal negations", one for each $\vec{C}$ in $\vec{\mathcal{H}}$. We let $\mathrm{val}_\Psi(x)$ denote the fraction of constraints satisfied by an assignment $x \in \{-1,1\}^n$, and we define the *value* of Ψ, $\mathrm{val}(\Psi)$, to be $\max_{x\in\{-1,1\}^n} \mathrm{val}_\Psi(x)$.

Definition I.2 (Semirandom planted k-ary Boolean CSPs)**.** Let $P\colon \{-1,1\}^k \to \{0,1\}$ be a predicate. We say that a distribution Q over $\{-1,1\}^k$ is a *planting distribution for* P if $\Pr_{y\leftarrow Q}[P(y)=1]=1$.

We say that an instance Ψ with predicate P is a *semirandom planted instance* with *planting distribution* Q if it is sampled from a distribution $\Psi(\vec{\mathcal{H}}, x^*, Q)$ where

(1) the scopes $\vec{\mathcal{H}} \subseteq [n]^k$ and planted assignment $x^* \in \{-1,1\}^n$ are arbitrary, and

(2) $\Psi(\vec{\mathcal{H}}, x^*, Q)$ is defined as follows: for each $\vec{C} \in \vec{\mathcal{H}}$, sample literal negations $\ell(\vec{C}) \leftarrow Q(\ell(\vec{C}) \odot x^*_{\vec{C}})$, where "$\odot$" denotes the element-wise product of two vectors. That is, $\Pr[\ell(\vec{C}) = \ell] = Q(\ell \odot x^*_{\vec{C}})$ for each $\ell \in \{-1,1\}^k$. Then, add the constraint $P(\ell(\vec{C})_1 x_{\vec{C}_1}, \ell(\vec{C})_2 x_{\vec{C}_2}, \ldots, \ell(\vec{C})_k x_{\vec{C}_k}) = 1$ to Ψ.

Notice that because Q is supported only on satisfying assignments to P, it follows that if $\Psi \leftarrow \Psi(\vec{\mathcal{H}}, x^*, Q)$, then x^* satisfies Ψ with probability 1.

A (fully) random planted CSP, e.g., as defined in [23], is generated by first sampling $\vec{\mathcal{H}} \leftarrow [n]^k$ uniformly at random, and then sampling $\Psi \leftarrow \Psi(\vec{\mathcal{H}}, x^*, Q)$. The difference in the semirandom planted model is that we allow $\vec{\mathcal{H}}$ to be *worst case*.

Notice that in Definition I.2, there are some choices of Q for which the planted instance becomes easy to solve. In the case of, e.g., 3-SAT, one could set the planting distribution Q to be uniform over all 7 satisfying assignments, which results in the literal negations in each clause being chosen uniformly conditioned on x^* satisfying the clause. However, by simply counting how many times the variable x_i appears negated versus not negated and taking the majority vote, we recover x^* with high probability [11], [33] (see Section C).

Instead of sampling clauses uniformly from all those satisfied by x^*, one can create more challenging distributions, e.g., ones where true and false literals appear in equal proportion. Such distributions are termed "quiet plantings" and have been studied extensively [17], [33], [37], [38]. Our semirandom model follows definitions in [23], [24] and is a general planted model with respect to a *planting distribution* Q, which unifies various plantings studied in the past.

Unlike in the case of random planted CSPs, we cannot hope to recover the planted assignment x^* exactly in the semirandom setting. Indeed, the scopes $\vec{\mathcal{H}}$ may not use some variable x_i at all, and so we cannot hope to recover x^*_i! Thus, our goal is instead to recover an assignment x that has nontrivially large value, ideally value $1-\varepsilon$ for arbitrarily small ε. Our main result, stated formally below, gives an algorithm to accomplish this task.

Theorem 2 (Formal Theorem 1)**.** *Let $k \in \mathbb{N}$ be constant. There is a polynomial-time algorithm that takes as input a k-CSP Ψ and outputs an assignment x with the following guarantee. If Ψ is a semirandom planted k-CSP with $m \geq c^k n^{k/2} \cdot \frac{\log^3 n}{\varepsilon^9}$ constraints drawn from $\Psi(\vec{\mathcal{H}}, x^*, Q)$, then with probability $1 - 1/\mathrm{poly}(n)$ over Ψ, the output x of the algorithm has $\mathrm{val}_\Psi(x) \geq 1-\varepsilon$. Here, c is a universal constant.*

In particular, setting $\varepsilon = 1/\mathrm{polylog}(n)$, if $m \geq \tilde{O}(n^{k/2})$, then with high probability over $\Psi \leftarrow \Psi(\vec{\mathcal{H}}, x^*, Q)$, the algorithm outputs x with $\mathrm{val}_\Psi(x) \geq 1 - o(1)$.

Theorem 2 shows that one can *nearly* solve a semirandom planted k-CSP at the same $\tilde{O}(n^{k/2})$ threshold as done in the random case [23], matching the same $\tilde{O}(n^{k/2})$ threshold as for semirandom refutation [1], [29], [31]. However, as explained earlier (and will be discussed further in Section II), there are several unanticipated technical hurdles to overcome in the semirandom planted setting that are not present in the semirandom refutation setting, and this causes many of the natural approaches that "springboard off" the refutation case to fail. Curiously enough, for the special case of $k=2$ there *is* a simple reduction from search to refutation for the case of 2-XOR, which we will describe in Section II-A, but the same approach for k-XOR encounters a hardness barrier for $k \geq 3$, as we will discuss in Section II-B.

Theorem 2 also breaks Goldreich's candidate pseudorandom generators [28] and its variants [6],[3] when they have $\tilde{\Omega}(n^{k/2})$ stretch and *any* k-hypergraph (not just a random one). In fact, not only does Theorem 2 break the PRG, it also gives an algorithm that nearly *inverts* it.

Noisy planted *k*-XOR. Similar to work on random planted CSPs [23] and the refutation setting [1], [4], [29], [31], [44], our proof of Theorem 2 goes through a reduction to noisy k-XOR. Our algorithm achieves very strong guarantees in the noisy k-XOR case, as we now explain. We define the noisy k-XOR model below and then state our result.

Definition I.3 (Noisy planted k-XOR)**.** Let $\mathcal{H} \subseteq \binom{[n]}{k}$ be a k-uniform hypergraph on n vertices, let $x^* \in \{-1,1\}^n$, and let $\eta \in [0,1/2)$. Let $\psi(\mathcal{H}, x^*, \eta)$ denote the distribution on k-XOR instances over n variables $x_1,\ldots,x_n \in \{-1,1\}$ obtained by, for each $C \in \mathcal{H}$, adding the constraint $\prod_{i\in C} x_i = \prod_{i\in C} x^*_i$ with probability $1-\eta$, and otherwise adding the constraint $\prod_{i\in C} x_i = -\prod_{i\in C} x^*_i$. In the latter case, we say that the constraint C is *corrupted* or *noisy*.

We call ψ a *noisy planted* k-XOR instance if it is sampled from $\psi(\mathcal{H}, x^*, \eta)$, for some $\mathcal{H}$, x^*, and η; the hypergraph $\mathcal{H}$ is the constraint hypergraph, x^* is the planted assignment, and η is the noise parameter. Furthermore, we let $\mathcal{E}_\psi \subseteq \mathcal{H}$ denote the (unknown) set of corrupted constraints.

[2] We additionally allow $\vec{\mathcal{H}}$ to be a multiset, i.e., that multiple clauses can contain the same ordered set of variables.

[3] Goldreich's original PRG is essentially a planted k-CSP with a Boolean predicate P on a random hypergraph, containing both P and $\neg P$ constraints.

Theorem 3 (Algorithm for noisy k-XOR). *Let $\eta \in [0, 1/2)$, let $k, n \in \mathbb{N}$, and let $\varepsilon \in (0, 1)$. Let $m \geq cn^{k/2} \cdot \frac{k^4 \log^3 n}{\varepsilon^5 (1-2\eta)^4}$ for a universal constant c. There is a polynomial-time algorithm $\mathcal{A}$ that takes as input a k-XOR instance ψ with constraint hypergraph $\mathcal{H}$ and outputs two disjoint sets $\mathcal{A}_1(\mathcal{H}), \mathcal{A}_2(\psi) \subseteq \mathcal{H}$ with the following guarantees: (1) for any instance ψ with m constraints, $|\mathcal{A}_1(\mathcal{H})| \leq \varepsilon m$ and $\mathcal{A}_1(\mathcal{H})$ only depends on $\mathcal{H}$, and (2) for any $x^* \in \{-1, 1\}^n$ and any k-uniform hypergraph $\mathcal{H}$ with at least m hyperedges, with probability at least $1 - 1/\mathrm{poly}(n)$ over $\psi \leftarrow \psi(\mathcal{H}, x^*, \eta)$, it holds that $\mathcal{A}_2(\psi) = \mathcal{E}_\psi \cap (\mathcal{H} \setminus \mathcal{A}_1(\mathcal{H}))$.*

In words, the algorithm discards a small number of constraints, and among the constraints that are not discarded, correctly identifies all (and only) the corrupted constraints. In particular, the subinstance obtained by discarding the $\lesssim (\varepsilon + \eta)m$ constraints $\mathcal{A}_1(\mathcal{H}) \cup \mathcal{A}_2(\psi)$ is satisfiable (and a solution can be found by Gaussian elimination). Thus, Theorem 3 immediately implies that for k-XOR, the NP-hard task of deciding if ψ has value $\geq 1 - \eta$ or $\leq \frac{1}{2} + \eta$ is actually *easy* if ψ has $\sim n^{k/2}$ constraints (far below the $\sim n^k$-hardness of [26]), provided that the η-fraction of corrupted constraints in the "yes" case are a *randomly chosen subset* of the otherwise arbitrary constraints.

Exact vs. approximate recovery. As alluded to above, the guarantees of Theorem 3 are much stronger: not only can we find a good assignment to ψ, we can break the constraints into two parts, a small fraction, $\mathcal{A}_1(\mathcal{H})$, where we are unable to determine the corrupted constraints,[4] and a large fraction, $\mathcal{H} \setminus \mathcal{A}_1(\mathcal{H})$, where we can determine *exactly* all of the corrupted constraints, $\mathcal{A}_2(\psi)$. Moreover, this partition depends only on the hypergraph $\mathcal{H}$ and is *independent of the noise*. We remark that it is not immediately obvious that this guarantee is achievable even for exponential-time algorithms, as x^* may not be the globally optimal assignment with constant probability. This strong guarantee of Theorem 3 is in fact required for the reduction from Theorem 2 to Theorem 3; the weaker (and more intuitive) guarantee of approximate recovery — obtaining an assignment of value $1 - \eta - o(1)$ for the noisy XOR instance — is insufficient for the reduction.

One can view Theorem 3 as an algorithm that extracts almost all the information about the planted assignment x^* encoded by the instance ψ. Indeed, notice that even if $\eta = 0$, the instance ψ only determines x^* "up to a linear subspace."[5] Namely, if we let $y \in \{-1, 1\}^n$ be any solution to the system of constraints $\prod_{i \in C} y_i = 1$ for $C \in \mathcal{H}$, then $y \odot x^*$ is also a planted assignment for ψ: formally, $\psi(\mathcal{H}, x^*, \eta) = \psi(\mathcal{H}, y \odot x^*, \eta)$ as distributions. So, aside from the εm constraints that are discarded, with high probability over ψ the algorithm determines the uncorrupted right-hand sides $\prod_{i \in C} x_i^*$ for every remaining constraint, which is all the information about the planted assignment x^* encoded in the remaining constraints.

The importance of relative spectral approximation. As a key technical ingredient in the algorithm, we uncover a *deterministic* condition — relative spectral approximation of the Laplacian of a graph (associated with the input instance) by a certain correlated random sample from it — which when satisfied implies uniqueness of the SDP solution (Lemma II.5). In Lemma II.6 and Lemma VI.7, we establish such spectral approximation guarantees.

This spectral approximation property is the key ingredient in our certificate of unique identifiability of the planted assignment in a noisy k-XOR instance (see Section II-D for details) and allows us to *exactly* recover the planted assignment for 2-XOR instances where the constraint graph G is a weak spectral expander (i.e., spectral gap $\gg 1/\mathrm{poly}\log n$) (Lemma II.5), and forms the backbone of our final algorithm. We note that our spectral approximation condition can be seen as an analog of (and is, in fact, stronger than) the related spectral norm upper bound property that underlie the refutation algorithm of [1].

This process of extracting a "deterministic property of random instances sufficient for the analysis" is an important conceptual theme underlying recent progress on semirandom optimization, and manifests as, e.g., the notion of "butterfly degree" in [1], "hypergraph regularity" or spreadness in [29] in the context of semirandom CSP refutation, and biclique number bounds in the context of planted clique [16].

II. Technical Overview

In this section, we give an overview of the proof of Theorem 3 and our algorithm for noisy planted k-XOR. We defer discussion of the reduction from general k-CSPs to k-XOR used to obtain Theorem 2 to Section IV. There, we explain the additional challenges encountered in the semirandom case as compared to the random case [23, Section 4]. Somewhat surprisingly, the reduction is complicated and quite different from the random planted case or even the semirandom refutation setting, where the reduction to XOR is straightforward.

We now explain Theorem 3. As is typical in algorithm design for k-XOR, the case when k is even is considerably simpler than when k is odd. For the purpose of this overview, we will focus mostly on the even case, and only briefly discuss the additional techniques for odd k in Section II-E.

Notation. Throughout this paper, given a k-XOR instance ψ on hypergraph $\mathcal{H} \subseteq \binom{[n]}{k}$ with $m = |\mathcal{H}|$ and right-hand sides $\{b_C\}_{C \in \mathcal{H}}$, we define $\psi(x) := \sum_{C \in \mathcal{H}} b_C \prod_{i \in C} x_i$ to be a degree-k polynomial mapping $\{-1, 1\}^n \to [-m, m]$. We note that $\mathrm{val}_\psi(x) = \frac{1}{2} + \frac{1}{2m}\psi(x) \in [0, 1]$ is the fraction of constraints in ψ satisfied by x. Moreover, we will write $x_C := \prod_{i \in C} x_i$.

Unless otherwise stated, we will use ϕ to denote a 2-XOR instance and ψ to denote a k-XOR instance for any $k \geq 2$.

We note that for even arity k-XOR, we have $\mathrm{val}_\psi(x) = \mathrm{val}_\psi(-x)$, and so it is only possible for the optimal solution

[4] Note that discarding a small fraction of constraints is necessary in the semirandom setting, as ψ may contain many disconnected constant-size subinstances where it is not possible, even information-theoretically, to exactly identify the corrupted constraints with $1 - o(1)$ probability.

[5] A k-XOR constraint $x_{C_1} \cdots x_{C_k} = b_C \in \{-1, 1\}$ can be equivalently written as a linear equation $x'_{C_1} + \cdots + x'_{C_k} = b'_C$ over $\mathbb{F}_2$, where we map $+1$ to 0 and -1 to 1.

to be unique *up to a global sign*. We will abuse terminology and say that x^* is the unique optimal assignment if $\pm x^*$ are the only optimal assignments, and we will say that we have recovered x^* exactly if we obtain one of $\pm x^*$.

A. Approximate recovery for 2-XOR from refutation

First, let us focus on the case of $k = 2$, the simplest case, and let us furthermore suppose that we only want to achieve the weaker goal of recovering an assignment of value $1 - \eta - o(1)$. (Note that we do need the stronger guarantee of Theorem 3 to solve general planted CSPs in Theorem 2.)

For 2-XOR, this goal is actually quite straightforward to achieve using 2-XOR refutation as a blackbox. Let us represent the 2-XOR instance ϕ as a graph G on n vertices, along with right-hand sides b_{ij} for each edge $(i, j) \in E$. Recall that we have $b_{ij} = x^*_i x^*_j$ with probability $1 - \eta$, and $b_{ij} = -x^*_i x^*_j$ otherwise. Note that by concentration, $\mathrm{val}_\phi(x^*) = 1-\eta \pm o(1)$ with high probability.

We now make the following observation. Let us suppose that we sample the noise in two steps: first, we add each $(i, j) \in E$ to a set E' with probability 2η independently; then for each $(i, j) \in E'$ we set b_{ij} to be uniformly random from $\{-1, 1\}$. Using known results for semirandom 2-XOR refutation, it is possible to certify, via an SDP relaxation, that no assignment x can satisfy (or violate) more than $\frac{1}{2} + o(1)$ fraction of the constraints in E'.

Thus, we can simply solve the SDP relaxation for ϕ and obtain a degree-2 pseudo-expectation $\tilde{\mathbb{E}}$ in the variables $x_1, \dots, x_n$ over $\{-1, 1\}^n$ that maximizes $\phi(x)$. Let $\phi_{E'}$ be the subinstance containing only the constraints in E', and let $\phi_{E \setminus E'}$ be the subinstance containing only the constraints in $E \setminus E'$, which are uncorrupted. We have $\tilde{\mathbb{E}}[\mathrm{val}_\phi(x)] \geq 1-\eta-o(1)$, and the guarantee of refutation implies that $\tilde{\mathbb{E}}[\mathrm{val}_{\phi_{E'}}(x)] \leq \frac{1}{2}+o(1)$. As $\mathrm{val}_\phi(x) = (1-2\eta)\cdot \mathrm{val}_{\phi_{E\setminus E'}}(x) + 2\eta \cdot \mathrm{val}_{\phi_{E'}}(x)$, we therefore have that $\tilde{\mathbb{E}}[\mathrm{val}_{\phi_{E\setminus E'}}(x)] \geq 1 - o(1)$, i.e., $\tilde{\mathbb{E}}$ satisfies $1 - o(1)$ fraction of the constraints in $E \setminus E'$. Then, applying the standard Gaussian rounding, we obtain an x that satisfies $1 - \sqrt{o(1)}$ fraction of the constraints in $E \setminus E'$ and thus has value $\mathrm{val}_\phi(x) \geq 1 - \eta - o(1)$ (as any x must satisfy at least $\frac{1}{2} - o(1)$ fraction of the constraints in E', with high probability over the noise).

One interesting observation is that in the above discussion, we can additionally allow E' to be an *arbitrary* subset of E of size $2\eta m$. Indeed, this is because the rounding only "remembers" that $\tilde{\mathbb{E}}[\mathrm{val}_{\phi_{E\setminus E'}}(x)]$ has value $1 - o(1)$. As we shall see shortly, this is the key reason that the reduction breaks down for k-XOR.

B. The challenges for k-XOR and our strategy

Unfortunately, the natural blackbox reduction to refutation given in Section II-A does not generalize to k-XOR for $k \geq 3$. Following the approach described in the previous section, given a k-XOR instance ψ, one can solve a sum-of-squares SDP and obtain a pseudo-expectation $\tilde{\mathbb{E}}$ where $\tilde{\mathbb{E}}[\mathrm{val}_\psi(x)] \geq 1 - \eta - \delta$ and $\tilde{\mathbb{E}}[\mathrm{val}_{\psi_{E\setminus E'}}(x)] \geq 1 - \delta$ as before, where $\delta \sim 1/\mathrm{polylog}(n)$ when $m \gtrsim n^{k/2}$, due to the guarantees of refutation algorithms [1]. However, unlike 2-XOR where we have Gaussian rounding, for k-XOR there is no known rounding algorithm that takes a pseudo-expectation $\tilde{\mathbb{E}}$ with $\tilde{\mathbb{E}}[\mathrm{val}_{\psi_{E\setminus E'}}(x)] \geq 1 - \delta$ and outputs an assignment x such that $\mathrm{val}_{\psi_{E\setminus E'}}(x) \geq 1 - f(\delta)$, for some $f(\cdot)$ such that $f(\delta) \to 0$ as $\delta \to 0$. In fact, if we only "remember" that $\psi_{E\setminus E'}$ has value $1 - \delta$, then it is NP-hard to find an x with value $> 1/2 + \delta$ even when $\delta = n^{-c}$ for some constant $c > 0$, assuming a variant of the Sliding Scale Conjecture [12][6] (see e.g. [40], [41] for more details).

As we have seen, while semirandom k-XOR refutation allows us to efficiently approximate and certify the *value* of the planted instance, the challenge lies in the *rounding* of the SDP, where the goal is to recover an assignment x. This is a technical challenge that does not arise in the context of CSP refutation, as there we are merely trying to bound the value of the instance. As a result, new ideas are required to address this challenge.

Reduction from k-XOR to 2-XOR for even k. One could still consider the following natural approach. For simplicity, let $k = 4$. Given a 4-XOR instance ψ, we can write down a natural and related 2-XOR instance ϕ, as follows.

Definition II.1 (Reduction to 2-XOR)**.** Let ψ be a 4-XOR instance, and let ϕ be the 2-XOR defined as follows. The variables of ϕ are $y_{\{i,j\}}$ and correspond to *pairs* of variables $\{x_i, x_j\}$, and for each constraint $x_i x_j x_{i'} x_{j'} = b_{i,j,i',j'}$ in ψ, we split $\{i, j, i', j'\}$ into $\{i, j\}$ and $\{i', j'\}$ arbitrarily and add a constraint $y_{\{i,j\}} y_{\{i',j'\}} = b_{i,j,i',j'}$ to ϕ. See Fig. 1 for an example. This reduction easily generalizes to k-XOR for any even k.

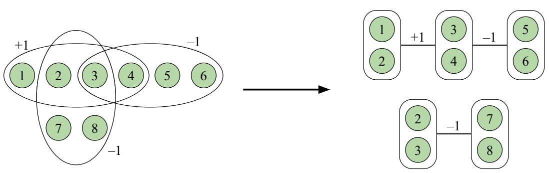

Fig. 1: An example of the 2-XOR instance ϕ from a 4-XOR instance ψ.

By following the approach for 2-XOR described in Section II-A, we can recover an assignment y that satisfies $1 - \eta - o(1)$ fraction of the constraints in ϕ. However, we need to recover an assignment x to the original k-XOR ψ, and it is quite possible that while y is a good assignment to ϕ, *it is not* close to $x^{\otimes 2}$ for *any* $x \in \{-1, 1\}^n$. If this happens, we will be unable to recover a good assignment to the 4-XOR instance ψ.

The key reason that this simple idea fails is because, unlike for random noisy XOR, the assignment y recovered is *not* necessarily unique, and we cannot hope for it to be in the

[6] Note that we do need the Sliding Scale Conjecture, as the hardness shown in [41] is not strong enough; it only proves hardness for $\delta \geq (\log\log n)^{-c}$, whereas we have $\delta \sim 1/\mathrm{polylog}(n)$.

semirandom setting! For random noisy XOR, one can argue that with high probability, y will be equal to $x^{*\otimes 2}$, and then we can immediately decode and recover x^* up to a global sign, i.e., we recover $\pm x^*$. But for semirandom instances, the situation can be far more complex.

Approximate 2-XOR recovery does not suffice for 4-XOR. When constructing the 2-XOR instance ϕ from the 4-XOR ψ (Definition II.1), it may be the case that ϕ can be partitioned into multiple disconnected clusters (or have very few edges across different clusters), even when the hypergraph $\mathcal{H}$ of ψ is connected; see Fig. 1 for example. By the algorithm described in Section II-A, we can get an assignment y that satisfies $1 - \eta - o(1)$ fraction of the constraints within each cluster.

The main challenge is to combine the information gathered from each cluster to recover an assignment x for the original 4-XOR ψ. Unfortunately, we do not know of a way to obtain a good assignment x based solely on the guarantee that y satisfies $1-\eta-o(1)$ fraction of constraints in each cluster. The issue occurs because the same variable $i \in [n]$ can appear in different clusters, e.g., $y_{\{1,2\}}$ and $y_{\{2,3\}}$ lie in different clusters in Fig. 1, and the recovered assignments in each cluster may implicitly choose different values for x_i because of the noise. Indeed, even if the local optimum is consistent with x^*, there can still be multiple "good" assignments that achieve $1-\eta-o(1)$ value on the subinstance restricted to a cluster. So, unless the SDP can certify unique optimality of x^*, standard rounding techniques such as Gaussian rounding will merely output a "good" y, which may be inconsistent with x^* and thus can choose inconsistent values of x_i across the different clusters.

Exact 2-XOR recovery implies exact 4-XOR recovery. This leads to our main insight: if the subinstance of ϕ admits a *unique* local optimal assignment y^* (restricted to the cluster) that matches the planted assignment up to a sign, i.e., $y^*_{\{i,j\}} = \pm x^*_i x^*_j$, then for each edge in the cluster we know $y^*_{\{i,j\}} y^*_{\{i',j'\}} = x^*_i x^*_j x^*_{i'} x^*_{j'}$, and so the local constraints that are violated must be exactly the corrupted ones. Moreover, if the SDP can certify the uniqueness of the local optimal assignment for a cluster, then the SDP solution will be a *rank* 1 *matrix* $y^* y^{*\top}$, and so we can precisely identify which constraints in ϕ are corrupted. By repeating this for every cluster, we can identify all corrupted constraints in the original 4-XOR ψ (except for the small number of "cross cluster" edges), and thus achieve the guarantee stated in Theorem 3.

The general algorithmic strategy. The above discussion suggests that given a k-XOR instance ψ, we should first construct the 2-XOR ϕ, and then decompose the constraint graph G of ϕ into pieces in some particular way so that the induced local instances have unique solutions. Namely, the examples suggest the following algorithmic strategy.

Strategy II.2 (Algorithm Blueprint for even k). Given a noisy k-XOR instance ψ with planted assignment x^* and m constraints, we do the following:

(1) Construct the 2-XOR instance ϕ as in Definition II.1, which is a noisy 2-XOR on $n^{k/2}$ variables with planted assignment y^*. Moreover, there is a one-to-one mapping between constraints in ϕ and ψ.

(2) Let G be the constraint graph of ϕ. Decompose G into subgraphs $G_1, \dots, G_T$ while only discarding a $o(1)$-fraction of edges such that each subgraph G_i satisfies "some property". For each subgraph G_i, we define ϕ_i to be the subinstance of ϕ corresponding to the constraints in G_i. The goal is to identify a local property that the G_i's satisfy so that (1) we can perform the decomposition efficiently, and (2) for each subinstance ϕ_i, we can "recover y^* locally", i.e., we can find an assignment $y^{(i)}$ to the 2-XOR instance ϕ_i that is consistent with the planted assignment y^*.

(3) As each $y^{(i)}$ is consistent with y^*, the constraints in ϕ_i violated by $y^{(i)}$ must be precisely the corrupted constraints in ϕ_i. Hence, for the constraints that appear in one of the ϕ_i's, we have determined exactly which ones are corrupted.

(4) We have thus determined, for all but $o(m)$ constraints, precisely which ones are corrupted in the original k-XOR instance ψ. (Note that this is the *stronger* guarantee that we achieve in Theorem 3.) By discarding the corrupted constraints along with the $o(m)$ constraints where we "give up", we thus obtain a system of k-sparse linear equations with $m(1-\eta-o(1))$ equations that has at least one solution (namely x^*), and so by solving it we obtain an x with $\mathrm{val}_\psi(x) \geq 1-\eta-o(1)$.

C. Information-theoretic exact recovery from relative cut approximation

Following Strategy II.2, the first technical question to now ask is: given a noisy 2-XOR instance ϕ with n variables, $m \gg n$ constraints, and planted assignment x^*, what conditions do we need to impose on the constraint graph G so that we can recover x^* (up to a sign) exactly? As a natural first step, we investigate what conditions are required so that we can accomplish this *information-theoretically*.

Fact II.3. *Let $G = (V, E_G)$ be an n-vertex graph, and let $H = (V, E_H)$ be a subgraph of G where $E_H \subseteq E_G$. Let L_G, L_H be the* unnormalized *Laplacians of G and H. Consider a noisy planted 2-XOR instance ϕ on G with planted assignment $x^* \in \{-1,1\}^n$ (Definition I.3), and suppose E_H is the set of corrupted edges. Suppose that for every $x \in \{-1,1\}^n \setminus \{\vec{1}, -\vec{1}\}$, it holds that $x^\top L_H x < \frac{1}{2} x^\top L_G x$. Then, x^* and $-x^*$ are the only two optimal assignments to ϕ.*

Note that the condition $x^\top L_H x < \frac{1}{2} x^\top L_G x$ for $x \notin \{\vec{1}, -\vec{1}\}$ implies that G is connected, as otherwise L_G has a kernel of dimension ≥ 2, which would contradict this assumption.

Proof. Let $x \in \{-1,1\}^n$ be any assignment. We wish to show that $\phi(x)$ is uniquely maximized when $x = x^*, -x^*$.

We observe that

$$\begin{aligned}\phi(x) &= \sum_{(i,j)\in E_G} x_i x_j b_{ij} \\ &= \sum_{(i,j)\in E_G} x_i x_j x_i^* x_j^* - 2 \sum_{(i,j)\in E_H} x_i x_j x_i^* x_j^* \ .\end{aligned}$$

Hence, by replacing x with $x \odot x^*$, without loss of generality we can assume that $x^* = \vec{1}$. Now, let D_G, D_H and A_G, A_H be the degree and adjacency matrices of G and H, so that $L_G = D_G - A_G$ and $L_H = D_H - A_H$. We thus have that

$$\begin{aligned}2\phi(x) &= x^\top A_G x - 2x^\top A_H x \\ &= x^\top (D_G - 2D_H)x - x^\top (L_G - 2L_H)x \\ &= 2(|E_G| - 2|E_H|) - x^\top (L_G - 2L_H)x \ .\end{aligned}$$

By assumption, if $x \in \{-1,1\}^n$ and $x \neq \vec{1}, -\vec{1}$, then we have that $x^\top (L_G - 2L_H)x > 0$, which implies that $\phi(x) < \phi(\vec{1})$, and finishes the proof. □

Fact II.3 shows that if we can argue that $x^\top L_H x < \frac{1}{2} x^\top L_G x$ for every $x \in \{-1,1\}^n \setminus \{\vec{1}, -\vec{1}\}$, then at least information-theoretically we can uniquely determine x^*. Observe that if we view x as the signed indicator vector of a subset $S \subseteq [n]$, then $x^\top L_G x = E_G(S, \bar{S})$, the number of edges in G crossing the cut defined by S, and similarly for $x^\top L_H x$. So, one can view the condition in Fact II.3 as saying that the subgraph H needs to be a (one-sided) cut sparsifier of G, i.e., it needs to roughly preserve the size of all cuts in G. The following relative cut approximation result of Karger [35] shows that this will hold with high probability when H is a randomly chosen subset of G, provided that the minimum cut in G is not too small.

Lemma II.4 (Relative cut approximation [35]). *Let $\eta \in (0,1)$. Suppose an n-vertex graph G has min-cut $c_{\min} \geq \frac{12 \log n}{\eta}$, and suppose H is a subgraph of G by selecting each edge with probability η. Then, with probability $1 - o(1)$,*

$$(1-\delta) x^\top L_G x \leq \frac{1}{\eta} \cdot x^\top L_H x \leq (1+\delta) x^\top L_G x \ , \ \forall x \in \{-1,1\}^n$$

for $\delta = \sqrt{\frac{12 \log n}{\eta c_{\min}}}$.

With Lemma II.4 and Fact II.3 in hand, we now have at least an information-theoretic algorithm with the same guarantees as in Theorem 3. We follow the strategy highlighted in Strategy II.2. To decompose the graph G, we recursively find a min cut and split if it is below the threshold in Lemma II.4. Notice that this discards at most $O(n \log n) = o(m)$ constraints (for $m \gg n \log n$), and these are precisely the constraints that we "give up" on and do not determine which ones are corrupted. Then, with high probability the local optimal assignment is consistent with x^*, and so locally we have learned *exactly* which constraints are corrupted. Hence, we have produced two sets of constraints: E_1, the $o(1)$-fraction of edges discarded during the decomposition, and $E_2 = (G \setminus E_1) \cap \mathcal{E}_\phi$, which is exactly the set of corrupted constraints after discarding E_1. We note that it is a priori not obvious that this is achievable even for an *exponential-time* algorithm, as even though the 2^n-time brute force algorithm will find the best assignment x to ϕ, it may not necessarily be x^*, and so the set of constraints violated by the globally optimal assignment might not be $\mathcal{E}_\phi$.

D. Efficient exact recovery from relative spectral approximation

Information-theoretic uniqueness implies that the planted assignment x^* is the unique optimal assignment. But can we efficiently recover x^*? One natural approach is to simply solve the basic SDP relaxation of ϕ: for $X \in \mathbb{R}^{n \times n}$, maximize $\phi(X) := \sum_{(i,j)\in G} X_{ij} b_{ij}$ subject to $X \succeq 0$, $X = X^\top$, and $\operatorname{diag}(X) = \mathbb{I}$. If the optimal SDP solution is simply $X = x^* {x^*}^\top$, then we trivially recover x^* from the SDP solution. We thus ask: does the min cut condition of Fact II.3 and Lemma II.4 imply that $x^* {x^*}^\top$ is the unique optimal solution to the SDP? Namely, is the min cut condition sufficient for the SDP to certify that x^* is the unique optimal assignment?

Unfortunately, it turns out that this is not the case, and we give a counterexample in Section A. We thus require a stronger condition than the min cut one in order to obtain efficient algorithms. Nonetheless, an analogue of Fact II.3 continues to hold, although now we require a stronger version that holds for all SDP solutions X, not just $x \in \{-1,1\}^n$. This stronger statement shows the SDP can *certify* that x^* is the unique optimal assignment if and only if a certain relative spectral approximation guarantee holds for the corrupted edges.

Lemma II.5 (SDP-certified uniqueness from relative spectral approximation). *Let $G = (V, E_G)$ be an n-vertex connected graph, and let $H = (V, E_H)$ be a subgraph of G where $E_H \subseteq E_G$. Let L_G, L_H be the unnormalized Laplacians of G and H. Consider a noisy planted 2-XOR instance ϕ on G with planted assignment $x^* \in \{-1,1\}^n$ (Definition I.3), and suppose E_H is the set of corrupted edges.*

The SDP relaxation of ϕ satisfies

$$\max_{X \succeq 0, \ X = X^\top, \ \operatorname{diag}(X) = \mathbb{I}} \phi(X) = \phi(x^*) = |E_G| - 2|E_H| \,,$$

where $X = x^* {x^*}^\top$ is the unique *optimum if and only if G and H satisfy*

$$\langle X, L_H \rangle < \frac{1}{2} \langle X, L_G \rangle \,,$$
$$\forall X \succeq 0, \ X = X^\top, \ \operatorname{diag}(X) = \mathbb{I}, \ X \neq \vec{1}\vec{1}^\top \ .$$

Proof. Recall that each $e = \{i,j\} \in E$ corresponds to a constraint $x_i x_j = b_e$ where $b_e = x_i^* x_j^*$ if $e \in E_G \setminus E_H$ and $b_e = -x_i^* x_j^*$ if $e \in E_H$, meaning that $\phi(X) = \sum_{\{i,j\}\in G \setminus E} X_{ij} x_i^* x_j^* - \sum_{\{i,j\}\in E} X_{ij} x_i^* x_j^*$. Without loss of generality, we can assume that $x^* = \vec{1}$ and that $\phi(X) = \frac{1}{2} \langle X, A_G - 2A_H \rangle$, where A_G, A_H are the adjacency matrices of G and H.

Note that $L_G = D_G - A_G$ and $L_H = D_H - A_H$, and $\operatorname{tr}(D_G) = 2|E_G|$, $\operatorname{tr}(D_H) = 2|E_H|$. For any $X \succeq 0$ with $\operatorname{diag}(X) = \mathbb{I}$,

$$\begin{aligned}\langle X, A_G - 2A_H \rangle &= \langle X, (D_G - L_G) - 2(D_H - L_H) \rangle \\ &= 2(|E_G| - 2|E_H|) + \langle X, 2L_H - L_G \rangle \ .\end{aligned}$$

Suppose $\langle X, L_H\rangle < \frac{1}{2}\langle X, L_G\rangle$ for all $X \neq \vec{1}\vec{1}^\top$. Since $\langle \vec{1}\vec{1}^\top, L_G\rangle = \langle \vec{1}\vec{1}^\top, L_H\rangle = 0$, we have that the maximum of $\frac{1}{2}\langle X, A_G - 2A_H\rangle$ is $|E_G| - 2|E_H|$ and $X = \vec{1}\vec{1}^\top$ is the unique maximum.

For the other direction, suppose there is an $X \neq \vec{1}\vec{1}^\top$ such that $\langle X, L_H\rangle \geq \frac{1}{2}\langle X, L_G\rangle$. Then, $\phi(X) \geq |E_G| - 2|E_H| = \phi(\vec{1}\vec{1}^\top)$, meaning that $\vec{1}\vec{1}^\top$ is not the unique optimum. □

Relative spectral approximation from uniform subsamples. We now come to a key technical observation. Suppose that H is a *spectral sparsifier* of G, so that $v^\top(\frac{1}{\eta}L_H)v$ is $(1 \pm \delta)v^\top L_G v$ for any $v \in \mathbb{R}^n$. Then clearly $\langle X, L_H\rangle < \frac{1}{2}\langle X, L_G\rangle$ if $\eta < 1/2$ and $\delta = o(1)$, as we can write $X = \sum_{i=1}^n \lambda_i v_i v_i^\top$, and

$$\begin{aligned}\langle X, L_H\rangle &= \sum_{i=1}^n \lambda_i v_i^\top L_H v_i \leq \eta(1+\delta)\sum_{i=1}^n \lambda_i v_i^\top L_G v_i \\ &= \eta(1+\delta)\cdot\langle X, L_G\rangle < \frac{1}{2}\langle X, L_G\rangle \,.\end{aligned}$$

Furthermore, note that above we only required that $L_H \preceq \eta(1+\delta)L_G$, i.e., we only use the upper part of the spectral approximation.

We are now ready to state the key relative spectral approximation lemma. We observe that when H is a uniformly random subsample of G and G has a *spectral gap* and minimum degree $\mathrm{polylog}(n)$, then with high probability $L_H \preceq \eta(1+\delta)L_G$. We note that, while we do not provide a formal proof, the same argument using the lower tail of Matrix Chernoff can also establish a lower bound on L_H, which proves that H is indeed a spectral sparsifier of G.

Lemma II.6 (Relative spectral approximation from uniform subsamples)**.** *Let $\eta \in (0,1)$. Suppose $G = (V,E)$ is an n-vertex graph with minimum degree $d_{\min}$ (self-loops allowed) and spectral gap $\lambda_2(\widetilde{L}_G) = \lambda$ such that $d_{\min}\lambda > \frac{18}{\eta}\log n$, where $\widetilde{L}_G := D_G^{-1/2}L_G D_G^{-1/2}$ is the* normalized *Laplacian. Let H be a subgraph of G obtained by selecting each edge with probability η. Then, with probability at least $1 - O(n^{-2})$,*

$$L_H \preceq \eta(1+\delta)\cdot L_G$$

for $\delta = \sqrt{\frac{18\log n}{\eta d_{\min}\lambda}}$.

Proof. First, note that $\vec{1}$ lies in the kernel of both L_G and L_H, and because of the spectral gap of G, $\dim(\ker(L_G)) = 1$. Therefore, recalling that $L_G = D_G^{1/2}\widetilde{L}_G D_G^{1/2}$, it suffices to prove that

$$\left\|(\widetilde{L}_G^\dagger)^{1/2}D_G^{-1/2}L_H D_G^{-1/2}(\widetilde{L}_G^\dagger)^{1/2}\right\|_2 \leq \eta(1+\delta)\,.$$

Here $\widetilde{L}_G^\dagger$ is the pseudo-inverse of $\widetilde{L}_G$, and $\|\widetilde{L}_G^\dagger\|_2 \leq 1/\lambda$ because G has spectral gap λ. We will write $X := (\widetilde{L}_G^\dagger)^{1/2}D_G^{-1/2}L_H D_G^{-1/2}(\widetilde{L}_G^\dagger)^{1/2}$ for convenience.

Note that $L_G = \sum_{e\in E} L_e$, where $L_e \succeq 0$ is the Laplacian of a single edge e and $\|L_e\|_2 = 2$. Let $X_e = (\widetilde{L}_G^\dagger)^{1/2}D_G^{-1/2}L_e D_G^{-1/2}(\widetilde{L}_G^\dagger)^{1/2}$ if e is chosen in H and 0 otherwise. Then, $X = \sum_{e\in E} X_e$ and $\|\mathbb{E}[X]\|_2 = \eta$. Moreover, each X_e satisfies $X_e \succeq 0$ and $\|X_e\|_2 \leq \|\widetilde{L}_G^\dagger\|_2 \cdot \|D_G^{-1}\|_2 \cdot \|L_e\|_2 \leq \frac{2}{d_{\min}\lambda}$. Thus, by Matrix Chernoff (Fact III.3),

$$\Pr\left[\|X\|_2 \geq \eta(1+\delta)\right] \leq n\cdot\exp\left(-\frac{\delta^2\eta}{3}\cdot\frac{d_{\min}\lambda}{2}\right) \leq O(n^{-2})$$

as long as $\frac{18\log n}{\eta d_{\min}\lambda} \leq \delta^2 \leq 1$. □

Finishing the algorithm. By Lemmas II.5 and II.6, we can thus recover x^* exactly if the constraint graph G of ϕ has a nontrivial spectral gap and minimum degree $d_{\min} \geq \mathrm{polylog}(n)$. To finish the implementation of Strategy II.2, we thus need to explain how to algorithmically decompose any graph G into subgraphs $G_1, \dots, G_T$, each with reasonable min degree and nontrivial spectral gap, while only discarding a $o(1)$-fraction of the edges in G. This is the well-studied task of expander decomposition, for which we appeal to known results [34], [45], [46], [49].

This completes the high-level description of the algorithm in the even k case. Below, we summarize the steps of the final algorithm.

Algorithm II.7 (Algorithm for k-XOR for even k)**.**

Input: k-XOR instance ψ on n variables with m constraints and constraint hypergraph $\mathcal{H}$.

Output: Disjoint sets of constraints $\mathcal{A}_1, \mathcal{A}_2 \subseteq \mathcal{H}$ such that $|\mathcal{A}_1| \leq o(m)$ and only depends on $\mathcal{H}$, and $\mathcal{A}_2 = (\mathcal{H}\setminus\mathcal{A}_1)\cap\mathcal{E}_\psi$.

Operation:

1) Construct the 2-XOR instance ϕ with constraint graph G, as described in Definition II.1.
2) Remove small-degree vertices and run expander decomposition on G to produce expanders $G_1, \dots, G_T$. Set $\mathcal{A}_1$ to be the set of discarded constraints of size $o(m)$.
3) For each $i \in [T]$, solve the basic SDP on the subinstance ϕ_i defined by the constraints G_i. Let $\mathcal{A}_2^{(i)}$ denote the set of constraints violated by the optimal local SDP solution.
4) Output $\mathcal{A}_1$ and $\mathcal{A}_2 = \bigcup_{i=1}^T \mathcal{A}_2^{(i)}$.

E. The case of odd k

We are now ready to briefly explain the differences in the case when k is odd. For the purposes of this overview, we will focus only on the case of $k = 3$. Recall that we are given a 3-XOR instance ψ, specified by a 3-uniform hypergraph $\mathcal{H} \subseteq \binom{[n]}{3}$, as well as the right-hand sides $b_C \in \{-1,1\}$ for $C \in \mathcal{H}$, where $b_C = x_C^*$ with probability $1-\eta$ and $b_C = -x_C^*$ otherwise and $x^* \in \{-1,1\}^n$ is the planted assignment.

We now produce a 4-XOR instance using the well-known "Cauchy-Schwarz trick" from CSP refutation [18]. The general idea is to, for any pair of clauses (C, C') that intersect, add the "derived constraint" $x_C x_{C'} = b_C b_{C'}$ to the 4-XOR instance. Notice that if, e.g., $C = \{u,i,j\}$ and $C' = \{u,i',j'\}$, then x_u appears twice on the left-hand side, and thus the constraint is

$x_i x_j x_{i'} x_{j'} = b_C b_{C'}$. Given this 4-XOR, we produce a 2-XOR following a similar strategy as in Definition II.1. The above description omits many technical details, which we handle in Sections V and VI; we remark here that these are the same issues that arise in the CSP refutation case, and we handle them using the techniques in [29].

We have thus produced a 2-XOR instance ϕ that is noisy but not in the sense of Definition I.3. Indeed, each edge e in ϕ is "labeled" by a pair (C, C') of constraints in ψ, and e is noisy if and only if *exactly* one of (C, C') is, and so the noise is not independent across constraints. Nonetheless, we can still follow the general strategy as in Algorithm II.7. The main technical challenge is to argue that the relative spectral approximation guarantee of Lemma II.6 holds even when the noise has the aforementioned correlations, and we do this in Lemma VI.7. This allows us to recover, for most intersecting pairs (C, C'), the quantity $\xi(C)\xi(C')$, where $\xi(C) = -1$ if C is corrupted, and is 1 otherwise, i.e., $b_C = x^*_C \xi(C)$; we do not determine $\xi(C)\xi(C')$ if and only if the pair (C, C') corresponds to an edge e that was discarded during the expander decomposition.

However, we are not quite done, as we would like to recover $\xi(C)$ for most C, but we only know $\xi(C)\xi(C')$ for most intersecting pairs (C, C'). Let us proceed by assuming that we know $\xi(C)\xi(C')$ for all intersecting pairs (C, C'), and then we will explain how to do a similar decoding process when we only know most pairs. Let us fix a vertex u, and let $\mathcal{H}_u$ denote the set of $C \in \mathcal{H}$ containing u. Now, we know $\xi(C)\xi(C')$ for all $C, C' \in \mathcal{H}_u$, and so by Gaussian elimination we can determine $\xi(C)$ for all $C \in \mathcal{H}_u$ up to a global sign. Now, we know that the vector $\{\xi(C)\}_{C \in \mathcal{H}_u}$ should have roughly $\eta|\mathcal{H}_u|$ entries that are -1. So, choosing the global sign that results in fewer -1's, we thus correctly determine $\xi(C)$ for all $C \in \mathcal{H}_u$. We can then repeat this process for each choice of u to decode $\xi(C)$ for all C.

Of course, we only actually know $\xi(C)\xi(C')$ for most intersecting pairs (C, C'). This implies that for most choices of u, the graph G_u with vertices $\mathcal{H}_u$ and edges (C, C') if we know $\xi(C)\xi(C')$ is obtained from the complete graph on vertices $\mathcal{H}_u$ and deleting some $o(1)$-fraction of edges. This implies that G_u has a connected component of size $(1-o(1))|\mathcal{H}_u|$, and again via Gaussian elimination and picking the proper global sign, we can determine $\xi(C)$ on this large connected component. By repeating this process for each choice of u, we thus recover $\xi(C)$ for most u.

F. Organization

The rest of the paper is organized as follows. In Section III, we introduce some notation, and recall the various concentration inequalities and facts that we will use in our proofs. In Section IV, we prove Theorem 2 from Theorem 3 by reducing semirandom planted CSPs to noisy XOR. In Sections V and VI, we prove Theorem 3; Section V handles the reduction from k-XOR to "bipartite k-XOR", and then Section VI gives the algorithm for the bipartite k-XOR case.

III. Preliminaries

a) Notation.: Given a graph $G = (V, E)$ with n vertices and m edges (including self-loops[7]), we write $D_G \in \mathbb{R}^{n \times n}$ as the diagonal degree matrix, $A_G \in \mathbb{R}^{n \times n}$ as the adjacency matrix, and $L_G = D_G - A_G$ as the unnormalized Laplacian (note that the self-loops do not contribute to L_G). Furthermore, we write $\widetilde{L}_G = D_G^{-1/2} L_G D_G^{-1/2}$ to be the *normalized* Laplacian, and denote its eigenvalues as $0 = \lambda_1(\widetilde{L}_G) \le \lambda_2(\widetilde{L}_G) \le \cdots \le \lambda_n(\widetilde{L}_G) \le 2$.

For any subset $S \subseteq V$, we denote $G[S]$ as the subgraph of G induced by S, and $G\{S\}$ as the induced subgraph $G[S]$ but with self-loops added so that any vertex in S has the same degree as its degree in G.

Definition III.1 (Uniform hypergraphs)**.** A k-uniform hypergraph $\mathcal{H}$ on n vertices is a collection $\mathcal{H}$ of subsets of $[n]$ of size exactly k. For a set $Q \subseteq [n]$, we define $\deg(Q) := |\{C \in \mathcal{H} : Q \subseteq C\}|$.

A. Concentration inequalities

Fact III.2 (Chernoff bound)**.** *Let $X_1, \ldots, X_n$ be independent random variables taking values in $\{0, 1\}$. Let $X = \sum_{i=1}^n X_i$ and $\mu = \mathbb{E}[X]$. Then, for any $\delta \in [0, 1]$,*

$$\Pr\left[|X - \mu| \ge \delta\mu\right] \le 2e^{-\delta^2\mu/3}\,.$$

Fact III.3 (Matrix Chernoff [47, Theorem 5.1.1])**.** *Let $X_1, \ldots, X_n \in \mathbb{R}^{d \times d}$ be independent, random, symmetric matrices such that $X_i \succeq 0$ and $\lambda_{\max}(X_i) \le R$ almost surely. Let $X = \sum_{i=1}^n X_i$ and $\mu = \lambda_{\max}(\mathbb{E}[X])$. Then, for any $\delta \in [0, 1]$,*

$$\Pr\left[\lambda_{\max}(X) \ge (1+\delta)\mu\right] \le d \cdot \exp\left(-\frac{\delta^2\mu}{3R}\right)\,.$$

B. Graph pruning and expander decomposition

It is a standard result that given a graph with m edges and average degree d, one can delete vertices such that the resulting graph has minimum degree εd and at least $(1 - 2\varepsilon)m$ edges. We include a short proof for completeness.

Lemma III.4 (Graph pruning)**.** *Let G be an n-vertex graph with average degree d and $m = \frac{nd}{2}$ edges, and let $\varepsilon \in (0, 1/2)$. There is an algorithm that deletes vertices of G such that the resulting graph has minimum degree εd and at least $(1-2\varepsilon)m$ edges.*

Proof. The algorithm is simple: repeatedly remove any vertex with degree $< \varepsilon d$. First, we show by induction that each deletion cannot decrease the average degree. Suppose there are $n' \le n$ vertices left and average degree $d' \ge d$. Then, after deleting a vertex u with degree $d_u < \varepsilon d$, the average degree becomes $\frac{n'd' - 2d_u}{n'-1} > \frac{n'd - 2\varepsilon d}{n'-1} = d \cdot \frac{n' - 2\varepsilon}{n'-1}$. Thus, for $\varepsilon < 1/2$, the average degree is always at least d. Furthermore, since the algorithm can delete at most n vertices, it can delete at most $\varepsilon dn = 2\varepsilon m$ edges. □

[7] Each self-loop contributes 1 to the degree of a vertex.

We will also need an algorithm that partitions a graph into expanding clusters such that total number of edges across different clusters is small. Expander decomposition has been developed in a long line of work [34], [45], [46], [49] and has a wide range of applications. For our algorithm, we only require a very simple expander decomposition that recursively applies Cheeger's inequality.

Fact III.5 (Expander decomposition)**.** *Given a (multi)graph $G = (V, E)$ with m edges and a parameter $\varepsilon \in (0, 1)$, there is a polynomial-time algorithm that finds a partition of V into $V_1, \dots, V_T$ such that $\lambda_2(\widetilde{L}_{G\{V_i\}}) \geq \Omega(\varepsilon^2/\log^2 m)$ for each $i \in [T]$ and the number of edges across partitions is at most εm.*

Proof. Fix $\lambda = c\varepsilon^2/\log^2 m$ for some constant c to be chosen later. The algorithm is very simple. Given a graph $G = (V, E)$ (with potentially parallel edges and self-loops), if $\lambda_2(\widetilde{L}_G) < \lambda$, then by Cheeger's inequality we can efficiently find a subset $S \subseteq V$ with $\mathrm{vol}(S) \leq \mathrm{vol}(\overline{S})$ such that $\frac{|E(S,\overline{S})|}{\mathrm{vol}(S)} < \sqrt{2\lambda}$. Here $\mathrm{vol}(S) := \sum_{v \in S} \deg(v)$. Then, we cut along S, add self-loops to the induced subgraphs $G[S]$ and $G[\overline{S}]$ so that the vertex degrees remain the same (each self-loop contributes 1 to the degree). This produces two graphs $G\{S\}$ and $G\{\overline{S}\}$, and we recurse on each. By construction, in the end we will have partitions $V_1, \dots, V_T$ where either V_i is either a single vertex or satisfies $\lambda_2(\widetilde{L}_{G\{V_i\}}) \geq \lambda$.

We now bound the number of edges cut via a charging argument. Consider the "half-edges" in the graph, where each edge (u, v) contributes one half-edge to u and one to v, and each self-loop counts as one half-edge. Then, $\mathrm{vol}(S)$ equals the number of half-edges attached to S. Now, imagine we have a counter for each half-edge, and every time we cut along S we add $\sqrt{2\lambda}$ to each half-edge attached to S (the smaller side). Since $E(S, \overline{S}) < \sqrt{2\lambda} \cdot \mathrm{vol}(S)$, it follows that the number of edges cut is at most the total sum of the counters. On the other hand, each half-edge can appear on the smaller side of the cut at most $\log_2 2m$ times, as each time the half-edge is on the smaller side of the cut, $\mathrm{vol}(S)$ decreases by at least a factor of 2, and $\mathrm{vol}([n]) = 2m$. So, the total sum must be $\leq \sqrt{2\lambda} \cdot 2m \log_2 2m \leq \varepsilon m$ for a small enough constant c. □

IV. From Planted CSPs to Noisy XOR

In this section, we show how to use Theorem 3 to prove Theorem 2. Before we delve into the formal proof, we will first explain the reduction given in [23]. We begin with some definitions.

Setup. Let Ψ be sampled from $\Psi(\vec{\mathcal{H}}, x^*, Q)$, where $x^* \in \{-1, 1\}^n$, $\vec{\mathcal{H}} \subseteq [n]^k$, and Q is a planting distribution for the predicate P. Let $Q(y) = \sum_{S \subseteq [k]} \hat{Q}(S) \prod_{i \in S} y_i$ be the Fourier decomposition of Q, where $\hat{Q}(S) = \frac{1}{2^k} \sum_{y \in \{-1,1\}^k} Q(y) \prod_{i \in S} y_i \in [-2^{-k}, 2^{-k}]$. Recall (Definition I.2) that Ψ is specified by a collection $\vec{\mathcal{H}} \subseteq [n]^k$ of scopes, along with a vector $\ell(\vec{C}) \in \{-1, 1\}^k$ for each $\vec{C} \in \vec{\mathcal{H}}$ of literal negations.

Definition IV.1. Let $S \subseteq [k]$ be nonempty. Let $\psi^{(S,+)}$ be the $|S|$-XOR instance obtained by, for each constraint $\vec{C}$ in Ψ, adding the constraint $\prod_{i \in S} x_{\vec{C}_i} = \prod_{i \in S} \ell(\vec{C})_i$. Similarly, let $\psi^{(S,-)}$ have constraints $\prod_{i \in S} x_{\vec{C}_i} = -\prod_{i \in S} \ell(\vec{C})_i$.

We make use of the following simple claim.

Claim IV.2. For each nonempty $S \subseteq [k]$, $\psi^{(S,+)}$ is a noisy $|S|$-XOR instance (Definition I.3) with planted assignment x^* and noise $\eta = \frac{1}{2}(1 - 2^k \hat{Q}(S))$. Similarly, $\psi^{(S,-)}$ is a noisy $|S|$-XOR instance with planted assignment x^* and noise $\eta = \frac{1}{2}(1 + 2^k \hat{Q}(S))$.

Proof. For each $\vec{C}$, the literal negation $\ell(\vec{C})$ is sampled such that $\Pr[\ell(\vec{C}) = \ell] = Q(\ell \odot x^*_{\vec{C}})$, where $\odot$ denotes the element-wise product. This is equivalent to sampling $y \leftarrow Q$ and setting $\ell(\vec{C}) = y \odot x^*_{\vec{C}}$. It thus follows that the probability that the constraint $\vec{C}$ produces a corrupted constraint in $\psi^{(S,+)}$ is

$$\Pr_{y \leftarrow Q}\left[\prod_{i \in S} y_i = -1\right] = \frac{1}{2}\left(1 - \mathbb{E}_{y \leftarrow Q}\left[\prod_{i \in S} y_i\right]\right)$$
$$= \frac{1}{2}(1 - 2^k \hat{Q}(S)) \ ,$$

and is independent for each $\vec{C}$. A similar calculation handles the case of $\psi^{(S,-)}$. □

With the above observations in hand, we can now easily describe the reduction in [23]. First, their reduction requires the algorithm to have a description of the distribution Q. Given Q, the algorithm then finds the smallest S such that $\hat{Q}(S)$ is nonzero. Since they know the exact value of $\hat{Q}(S)$, they can determine its sign correctly. Suppose that $\hat{Q}(S) > 0$ (the other case is similar). Then, by solving the $|S|$-XOR instance $\psi^{(S,+)}$, they recover the planted assignment of $\psi^{(S,+)}$ *exactly*.[8] But this planted assignment is precisely x^*, and so they have also succeeded in recovering the planted assignment of ψ.

The aforementioned reduction clearly does not generalize to the semirandom setting, as in general the subinstances $\psi^{(S,\pm)}$ will not uniquely determine x^*. Furthermore, their reduction additionally requires knowing Q, and while it is not too unreasonable to assume this for random planted CSPs (as it is perhaps natural for the algorithm to know the distribution), in the semirandom setting this assumption is a bit strange because we want to view semirandom CSPs as "moving towards" worst case ones.

We now prove Theorem 2 from Theorem 3.

Proof of Theorem 2 from Theorem 3. We will present the proof in three steps. First, like [23], we will assume that the algorithm is given a description of Q and we will assume that each $|\hat{Q}(S)|$ is either 0 or at least $2^{-k}\varepsilon > 0$.[9] Then, we will remove this assumption provided that $Q(y) > 2\varepsilon$ for all y with $Q(y) > 0$, i.e., the every y in the support of Q has

[8] Here, they also treat $|\hat{Q}(S)|$ as constant, as if $|\hat{Q}(S)| \ll 1/n$, say, then their algorithm would not succeed in recovering the planted assignment on the XOR instance.

[9] This assumption is implicit in [23]; see the previous footnote.

some minimum probability. Finally, we will remove the last assumption.

Step 1: the proof when we are given Q. For each S where $\hat{Q}(S) \neq 0$, we construct the instance $\psi^{(S,+)}$ (if $\hat{Q}(S) > 0$) or $\psi^{(S,-)}$ (if $\hat{Q}(S) < 0$). We then apply[10] Theorem 3 to each such instance. Note that by Claim IV.2, the instance has noise $\eta = \frac{1}{2}(1 - 2^k|\hat{Q}(S)|) \leq \frac{1}{2}(1 - \varepsilon)$ (because we picked the correct sign when choosing between $\psi^{(S,+)}$ and $\psi^{(S,-)}$, and we assume $|\hat{Q}(S)| \geq 2^{-k}\varepsilon$). Then, since $m \geq c^k n^{k/2} \cdot \frac{\log^3 n}{\varepsilon^9}$ and $|S| \leq k$, by applying Theorem 3 with noise η and parameter $\varepsilon' := 2^{-k}\varepsilon$, we obtain sets $\vec{\mathcal{H}}^{(S,1)}$ (the discarded set) and $\vec{\mathcal{H}}^{(S,2)}$ (the corrupted constraints) where $|\vec{\mathcal{H}}^{(S,1)}| \leq \varepsilon' m$ and $\vec{\mathcal{H}}^{(S,2)} = (\vec{\mathcal{H}} \setminus \vec{\mathcal{H}}^{(S,1)}) \cap \mathcal{E}_{\psi(S)}$. Hence, for every constraint $\vec{C} \in \vec{\mathcal{H}} \setminus \vec{\mathcal{H}}^{(S,1)}$, it follows that we have learned $\prod_{i \in S} x^*_{\vec{C}_i}$, where x^* is the planted assignment for Ψ. By setting $\vec{\mathcal{H}}' := \vec{\mathcal{H}} \setminus \cup_{S : \hat{Q}(S) \neq 0} \vec{\mathcal{H}}^{(S,1)}$, it follows that we know $\prod_{i \in S} x^*_{\vec{C}_i}$ for all $\vec{C} \in \vec{\mathcal{H}}'$ and S with $\hat{Q}(S) \neq 0$, where $|\vec{\mathcal{H}}'| \geq (1 - 2^k\varepsilon')m = (1 - \varepsilon)m$.

We now solve the system of linear equations given by $\prod_{i \in S} x^*_{\vec{C}_i}$ for all $\vec{C} \in \vec{\mathcal{H}}'$ and S with $\hat{Q}(S) \neq 0$ to obtain some assignment $x \in \{-1,1\}^n$. As x^* is a valid solution to these equations, such an x exists, although it may not be x^*.

The final step is to argue that for every $\vec{C} \in \vec{\mathcal{H}}'$, x satisfies the constraint $\vec{C}$, namely that $P(\ell(\vec{C})_1 x_{\vec{C}_1}, \ell(\vec{C})_2 x_{\vec{C}_2}, \dots, \ell(\vec{C})_k x_{\vec{C}_k}) = 1$. Indeed, if this is true then we are done, as x satisfies at least $(1-\varepsilon)m$ constraints in Ψ, and so we have obtained the desired assignment.

Let $\vec{C} \in \vec{\mathcal{H}}'$. We know that for every S with $\hat{Q}(S) \neq 0$, we have that $\prod_{i \in S} x_{\vec{C}_i} = \prod_{i \in S} x^*_{\vec{C}_i}$. Hence, it follows that

$$\begin{aligned} Q(\ell(\vec{C}) \odot x) &= \sum_{S \subseteq [k]} \hat{Q}(S) \prod_{i \in S} \ell(\vec{C})_i x_{\vec{C}_i} \\ &= \sum_{S \subseteq [k]} \hat{Q}(S) \prod_{i \in S} \ell(\vec{C})_i x^*_{\vec{C}_i} \\ &= Q(\ell(\vec{C}) \odot x^*) > 0 \,, \end{aligned}$$

where the last inequality is because $\ell(\vec{C})$ was sampled from the distribution $Q(\ell(\vec{C}) \odot x^*)$, and so it must be sampled with nonzero probability. As Q is supported only on satisfying assignments to the predicate P, it thus follows that $\ell(\vec{C}) \odot x^*$ must also satisfy P.

Step 2: removing the dependence on Q assuming a lower bound on $Q(y)$. First, we observe that because k is constant, we can, for each S, guess a symbol $\{0,+,-\}$, where 0 denotes, informally, the belief that $|\hat{Q}(S)| < 2^{-k}\varepsilon$, $+$ denotes that $\hat{Q}(S) \geq 2^{-k}\varepsilon$, and $-$ denotes that $\hat{Q}(S) \leq -2^{-k}\varepsilon$. For each of the 3^{2^k} choices of guesses, i.e., functions $f \colon \{S \subseteq [k]\} \to \{0,+,-\}$, we run algorithm mentioned in the previous step. Namely, for each S: (1) if $f(S) = 0$, then we ignore S, (2) if $f(S) = +$, then we run Theorem 3 on $\psi^{(S,+)}$ to obtain $\vec{\mathcal{H}}^{(S,1)}$ and $\vec{\mathcal{H}}^{(S,2)}$, and (3) if $f(S) = -$, then we run Theorem 3 on $\psi^{(S,+)}$ to obtain $\vec{\mathcal{H}}^{(S,1)}$ and $\vec{\mathcal{H}}^{(S,2)}$. As before, we solve the system of linear equations to obtain some assignment $x^{(f)} \in \{-1,1\}^n$. By enumerating over all possible choices of f, we obtain a list of at most $3^{2^k} = O(1)$ assignments. We then try all of them and output the best one.

It thus remains to show that at least one of the assignments in the list has high value. As one may expect, this will be the assignment $x^{(f^*)}$, where f^* is the correct label function. Indeed, when $f = f^*$, then we are precisely running the algorithm in Step 1, and as observed, after solving the linear system of equations we obtain an assignment $x := x^{(f^*)}$ with the following property. For every $\vec{C} \in \vec{\mathcal{H}}'$ and every S with $|\hat{Q}(S)| \geq 2^{-k}\varepsilon$, we have that $\prod_{i \in S} x_{\vec{C}_i} = \prod_{i \in S} x^*_{\vec{C}_i}$, where $\vec{\mathcal{H}}' \subseteq \vec{\mathcal{H}}$ has size $\geq (1-\varepsilon)m$.

Finally, we show that for every $\vec{C} \in \vec{\mathcal{H}}'$, x satisfies the constraint $\vec{C}$. Namely, we have $P(\ell(\vec{C})_1 x_{\vec{C}_1}, \ell(\vec{C})_2 x_{\vec{C}_2}, \dots, \ell(\vec{C})_k x_{\vec{C}_k}) = 1$. Let $\vec{C} \in \vec{\mathcal{H}}'$. We know that for every S with $|\hat{Q}(S)| \geq 2^{-k}\varepsilon$, we have that $\prod_{i \in S} x_{\vec{C}_i} = \prod_{i \in S} x^*_{\vec{C}_i}$. Hence, it follows that

$$\begin{aligned} &\left| Q(\ell(\vec{C}) \odot x) - Q(\ell(\vec{C}) \odot x^*) \right| \\ &= \left| \sum_{S \subseteq [k]} \hat{Q}(S) \prod_{i \in S} \ell(\vec{C})_i x_{\vec{C}_i} - \sum_{S \subseteq [k]} \hat{Q}(S) \prod_{i \in S} \ell(\vec{C})_i x^*_{\vec{C}_i} \right| \\ &= \left| \sum_{S \subseteq [k] : |\hat{Q}(S)| < 2^{-k}\varepsilon} \hat{Q}(S) \left(\prod_{i \in S} \ell(\vec{C})_i x_{\vec{C}_i} - \prod_{i \in S} \ell(\vec{C})_i x^*_{\vec{C}_i} \right) \right| \\ &\leq 2^k \cdot 2^{-k+1}\varepsilon \,. \end{aligned}$$

Now, if we assume that $Q(y) > 2\varepsilon$ for every $y \in \{-1,1\}^k$ with $Q(y) > 0$, then it follows that $Q(\ell(\vec{C}) \odot x) > 0$, and so x satisfies the constraint $P(\ell(\vec{C})_1 x_{\vec{C}_1}, \ell(\vec{C})_2 x_{\vec{C}_2}, \dots, \ell(\vec{C})_k x_{\vec{C}_k}) = 1$.

Step 3: removing the lower bound on $Q(y)$. In Step 2, we assumed that $Q(y) > 2\varepsilon$ for all $y \in \{-1,1\}^k$ with $Q(y) > 0$. However, we only used this fact in the final step, when we argue that $Q(\ell(\vec{C}) \odot x) > 0$ by observing that $Q(\ell(\vec{C}) \odot x) \geq Q(\ell(\vec{C}) \odot x^*) - 2\varepsilon > 0$. To remove the assumption, we will show that for at most $2^{k+2}\varepsilon$ constraints $\vec{C} \in \vec{\mathcal{H}}$, it holds that $Q(\ell(\vec{C}) \odot x^*) \leq 2\varepsilon$. This then implies that x satisfies at least $(1 - \varepsilon - 2^{k+2}\varepsilon)m = (1 - O(\varepsilon))m$ constraints, which finishes the proof.

Let $\mathcal{S}$ denote the set of $\vec{C} \in \vec{\mathcal{H}}$ where $Q(\ell(\vec{C}) \odot x^*) \leq 2\varepsilon$. Observe that the probability, over the choice of $\ell(\vec{C})$, that $\vec{C} \in \mathcal{S}$ is at most $2^k \cdot 2\varepsilon = 2^{k+1}\varepsilon$, and moreover this is independent for each $\vec{C} \in \vec{\mathcal{H}}$. Thus, by a Chernoff bound, it follows that with probability $\geq 1 - \exp(-O(\varepsilon m)) \geq 1 - 1/\mathrm{poly}(n)$, it holds that $|\mathcal{S}| \leq 2 \cdot 2^{k+1}\varepsilon$, and so we are done. □

Remark IV.3 (Tolerating fewer constraints for structured Q's). We have shown that the above algorithm succeeds in finding an assignment x that satisfies at least $(1 - O(\varepsilon))m$ constraints when $m \geq n^{k/2} \cdot \mathrm{poly}(\log n, 1/\varepsilon)$. However, if the distribution Q has $|\hat{Q}(S)| < 2^{-k}\varepsilon$ for all S with $|S| > r$, then we

[10] Note that Theorem 3 only applies when $|S| \geq 2$. When $|S| = 1$, there is a trivial algorithm; see Section C for details.

only need $n^{r/2} \cdot \mathrm{poly}(\log n, 1/\varepsilon)$ constraints. (If $r = 0$, then for small enough constant ε, Q will be supported on all of $\{-1,1\}^k$, and so any assignment satisfies all constraints. If $r = 1$, we require $O(n \cdot \frac{\log n}{\varepsilon})$ constraints; see Lemma C.1.) Indeed, this follows because for such Q, the true label function f^* will have $f^*(S) = 0$ for any S with $|S| > r$. Hence, for this choice of f^*, we only call Theorem 3 on noisy t-XOR instances for $t \leq r$, and so we have enough constraints. It therefore follows that the assignment $x^{(f^*)}$ that we obtain for the label function f^* will be, with high probability an assignment that satisfies at least $(1 - O(\varepsilon))m$ constraints.

An example where this gives an improvement is the well-studied NAE-3-SAT (not-all-equal-3SAT) predicate [3], [5], [19]. Suppose Q is the uniform distribution over satisfying assignments to NAE-3-SAT: $Q(x_1, x_2, x_3) = \frac{1}{6} \cdot \frac{1}{4}(3 - x_1 x_2 - x_2 x_3 - x_1 x_3)$. Then, we only need $m \geq \tilde{O}(n)$ constraints, even though it is a 3-CSP ($k = 3$).

V. From k-XOR to spread bipartite k-XOR

In this section, we begin the proof of Theorem 3. See Definition I.3 for a reminder of our semirandom planted k-XOR model $\psi(\mathcal{H}, x^*, \eta)$ given a k-uniform hypergraph $\mathcal{H}$, assignment $x^* \in \{-1,1\}^n$, and noise parameter $\eta \in (0, 1/2)$. Recall also that $\mathcal{E}_\psi$ denotes the set of corrupted hyperedges.

We think of $\mathcal{A}_1(\mathcal{H})$ as the small set of edges that we discard (or give up on), and this will only depend on the hypergraph $\mathcal{H}$. For the rest of the graph, the algorithm will correctly identify which edges are corrupted.

Our proof of Theorem 3 goes via a reduction to *spread bipartite t-XOR* instances for $t = 2, \dots, k$, which are t-XOR instances with some additional desired structure. Such instances were introduced in [29] to study the refutation of semirandom k-XOR instances. The reduction here is nearly identical to the corresponding reduction in [29, Section 4].

Definition V.1 (Spread bipartite k-XOR)**.** A p-bipartite k-XOR instance ψ on n variables with m constraints is defined by a collection of $(k-1)$-uniform hypergraphs $\mathcal{H} = \{\mathcal{H}_u\}_{u \in [p]}$ on the vertex set $[n]$, as well as "right-hand sides" $b_{u,C}$ for each $u \in [p]$ and $C \in \mathcal{H}_u$. There are two sets of variables of ψ: the "normal" variables $x_1, \dots, x_n$, and the "special" variables $y_1, \dots, y_p$. The constraints of ψ are $y_u \prod_{i \in C} x_i = b_{u,C}$ for each $u \in [p]$, $C \in \mathcal{H}_u$.

We furthermore say that ψ is *τ-spread* if it has the following additional properties:

(1) $|\mathcal{H}_u| = \frac{m}{p} \geq 2\lfloor \frac{1}{2\tau^2} \rfloor$ and $\frac{m}{p}$ is even for each $u \in [p]$,
(2) For each $u \in [p]$ and set $Q \subseteq [n]$, $\deg_u(Q) \leq \frac{1}{\tau^2} \max(1, n^{\frac{k}{2} - 1 - |Q|})$.

Analogously to Definition I.3, we call ψ a *semirandom planted* instance with planted assignment (x^*, y^*) and noise parameter η if the right-hand sides $b_{u,C}$ are generated by setting $b_{u,C} = y^*_u \prod_{i \in C} x^*_i$ with probability $1 - \eta$ and $b_{u,C} = -y^*_u \prod_{i \in C} x^*_i$ otherwise, independently for each choice of u, C. For a choice of x^*, y^*, $\mathcal{H} = \{\mathcal{H}_u\}_{u \in [p]}$, and η, we call this distribution $\psi(\{\mathcal{H}_u\}_{u \in [p]}, x^*, y^*, \eta)$. As before, if an edge (u, C) has $b_{u,C} = -y^*_u \prod_{i \in C} x^*_i$, we call (u, C) a *corrupted* hyperedge, and we denote the set of corrupted hyperedges in ψ by $\mathcal{E}_\psi$.

The main technical result of the paper is the following lemma, which gives an algorithm to find the noisy constraints in a semirandom planted τ-spread bipartite k-XOR instance.

Lemma V.2 (Algorithm for τ-spread bipartite k-XOR)**.** *Let $k \geq 2$, $n, p \in \mathbb{N}$, $\varepsilon \in (0,1)$, $\eta \in [0, 1/2)$, and let $\gamma := 1 - 2\eta > 0$. Let $\tau \leq \frac{c\gamma}{\sqrt{k \log n}}$, and let $m \geq C n^{\frac{k-1}{2}} \sqrt{p} \cdot \frac{(k \log n)^{3/2}}{\tau \gamma^2 \varepsilon^{3/2}}$ for some universal constants c, C. There is a polynomial-time algorithm $\mathcal{A}$ that takes as input an τ-spread p-bipartite k-XOR instance ψ with constraint hypergraph $\mathcal{H} = \{\mathcal{H}_u\}_{u \in [p]}$ and outputs two disjoint sets $\mathcal{A}_1(\mathcal{H}), \mathcal{A}_2(\psi) \subseteq \mathcal{H}$ with the following guarantee: (1) for any instance ψ with m constraints, $|\mathcal{A}_1(\mathcal{H})| \leq \varepsilon m$ and $\mathcal{A}_1(\mathcal{H})$ only depends on $\mathcal{H}$, and (2) for any $x^* \in \{-1,1\}^n, y^* \in \{-1,1\}^p$ and any $\mathcal{H} = \{\mathcal{H}_u\}_{u \in [p]}$ with $|\mathcal{H}| := \sum_{u \in [p]} |\mathcal{H}_u| \geq m$, with probability $1 - \frac{1}{\mathrm{poly}(n)}$ over $\psi \leftarrow \psi(\{\mathcal{H}_u\}_{u \in [p]}, x^*, y^*, \eta)$, it holds that $\mathcal{A}_2(\psi) = \mathcal{E}_\psi \cap (\mathcal{H} \setminus \mathcal{A}_1(\mathcal{H}))$.*

Note that as $\eta \to \frac{1}{2}$, $\gamma = 1 - 2\eta \to 0$ and $\tau \to 0$, which blows up m. This is the expected behavior since when $\eta = \frac{1}{2}$, it is impossible to recover the planted assignment since the signs of the constraints are uniformly random.

A. Proof of Theorem 3 from Lemma V.2

With Lemma V.2, we can finish the proof of Theorem 3. The high-level idea of this proof is very simple. First, we decompose the k-XOR instance ψ into subinstances $\psi^{(t)}$ for each $t = 2, \dots, k$, using a hypergraph decomposition algorithm very similar to the one used in [29], [31]. The algorithm and its guarantees are shown in Section B. Then, we run the algorithm in Lemma V.2 to identify a set of corrupted constraints and a small set of discarded constraints within each subinstance $\psi^{(t)}$. We then take the union of these outputs to be the final output of the algorithm.

Proof of Theorem 3. We begin with the decomposition of ψ into $\psi^{(2)}, \dots, \psi^{(k)}$ along with a set of "discarded" hyperedges $\mathcal{H}^{(1)}$, which is done using Algorithm B.1 with spread parameter $\tau := \frac{c(1-2\eta)}{\sqrt{k \log n}}$ where c is the constant in Lemma V.2. For each $t = 2, \dots, k$, $\psi^{(t)}$ is a semirandom (with noise η) planted τ-spread $p^{(t)}$-bipartite t-XOR instance specified by $(t-1)$-uniform hypergraphs $\{\mathcal{H}^{(t)}_u\}_{u \in [p^{(t)}]}$.

Let $m^{(t)} := \sum_{u \in [p^{(t)}]} |\mathcal{H}^{(t)}_u|$. Algorithm B.1 has the following guarantees:

(1) The runtime is $n^{O(k)}$,
(2) For each $t \in \{2, \dots, k\}$ and $u \in [p^{(t)}]$, $|\mathcal{H}^{(t)}_u| = \frac{m^{(t)}}{p^{(t)}} = 2\lfloor \frac{1}{2\tau^2} \max(1, n^{t - \frac{k}{2} - 1}) \rfloor$; in particular, $|\mathcal{H}^{(t)}_u|$ is even and is at least $2\lfloor \frac{1}{2\tau^2} \rfloor$,
(3) For each $t = 2, \dots, k$, the instance $\psi^{(t)}$ is τ-spread,
(4) The number of "discarded" hyperedges is $m^{(1)} := |\mathcal{H}^{(1)}| \leq \frac{1}{k\tau^2} n^{\frac{k}{2}}$,
(5) For $t \in \{2, \dots, k\}$, each $C \in \mathcal{H}^{(t)}_u$ is obtained by removing $k - (t-1)$ vertices from an edge in the

original hypergraph $\mathcal{H}$. Thus, there is a one-to-one map $\mathsf{Decomp}\colon \mathcal{H} \to \mathcal{H}^{(1)} \cup \bigcup_{t=2}^{k} \{\mathcal{H}_u^{(t)}\}_{u \in [p^{(t)}]}$, such that an edge $C \in \mathcal{H}$ is corrupted if and only if the edge $\mathsf{Decomp}(C)$ is corrupted in the instance $\psi^{(t)}$ that it lies in.

For convenience, we denote $\gamma := 1 - 2\eta$ and $\beta := 4C \cdot \frac{(k \log n)^{3/2}}{\tau \gamma^2 \varepsilon^{3/2}} = \frac{4C}{c} \cdot \frac{k^2 \log^2 n}{\gamma^3 \varepsilon^{3/2}}$ where C, c are the constants in Lemma V.2. The algorithm in Theorem 3 works as follows. First, it runs Algorithm B.1 to produce the instances $\psi^{(2)}, \dots, \psi^{(k)}$. Then, for each $t = 2, \dots, k$, if $m^{(t)} \geq n^{\frac{t-1}{2}} \sqrt{p^{(t)}} \cdot \beta$, we run Lemma V.2 on $\psi^{(t)}$ and obtain, with probability $1 - 1/\mathrm{poly}(n)$, a set $A_1^{(t)}$ where $|A_1^{(t)}| \leq \frac{\varepsilon}{2} m^{(t)}$ and $A_2^{(t)} = \mathcal{E}_{\psi^{(t)}} \setminus A_1^{(t)}$. Otherwise, if $m^{(t)} < n^{\frac{t-1}{2}} \sqrt{p^{(t)}} \cdot \beta$, we set $A_1^{(t)} = \mathcal{H}^{(t)}$ and $A_2^{(t)} = \emptyset$. Finally, we output $\mathcal{A}_1 := \mathcal{H}^{(1)} \cup \bigcup_{t=2}^{k} \mathsf{Decomp}^{-1}(A_1^{(t)})$ and $\mathcal{A}_2 := \bigcup_{t=2}^{k} \mathsf{Decomp}^{-1}(A_2^{(t)})$, where Decomp is the mapping in property (5) of Algorithm B.1.

Note that $m^{(t)} = p^{(t)} |\mathcal{H}_u^{(t)}| \geq p^{(t)} \cdot \frac{1}{2\tau^2} n^{t - \frac{k}{2} - 1}$, which means $p^{(t)} \leq 2\tau^2 n^{\frac{k}{2} - t + 1} m^{(t)}$, and since $\sum_t \sqrt{m^{(t)}} \leq \sqrt{k \sum_t m^{(t)}} \leq \sqrt{km}$ by Cauchy-Schwarz, we have

$$\sum_{t=2}^{k} n^{\frac{t-1}{2}} \sqrt{p^{(t)}} \cdot \beta \leq O(\tau) \cdot n^{\frac{k}{4}} \sqrt{km} \cdot \beta \leq o(\varepsilon) m$$

as long as $m \gg n^{\frac{k}{2}} \cdot k\tau^2 \beta^2 / \varepsilon^2$. Moreover, $m^{(1)} \leq \frac{1}{k\tau^2} n^{\frac{k}{2}} = \frac{\log n}{c^2 \gamma^2} n^{\frac{k}{2}} \leq o(\varepsilon) m$. One can verify, by plugging in β, that the lower bound on m in Theorem 3 suffices.

By union bound over t, it thus follows that

$$|\mathcal{A}_1| \leq m^{(1)} + \sum_{t=2}^{k} \frac{\varepsilon}{2} m^{(t)} + \sum_{t=2}^{k} n^{\frac{t-1}{2}} \sqrt{p^{(t)}} \beta \leq \varepsilon m \,,$$

and $\mathcal{A}_2 = \mathcal{E}_\psi \setminus \mathcal{A}_1$. Moreover, by Lemma V.2, $\mathcal{A}_1$ only depends on the hypergraph $\mathcal{H}$. This completes the proof. □

VI. Identifying Noisy Constraints in Spread Bipartite k-XOR

In this section, we prove Lemma V.2. The proof will be decomposed into the following steps. First, we take the semirandom planted bipartite k-XOR instance ψ and transform it into a 2-XOR instance ϕ. Second, we decompose the constraint graph of ϕ into expanders. For each expander in the decomposition, we argue that the SDP solution to this subinstance is rank 1, and moreover agrees *exactly* with the planted assignment. This allows us to identify, for each expanding subinstance, *exactly* which edges in ϕ are errors. Finally, we use this information to identify the set of corrupted constraints in the original instance ψ, which finishes the proof.

A. Setup and key notation

We now introduce the key notation that shall be used throughout this section. Let ψ be the semirandom τ-spread p-bipartite k-XOR instance (recall Definition V.1) with m constraints given as the input to the algorithm. Recall that the instance ψ is specified by a collection of p hypergraphs $\{\mathcal{H}_u\}_{u \in [p]}$, where each $\mathcal{H}_u$ is a $(k-1)$-uniform hypergraph on n vertices and $|\mathcal{H}_u| = m/p$. Each constraint in ψ is specified by a pair (u, C) where $u \in [p]$, $C \in \mathcal{H}_u$, and has a right-hand side $b_{u,C} \in \{-1, 1\}$, and the constraints are $y_u \prod_{i \in C} x_i = b_{u,C}$, where $\{y_u\}_{u \in [p]}$ and $\{x_i\}_{i \in [n]}$ are variables. Because the instance ψ is semirandom with noise parameter η and planted assignment (x^*, y^*), for each constraint (u, C) we have, with probability $1 - \eta$ independently, $b_{u,C} = y_u^* \prod_{i \in C} x_i^*$, and otherwise $b_{u,C} = -y_u^* \prod_{i \in C} x_i^*$. Our goal is to output, in $n^{O(k)}$-time, a set $\mathcal{A}_1(\mathcal{H})$ of size $\leq \tau m$ to discard, and then for the rest of the instance, identify exactly the corrupted constraints, i.e., those for which $b_{u,C} = -y_u^* \prod_{i \in C} x_i^*$.

We now define the 2-XOR instance ϕ from ψ. An example is shown in Fig. 2.

Definition VI.1 **(2-XOR instance ϕ from bipartite k-XOR ψ).** For every $u \in [p]$ and $\mathcal{H}_u$, we partition $\mathcal{H}_u$ arbitrarily into two sets $\mathcal{H}_u^{(L)}$ and $\mathcal{H}_u^{(R)}$ of equal size.

- If k is odd, then there are $\binom{n}{\frac{k-1}{2}}^2$ variables in ϕ, one variable $z_{(S_1, S_2)}$ for each pair of sets $S_1, S_2 \subseteq [n]$ where $|S_1| = |S_2| = \frac{k-1}{2}$.
- If k is even, then there are $2\binom{n}{\lceil \frac{k-1}{2} \rceil}\binom{n}{\lfloor \frac{k-1}{2} \rfloor}$ variables in ϕ, one variable $z_{(S_1, S_2)}$ for each pair of sets $S_1, S_2 \subseteq [n]$ where either $|S_1| = \lceil \frac{k-1}{2} \rceil$ and $|S_2| = \lfloor \frac{k-1}{2} \rfloor$ or $|S_1| = \lfloor \frac{k-1}{2} \rfloor$ and $|S_2| = \lceil \frac{k-1}{2} \rceil$.

For each $u \in [p]$, $C \in \mathcal{H}_u^{(L)}$ and $C' \in \mathcal{H}_u^{(R)}$, we arbitrarily partition C into sets $S_1 \cup S_2$ and C' into sets $S_1' \cup S_2'$, where $|S_1| = |S_1'| = \lceil \frac{k-1}{2} \rceil$ and $|S_2| = |S_2'| = \lfloor \frac{k-1}{2} \rfloor$. We then add the constraint $z_{(S_1, S_2')} z_{(S_2, S_1')} = b_{u,C} b_{u,C'}$ to ϕ.

It is intuitive to think of clauses from $\mathcal{H}_u^{(L)}$ and $\mathcal{H}_u^{(R)}$ as having different colors, and each variable $z_{(S_1, S_2')}$ contains roughly $k/2$ of each color. See Fig. 2 for an example of a 2-XOR ϕ constructed from a bipartite k-XOR ψ.

Observation VI.2 (Size of ϕ). The number of variables in ϕ is at most n^{k-1} (for both even and odd k). Since each $|\mathcal{H}_u| = m/p$, $|\mathcal{H}_u^{(L)}| = |\mathcal{H}_u^{(R)}| = \frac{m}{2p}$, and the number of constraints in ϕ is exactly $p \cdot (\frac{m}{2p})^2 = \frac{m^2}{4p}$. In particular, when $m \geq n^{\frac{k-1}{2}} \sqrt{p} \cdot \beta$ for $\beta = \mathrm{poly}(\log n)$ as assumed in Lemma V.2, the average degree of ϕ is at least $\frac{1}{4}\beta^2$.

Remark VI.3 (Corrupted constraints in ϕ). A constraint $z_{(S_1, S_2')} z_{(S_2, S_1')} = b_{u,C} b_{u,C'}$ in ϕ is *corrupted* if exactly one of $b_{u,C}$ and $b_{u,C'}$ is corrupted in ψ. Thus, if each constraint in ψ is corrupted with probability $\eta \in (0, 1/2)$, then each constraint in ϕ is corrupted with probability $2\eta(1 - \eta) < 1/2$. Note, however, that the constraints in ϕ are not corrupted independently.

We need some more definitions about the constraint graph of ϕ.

Definition VI.4 (Constraint graph of ϕ)**.** Let $G(\phi) = (V, E)$ be the constraint graph of ϕ. Notice that each edge $e \in E$ uniquely identifies $u(e) \in [p]$ and $C_L(e) \in \mathcal{H}_{u(e)}^{(L)}$, $C_R(e) \in \mathcal{H}_{u(e)}^{(R)}$. For each $u \in [p]$, $C \in \mathcal{H}_u^{(L)}$, define

$G^{(L)}_{u,C}(\phi)$ to be the subgraph of G that C participates in, i.e., with edge set $\{e \in E : u(e) = u,\ C_L(e) = C\}$. We similarly define $G^{(R)}_{u,C'}(\phi)$ for $C' \in \mathcal{H}^{(R)}_u$.

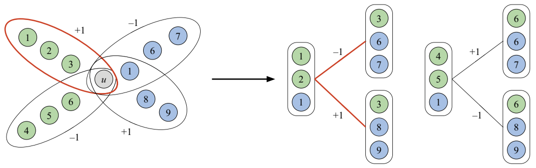

Fig. 2: An example of the 2-XOR instance ϕ from a bipartite 4-XOR ψ (Definition VI.1). On the left, $\mathcal{H}^{(L)}_u$ consists of $C_1 = \{1, 2, 3\}$ and $C_2 = \{4, 5, 6\}$ (with green vertices), and $\mathcal{H}^{(R)}_u$ consists of $C'_1 = \{1, 6, 7\}$ and $C'_2 = \{1, 8, 9\}$ (with blue vertices). On the right, the constraint graph $G(\phi)$ has vertices z_{S_1,S_2} where either $|S_1| = 2$, $|S_2| = 1$ or $|S_1| = 1$, $|S_2| = 2$ (we can view S_1, S_2 as having green, blue vertices). Each edge corresponds to two clauses in ψ; for example, the edge $\left\{z_{\{1,2\},\{1\}}, z_{\{3\},\{6,7\}}\right\}$ comes from the clauses C_1 and C'_1. **Corruptions.** In the figure, we label a clause -1 if it is corrupted and $+1$ otherwise. An edge in G is corrupted if exactly one of the two corresponding clauses in ψ is corrupted. **Degree of $G^{(L)}_{u,C}(\phi)$.** For $C_1 \in \mathcal{H}^{(L)}_u$, the subgraph $G^{(L)}_{u,C_1}(\phi)$ corresponds to the edges colored red, i.e., all edges that C_1 participates in. The vertex $z_{\{1,2\},\{1\}}$ has degree 2 in $G^{(L)}_{u,C_1}(\phi)$ because $|C'_1 \cap C'_2| = 1$.

We next make the important observation that the degree of a vertex in $G^{(L)}_{u,C}(\phi)$ is upper bounded by the number of $C' \in \mathcal{H}^{(R)}_u$ sharing at least $\lfloor \frac{k-1}{2} \rfloor$ vertices. See Fig. 2 also for an illustration. Therefore, assuming that ψ is τ-spread, we have a maximum degree bound on $G^{(L)}_{u,C}(\phi)$ and $G^{(R)}_{u,C'}(\phi)$ for all $u \in [p]$, $C \in \mathcal{H}^{(L)}_u$ and $C' \in \mathcal{H}^{(R)}_u$.

Lemma VI.5 (Degree bounds for $G^{(L)}_{u,C}$, $G^{(R)}_{u,C'}$)**.** *Let ψ be an τ-spread p-bipartite k-XOR instance. Then, for any $u \in [p]$, $C \in \mathcal{H}^{(L)}_u$ and $C' \in \mathcal{H}^{(R)}_u$, the maximum degree of $G^{(L)}_{u,C}(\phi)$, $G^{(R)}_{u,C'}(\phi)$ is at most $1/\tau^2$.*

Proof. Consider any $C \in \mathcal{H}^{(L)}_u$ and two adjacent edges $\{z_{(S_1,S'_2)}, z_{(S_2,S'_1)}\}$ and $\{z_{(S_1,S''_2)}, z_{(S_2,S''_1)}\}$ in $G^{(L)}_{u,C}(\phi)$ formed by joining $C = S_1 \cup S_2$ with $C' = S'_1 \cup S'_2$ and $C'' = S''_1 \cup S''_2 \in \mathcal{H}^{(R)}_u$. As the edges are adjacent, it must be the case that either $S'_1 = S''_1$ or $S'_2 = S''_2$, which means that $|C' \cap C''| \geq \lfloor \frac{k-1}{2} \rfloor$. Thus, the degree of a vertex $z_{(S_1,S'_2)}$ in G is upper bounded by the maximum number of $C' \in \mathcal{H}^{(R)}_u$ that all share the same $\lfloor \frac{k-1}{2} \rfloor$ variables.

Suppose ψ is τ-spread, meaning that $\deg_u(Q) \leq \frac{1}{\tau^2} \max(1, n^{\frac{k}{2}-1-|Q|})$ for $Q \subseteq [n]$. Since $\frac{k}{2} - 1 - \lfloor \frac{k-1}{2} \rfloor \leq 0$, we have that $G^{(L)}_{u,c}(\phi)$ has maximum degree $\leq 1/\tau^2$. □

B. Proof outline

With the setup in Section VI-A in hand, our proof now proceeds in three conceptual steps.

Step 1: graph pruning and expander decomposition. Suppose the instance ϕ has average degree d. We first prune the instance using Lemma III.4 such that the resulting constraint graph has minimum degree $\geq \varepsilon d$ while only removing ε fraction of the constraints, where $\varepsilon = o(1)$. We further apply expander decomposition (Fact III.5) to the pruned instance to obtain subinstances $\phi_1, \ldots, \phi_T$ while discarding only a ε fraction of the constraints of ϕ such that the constraint graph of each ϕ_i has spectral gap $\widetilde{\Omega}(\varepsilon^2)$.

Step 2: relative spectral approximation and recovery of corrupted pairs. We show that for each expanding subinstance ϕ_i, the basic SDP for the 2-XOR instance ϕ_i is equal to $x^*(x^*)^\top$, where x^* is the planted assignment for ϕ. That is, the SDP solution is *rank* 1 and agrees with the *planted assignment* for ϕ. We show this by arguing that, for each ϕ_i, the Laplacian of the corrupted constraints in ϕ_i is a *spectral sparsifier* of the Laplacian of the constraint graph of ϕ_i (see Lemma II.5). Here, we crucially use that each such constraint graph has large minimum degree and spectral gap.

From this, it is trivial to identify the corrupted edges in each ϕ_i, as they are the ones violated by the SDP solution. We are not quite done yet, however, because each constraint in ϕ corresponds to a *pair* of constraints in the original instance ψ.

Step 3: recovery of corrupted constraints from corrupted pairs. The previous step shows that for all but a ε fraction of tuples (u, C, C') where $u \in [p]$, $C \in \mathcal{H}^{(L)}_u$, and $C' \in \mathcal{H}^{(R)}_u$, we can recover the product $\xi_u(C)\xi_u(C')$, where $\xi_u(C) = -1$ if (u, C) is noisy in ψ, and is $+1$ otherwise. Because ε is small, it must be the case that for most $u \in [p]$, we know the product $\xi_u(C)\xi_u(C')$ (from Step 2) for *most* pairs (C, C') with $C \in \mathcal{H}^{(L)}_u$ and $C' \in \mathcal{H}^{(R)}_u$.

Suppose we knew $\xi_u(C)\xi_u(C')$ for all $(C, C') \in \mathcal{H}^{(L)}_u \times \mathcal{H}^{(R)}_u$. Then, it is trivial to decode $\xi_u(C)$ *up to a global sign*. Formally, we could obtain $z \in \{-1, 1\}^{\mathcal{H}_u}$ where $z_C = \alpha\xi_u(C)$ for some $\alpha \in \{-1, 1\}$. From this, it is easy to obtain $\xi_u(C)$, as the fraction of $C \in \mathcal{H}_u$ for which $\xi_u(C) = -1$ should be roughly $\eta < \frac{1}{2}$; so, if z has $< \frac{1}{2}$-fraction of -1's, then $z = \xi_u(C)$, and otherwise $-z = \xi_u(C)$. This, however, requires $|\mathcal{H}_u| \geq \Omega\left(\frac{\log n}{(1-2\eta)^2}\right)$ for a high-probability result.

Additionally, we do not quite know $\xi_u(C)\xi_u(C')$ for all $(C, C') \in \mathcal{H}^{(L)}_u \times \mathcal{H}^{(R)}_u$: we only know this for all but a ε_u-fraction of the pairs. By forming a graph G_u where we have an edge (C, C') if (C, C') is a pair where we know $\xi_u(C)\xi_u(C')$, we can thus obtain such a z for all C in the largest connected component of G_u. Because G_u is obtained by taking a *complete biclique* and deleting only a ε_u-fraction of all edges, the largest connected component has size $(1 - \varepsilon_u)|\mathcal{H}_u|$, and so we can recover $\xi_u(C)$ for all but a ε_u-fraction of constraints in $\mathcal{H}_u$. We do this for each partition u, which finishes the proof.

C. Graph pruning and expander decomposition

This step is a simple combination of graph pruning and expander decomposition.

Lemma VI.6. *Fix $\varepsilon \in (0,1)$. There is a polynomial-time algorithm such that, given a 2-XOR instance ϕ whose constraint graph has m edges and average degree d, outputs subinstances $\phi_1, \dots, \phi_T$ on disjoint variables with the following guarantees: $\phi_1, \dots, \phi_T$ contain at least $1-\varepsilon$ fraction of the constraints in ϕ, and for each $i \in [T]$, the constraint graph G_i of ϕ_i, after adding some self-loops, has minimum degree at least $\frac{1}{3}\varepsilon d$ and $\lambda_2(\widetilde{L}_{G_i}) \geq \Omega(\varepsilon^2/\log^2 m)$.*

The self-loops in Lemma VI.6 are only for the analysis of $\widetilde{L}_{G_i}$ and do not correspond to actual constraints in ϕ_i. Observe that adding self-loops to a graph G does not change the *unnormalized* Laplacian L_G, but as D_G (the degree matrix) increases, the spectral gap of the *normalized* Laplacian, i.e. $\lambda_2(\widetilde{L}_G) = \lambda_2(D_G^{-1/2} L_G D_G^{-1/2})$, may decrease. The expander decomposition algorithm (Fact III.5) guarantees that each piece, even after adding self-loops to preserve degrees, has large spectral gap. This does not change the subinstances $\phi_1, \dots, \phi_T$, but in the next section, it is crucial that we use this stronger guarantee to ensure a lower bound on the minimum degree.

Proof of Lemma VI.6. We first apply the graph pruning algorithm (Lemma III.4) such that the resulting instance has minimum degree $\geq \frac{\varepsilon}{3}d$ and at least $(1-\frac{2}{3}\varepsilon)m$ constraints. Then, we apply expander decomposition (Fact III.5) that partitions the vertices of the pruned graph G' into $V_1, \dots, V_T$ such that the number of edges across partitions is at most $\frac{\varepsilon}{3}m$, and for each $i \in [T]$, the normalized Laplacian satisfies $\lambda_2(\widetilde{L}_{G'\{V_i\}}) \geq \Omega(\varepsilon^2/\log^2 m)$. Here we recall that $G'\{V_i\}$ is the induced subgraph of G' with self-loops such that the vertices in $G'\{V_i\}$ have the same degrees as in G'.

In total, we have removed at most εm edges. This completes the proof. □

D. Rank-1 SDP solution from expansion and relative spectral approximation

We next show that for each subinstance ϕ_i obtained from Lemma VI.6, its constraint graph G and the subgraph of corrupted edges H satisfy $L_H \prec \frac{1}{2}L_G$. Recall from Lemmas II.5 and II.6 that this implies the basic SDP for the 2-XOR ϕ_i is rank 1 and agrees with the planted assignment of ϕ.

The next lemma is analogous to Lemma II.6 but differs in an important way: a constraint in ϕ is corrupted if and only if exactly one of the two corresponding constraints in ψ is corrupted; thus, the corruptions in ϕ are *correlated*. This is why each constraint in ϕ is obtained from one clause in $\mathcal{H}_u^{(L)}$ and one clause in $\mathcal{H}_u^{(R)}$ (recall Definition VI.1), so that in the proof below we have independent randomness to perform a "2-step sparsification" proof. It is also worth noting that the following lemma requires not just a lower bound on the minimum degree and spectral gap of G but also that the original bipartite k-XOR instance ψ is *well-spread*, which allows us to apply Lemma VI.5.

Same as Lemma II.6, the following lemma is a purely graph-theoretic statement.

Lemma VI.7 (Relative spectral approximation with correlated subsamples)**.** *Suppose $G = (V, E)$ is an n-vertex graph with minimum degree $d_{\min}$ (self-loops and parallel edges allowed) and spectral gap $\lambda_2(\widetilde{L}_G) = \lambda > 0$. Let $m_1, m_2 \in \mathbb{N}$, $\eta \in [0, 1/2)$, and let $\xi_1^{(1)}, \dots, \xi_{m_1}^{(1)}, \xi_1^{(2)}, \dots, \xi_{m_2}^{(2)}$ be i.i.d. random variables that take value -1 with probability η and $+1$ otherwise. Suppose there is an injective map that maps each edge $e \mapsto (c_1(e), c_2(e)) \in [m_1] \times [m_2]$, and for each $i \in [m_1]$ (resp. $j \in [m_2]$) define $G_i^{(1)}$ (resp. $G_j^{(2)}$) be the subgraph of G with edge set $\{e \in E : c_1(e) = i\}$ (resp. $\{e \in E : c_2(e) = j\}$). Moreover, suppose $G_i^{(1)}$ and $G_j^{(2)}$ have maximum degree $\leq \Delta$ for all $i \in [m_1]$, $j \in [m_2]$.*

Let H be the subgraph of G with edge set $\{e \in E : \xi_{c_1(e)}^{(1)} \xi_{c_2(e)}^{(2)} = -1\}$. There is a universal constant $B > 0$ such that if $d_{\min}\lambda \geq B\Delta \log n$, then with probability $1 - O(n^{-2})$,

$$L_H \preceq \max\left((1+\delta) \cdot 2\eta(1-\eta),\ \frac{1}{3}\right) \cdot L_G$$

for $\delta = \sqrt{\frac{B\Delta \log n}{d_{\min}\lambda}}$.

Let $\gamma := 1 - 2\eta > 0$ since $\eta < \frac{1}{2}$. Notice that $2\eta(1-\eta) = \frac{1}{2}(1-\gamma^2)$, which approaches $\frac{1}{2}$ as $\eta \to \frac{1}{2}$. Thus, if $\delta \leq \gamma^2$, then $(1+\delta) \cdot 2\eta(1-\eta) \leq (1+\gamma^2) \cdot \frac{1}{2}(1-\gamma^2) < \frac{1}{2}$, and $L_H \prec \frac{1}{2}L_G$ suffices to conclude via Lemma II.5 that the SDP relaxation on the expanding subinstance is rank 1 and recovers the planted assignment, which also gives us the set of corrupted constraints.

Proof of Lemma VI.7. First, note that by the definition of Laplacian and the spectral gap of L_G, $\mathrm{span}(\vec{1})$ is exactly the null space of L_G and is contained in the null space of L_H. Therefore, recalling that $L_G = D_G^{1/2} \widetilde{L}_G D_G^{1/2}$, it suffices to prove that

$$\left\| (\widetilde{L}_G^{\dagger})^{1/2} D_G^{-1/2} L_H D_G^{-1/2} (\widetilde{L}_G^{\dagger})^{1/2} \right\|_2 \leq \max\left((1+\delta) \cdot 2\eta(1-\eta),\ \frac{1}{3}\right). \tag{1}$$

Here $\widetilde{L}_G^{\dagger}$ is the pseudo-inverse of $\widetilde{L}_G$, and $\|\widetilde{L}_G^{\dagger}\|_2 \leq 1/\lambda$. For simplicity, for any graph G', we will write $\widehat{L}_{G'} := (\widetilde{L}_G^{\dagger})^{1/2} D_G^{-1/2} L_{G'} D_G^{-1/2} (\widetilde{L}_G^{\dagger})^{1/2}$. Thus,

$$\widehat{L}_H = \sum_{e \in E} \mathbf{1}\left(\xi_{c_1(e)}^{(1)} \xi_{c_2(e)}^{(2)} = -1\right) \cdot \widehat{L}_e\,,$$
$$\text{and } \mathbb{E}[\widehat{L}_H] = 2\eta(1-\eta) \sum_{e \in E} \widehat{L}_e\,.$$

Note that $\sum_{e \in E} \widehat{L}_e = \widehat{L}_G$, a projection matrix, thus $\left\| \sum_{e \in E} \widehat{L}_e \right\|_2 = 1$.

For each $i \in [m_1]$, we further define $G_{i,+}^{(1)}$ and $G_{i,-}^{(1)}$ to be (random) edge-disjoint subgraphs of $G_i^{(1)}$ where $G_{i,+}^{(1)}$ has edge set $\{e \in E : c_1(e) = i, \xi_{c_2(e)}^{(2)} = +1\}$ and $G_{i,-}^{(1)}$ has edge set $\{e \in E : c_1(e) = i, \xi_{c_2(e)}^{(2)} = -1\}$. Note that $G_{i,+}^{(1)}$, $G_{i,-}^{(1)}$ are independent of $\xi^{(1)} = (\xi_1^{(1)}, \dots, \xi_{m_1}^{(1)})$. By the maximum degree bound on $G_i^{(1)}$, we have that $\big\|L_{G_{i,+}^{(1)}}\big\|_2$ and $\big\|L_{G_{i,-}^{(1)}}\big\|_2 \le \big\|L_{G_i^{(1)}}\big\|_2 \le 2\Delta$. Thus,

$$\begin{aligned}\Big\|\widehat{L}_{G_{i,+}^{(1)}}\Big\|_2, \Big\|\widehat{L}_{G_{i,-}^{(1)}}\Big\|_2 &\le \Big\|\widehat{L}_{G_i^{(1)}}\Big\|_2 \\ &\le 2\Delta \cdot \Big\|\widetilde{L}_G^{\dagger}\Big\|_2 \cdot \big\|D_G^{-1}\big\|_2 \\ &\le \frac{2\Delta}{d_{\min}\lambda} .\end{aligned} \tag{2}$$

Similarly, for $j \in [m_2]$, $G_{j,+}^{(2)}$ and $G_{j,-}^{(2)}$ are (random) edge-disjoint subgraphs of $G_j^{(2)}$ independent of $\xi^{(2)} = (\xi_1^{(2)}, \dots, \xi_{m_2}^{(2)})$ such that $\big\|\widehat{L}_{G_{j,+}^{(2)}}\big\|_2$ and $\big\|\widehat{L}_{G_{j,-}^{(2)}}\big\|_2 \le \frac{2\Delta}{d_{\min}\lambda}$.

Now, we first fix $\xi^{(2)} \in \{-1,1\}^{m_2}$. Observe that we can write $\widehat{L}_H$ as

$$\widehat{L}_H = \sum_{i\in[m_1]} \mathbf{1}(\xi_i^{(1)} = +1)\cdot\widehat{L}_{G_{i,-}^{(1)}} + \mathbf{1}(\xi_i^{(1)} = -1)\cdot\widehat{L}_{G_{i,+}^{(1)}} , \tag{3}$$

and

$$\begin{aligned}&\mathbb{E}[\widehat{L}_H|\xi^{(2)}] \\ &= (1-\eta)\sum_{i\in[m_1]}\widehat{L}_{G_{i,-}^{(1)}} + \eta\sum_{i\in[m_1]}\widehat{L}_{G_{i,+}^{(1)}} \\ &= \sum_{e\in E}\Big((1-\eta)\cdot\mathbf{1}(\xi_{c_2(e)}^{(2)} = -1) + \eta\cdot\mathbf{1}(\xi_{c_2(e)}^{(2)} = +1)\Big)\cdot\widehat{L}_e \\ &\coloneqq \sum_{e\in E} w_{c_2(e)}\cdot\widehat{L}_e .\end{aligned} \tag{4}$$

Here $w_{c_2(e)} \in \{\eta, 1-\eta\}$, thus $\big\|\mathbb{E}[\widehat{L}_H|\xi^{(2)}]\big\|_2 \ge \eta\big\|\sum_{e\in E}\widehat{L}_e\big\|_2 = \eta$.

We now split the analysis into two cases. Let $\eta_0 \coloneqq 1/12$.

Case 1: $\boldsymbol{\eta \ge \eta_0}$.

In light of Eq. (3), we define $X_i \coloneqq \mathbf{1}(\xi_i^{(1)} = +1)\cdot\widehat{L}_{G_{i,-}^{(1)}} + \mathbf{1}(\xi_i^{(1)} = -1)\cdot\widehat{L}_{G_{i,+}^{(1)}}$ such that $\widehat{L}_H = \sum_{i\in[m_1]} X_i$. Moreover, we have that $X_i \succeq 0$ and $\|X\|_2 \le \frac{2\Delta}{d_{\min}\lambda}$ almost surely from Eq. (2). Thus, applying matrix Chernoff (Fact III.3), we get

$$\begin{aligned}&\Pr_{\xi^{(1)}}\Big[\Big\|\widehat{L}_H\Big\|_2 \ge (1+\delta)\Big\|\mathbb{E}[\widehat{L}_H|\xi^{(2)}]\Big\|_2\Big] \\ &\le n\cdot\exp\left(-\frac{1}{3}\delta^2\Big\|\mathbb{E}[\widehat{L}_H|\xi^{(2)}]\Big\|_2\cdot\frac{d_{\min}\lambda}{2\Delta}\right) \\ &\le n\cdot\exp\left(-\frac{\delta^2\eta d_{\min}\lambda}{6\Delta}\right) ,\end{aligned} \tag{5}$$

which is at most $O(n^{-2})$ as long as $\delta^2 \ge \frac{B_1\Delta\log n}{d_{\min}\lambda}$ for a large enough constant B_1.

Next, we similarly prove concentration for $\big\|\mathbb{E}[\widehat{L}_H|\xi^{(2)}]\big\|_2$ over $\xi^{(2)}$. Recalling Eq. (4),

$$\begin{aligned}\mathbb{E}[\widehat{L}_H|\xi^{(2)}] &= \sum_{e\in E} w_{c_2(e)}\cdot\widehat{L}_e \\ &= \sum_{j\in[m_2]} w_j \sum_{e\in G_j^{(2)}}\widehat{L}_e = \sum_{j\in[m_2]} w_j\cdot\widehat{L}_{G_j^{(2)}} .\end{aligned}$$

$\mathbb{E}[w_j] = 2\eta(1-\eta)$, and $\big\|\mathbb{E}_{\xi^{(2)}}\mathbb{E}[\widehat{L}_H|\xi^{(2)}]\big\|_2 = 2\eta(1-\eta)\big\|\sum_{e\in E}\widehat{L}_e\big\|_2 = 2\eta(1-\eta)$. Since $\big\|w_j\widehat{L}_{G_j^{(2)}}\big\|_2 \le \frac{2(1-\eta)\Delta}{d_{\min}\lambda}$, we can apply matrix Chernoff again:

$$\begin{aligned}&\Pr_{\xi^{(2)}}\Big[\Big\|\mathbb{E}[\widehat{L}_H|\xi^{(2)}]\Big\|_2 \ge (1+\delta')\cdot 2\eta(1-\eta)\Big] \\ &\le n\cdot\exp\left(-\frac{1}{3}\delta'^2\cdot 2\eta(1-\eta)\cdot\frac{d_{\min}\lambda}{2(1-\eta)\Delta}\right)\end{aligned} \tag{6}$$

which is at most $O(n^{-2})$ as long as $\delta'^2 \ge \frac{B_2\Delta\log n}{d_{\min}\lambda}$ for a large enough constant B_2. Combining both tail bounds, by the union bound, we have that with probability at least $1 - O(n^{-2})$, $\big\|\widehat{L}_H\big\|_2 \le (1+\delta)\cdot 2\eta(1-\eta)$ as long as $\delta^2 \ge \frac{B\Delta\log n}{d_{\min}\lambda}$ for a large enough B. This establishes Eq. (1), proving the lemma for this case.

Case 2: $\boldsymbol{\eta < \eta_0}$. To handle this case, observe that the exact same analysis goes through for $\widetilde{H} = \{e \in E : \xi_{c_1(e)}^{(1)} = -1 \text{ or } \xi_{c_2(e)}^{(2)} = -1\} \supseteq H$. Indeed, similar to Eq. (3) and (4), we have $\widehat{L}_{\widetilde{H}} = \sum_{i\in[m_1]}\widetilde{X}_i$ where $\widetilde{X}_i = \mathbf{1}(\xi_i^{(1)} = +1)\cdot\widehat{L}_{G_{i,-}^{(1)}} + \mathbf{1}(\xi_i^{(1)} = -1)\cdot\widehat{L}_{G_i^{(1)}}$ (notice the 2nd term is $G_i^{(1)}$ instead of $G_{i,+}^{(1)}$), and

$$\begin{aligned}\mathbb{E}[\widehat{L}_{\widetilde{H}}|\xi^{(2)}] &= (1-\eta)\sum_{i\in[m_1]}\widehat{L}_{G_{i,-}^{(1)}} + \eta\sum_{i\in[m_1]}\widehat{L}_{G_i^{(1)}} \\ &= \sum_{e\in E}\widetilde{w}_{c_2(e)}\cdot\widehat{L}_e = \sum_{j\in[m_2]}\widetilde{w}_j\cdot\widehat{L}_{G_j^{(2)}} ,\end{aligned}$$

where $\widetilde{w}_j = 1$ if $\xi_j^{(2)} = -1$ and η if $\xi_j^{(2)} = +1$, hence $\mathbb{E}[\widetilde{w}_j] = \eta + \eta(1-\eta) = \eta(2-\eta)$. Moreover, $\big\|\mathbb{E}_{\xi^{(2)}}\mathbb{E}[\widehat{L}_{\widetilde{H}}|\xi^{(2)}]\big\|_2 = \eta(2-\eta)\big\|\sum_{e\in E}\widehat{L}_e\big\|_2 = \eta(2-\eta)$.

First, set $\eta = \eta_0$, and let $\widetilde{H}_0$ be the random subgraph as defined above. Similar to Eq. (5) and (6), we apply matrix Chernoff (Fact III.3) and get that with probability $1-O(n^{-2})$, $\big\|\widehat{L}_{\widetilde{H}_0}\big\|_2 \le (1+\delta)\cdot\eta_0(2-\eta_0)$ for $\delta = \sqrt{\frac{B\Delta\log n}{d_{\min}\lambda}} \le 1$. In particular, this means that $L_{\widetilde{H}_0} \preceq 2\eta_0(2-\eta_0)L_G \preceq \frac{1}{3}L_G$ when $\eta_0 = 1/12$.

Now, fix any $\eta < \eta_0$. We can obtain a coupling between this case and the case when $\eta = \eta_0$ by randomly changing $\xi_i^{(1)}$ and $\xi_j^{(2)}$ from $+1$ to -1 (while not flipping the ones with -1). Notice that $\widetilde{H}$ is monotone increasing as we change any $+1$ to -1 (whereas H is not!), thus we must have $\widetilde{H} \subseteq \widetilde{H}_0$ in this coupling. Then, as $H \subseteq \widetilde{H}$, we have

$$L_H \preceq L_{\widetilde{H}} \preceq L_{\widetilde{H}_0} \preceq \frac{1}{3}L_G$$

with probability $1 - O(n^{-2})$. This finishes the proof of Lemma VI.7. □

E. Recovery of corrupted constraints from corrupted pairs

We have thus shown that, with probability $\ge 1 - 1/\mathrm{poly}(n)$, we can *exactly* recover the set of corrupted constraints within each expanding subinstance $\phi_1, \dots, \phi_T$. Recall that after pruning and expander decomposition (Lemma VI.6), the expanding subinstances contain a $(1-\varepsilon)$-fraction of all edges in the instance ϕ, and the set of edges removed only depends on the constraint graph and not the right-hand sides of ϕ. As stated in Observation VI.2, the instance ϕ has exactly $m^2/4p$ edges, and they correspond exactly to the set $\{(u, C, C') : u \in [p], C \in \mathcal{H}_u^{(L)}, C' \in \mathcal{H}_u^{(R)}\}$, and moreover an edge e in ϕ is corrupted if and only if exactly one of the two constraints $(u, C), (u, C')$ is corrupted in the original instance ψ, where e corresponds to (u, C, C'). For each $u \in [p]$ and $C \in \mathcal{H}_u = \mathcal{H}_u^{(L)} \cup \mathcal{H}_u^{(R)}$, let $\xi_u(C) = -1$ if (u, C) is corrupted in ψ, and 1 otherwise. It thus follows that we have learned, for $1 - \varepsilon$ fraction of all $\{(u, C, C') : u \in [p], C \in \mathcal{H}_u^{(L)}, C' \in \mathcal{H}_u^{(R)}\}$, the product $\xi_u(C) \cdot \xi_u(C')$.

It now remains to show how to recover $\xi_u(C)$ for most $u \in [p]$, $C \in \mathcal{H}_u$. For each $u \in [p]$, let $P_u \subseteq \{(C, C') : C \in \mathcal{H}_u^{(L)}, C' \in \mathcal{H}_u^{(R)}\}$ such that we have determined $\xi_u(C) \cdot \xi_u(C')$, and let $P = \cup_{u \in [p]} P_u$. We know that $|P| \geq (1 - \varepsilon)\frac{m^2}{4p}$. Let ε_u be chosen so that $|P_u| = (1 - \varepsilon_u)\frac{m^2}{4p^2}$, i.e., ε_u is the fraction of pairs in $\mathcal{H}_u^{(L)} \times \mathcal{H}_u^{(R)}$ that were deleted in Lemma VI.6. Notice that we have

$$\begin{aligned}(1 - \varepsilon)\frac{m^2}{4p} \leq |P| = \sum_{u \in [p]} |P_u| = \frac{m^2}{4p^2} \sum_{u \in [p]} (1 - \varepsilon_u) \\ \implies \frac{1}{p} \sum_{u \in [p]} \varepsilon_u \leq \varepsilon\,. \end{aligned} \tag{7}$$

One can think of this problem as a collection of disjoint *satisfiable* (noiseless) 2-XOR instances on P_u, where each P_u is a biclique ($\frac{m}{2p}$ vertices on each side) with ε_u fraction of edges are removed.

Algorithm VI.8 (Recover corrupted constraints from corrupted pairs)**.**

```
Given: For each $u \in [p]$, a set $P_u \subseteq \mathcal{H}_u^{(L)} \times \mathcal{H}_u^{(R)}$ such
       that $|P_u| = (1 - \varepsilon_u)\frac{m^2}{4p^2}$ for $\varepsilon_u \in [0, 1]$, along
       with "right-hand sides" $\xi_u(C) \cdot \xi_u(C')$ for each
       $(C, C') \in P_u$.
Output: For each $u \in [p]$, disjoint subsets $\mathcal{A}_u^{(1)}, \mathcal{A}_u^{(2)} \subseteq$
        $\mathcal{H}_u$.
Operation:
    1) Initialize: $\mathcal{A}_u^{(1)}, \mathcal{A}_u^{(2)} = \emptyset$ for each $u \in [p]$.
    2) For each $u \in [p]$:
        a) If $\varepsilon_u \geq 1/3$, set $\mathcal{A}_u^{(1)} = \mathcal{H}_u$ and $\mathcal{A}_u^{(2)} =$
           $\emptyset$.
        b) Else if $\varepsilon_u < 1/3$, let $G_u$ be the graph with
           vertex set $\mathcal{H}_u = \mathcal{H}_u^{(L)} \cup \mathcal{H}_u^{(R)}$ with edges
           given by $P_u$, and let $S_u$ be the size of the
           largest connected component in $G_u$.
        c) As $S_u$ is connected in $G_u$, and we know
           $\xi_u(C)\xi_u(C')$ for each edge $(C, C')$ in
           $G_u$, by solving a linear system of equa-
           tions we obtain $z \in \{-1, 1\}^{\mathcal{H}_u}$ such that
           either $z_C = \xi_u(C)$ for all $C \in S_u$, or
           $z_C = -\xi_u(C)$ for all $C \in S_u$. That is,
           $z_C = \xi_u(C)$ up to a global sign.
        d) Pick the global sign to minimize the num-
           ber of $C \in S_u$ for which $z_C = -1$. Set
           $\mathcal{A}_u^{(1)} = \mathcal{H}_u \setminus S_u$ and $\mathcal{A}_u^{(2)} = \{C \in S_u :$
           $z_C = -1\}$.
    3) Output $\{\mathcal{A}_u^{(1)}\}_{u \in [p]}$, $\{\mathcal{A}_u^{(2)}\}_{u \in [p]}$.
```

We now analyze Algorithm VI.8 via the following lemma.

Lemma VI.9. *Let $\eta \in [0, 1/2)$, and let $|\mathcal{H}_u| = \frac{m}{p} \geq \frac{24k}{(1-2\eta)^2} \log n$ and $|P_u| = (1 - \varepsilon_u)\frac{m^2}{4p^2}$ with $\varepsilon_u \in [0, 1]$ for each $u \in [p]$, and $\frac{1}{p}\sum_{u \in [p]} \varepsilon_u \leq \varepsilon$. The outputs of Algorithm VI.8 satisfy the following: (1) $\sum_{u \in [p]} |\mathcal{A}_u^{(1)}| \leq 4\varepsilon m$, and (2) with probability $1 - n^{-k}$ over the noise $\{\xi_u(C)\}_{u \in [p], C \in \mathcal{H}_u}$, for every $u \in [p]$ we have that $\mathcal{A}_u^{(2)} = \{C \in \mathcal{H}_u : \xi_u(C) = -1\} \setminus \mathcal{A}_u^{(1)}$.*

Proof. Suppose that $\varepsilon_u < 1/3$. Observe that G_u is a graph obtained by taking a biclique with left vertices $\mathcal{H}_u^{(L)}$ and right vertices $\mathcal{H}_u^{(R)}$, i.e., with $m/2p$ left vertices and $m/2p$ right vertices. The following lemma shows that the largest connected component S_u in G_u has size at least $\frac{m}{p}(1 - \varepsilon_u)$.

Claim VI.10. Let $K_{n,n}$ be the complete bipartite graph with n left vertices L and n right vertices R. Let G be a graph obtained by deleting εn^2 edges from $K_{n,n}$. Then, the largest connected component in G has size $\geq 2n(1 - \varepsilon)$.

We postpone the proof of Claim VI.10 to the end of the section, and continue with the proof of Lemma VI.9.

We now argue that we can efficiently obtain the vector z in Step (2c) of Algorithm VI.8. Indeed, this is done as follows. First, pick one $C_0 \in S_u$ arbitrarily, and set $z_{C_0} = 1$. Then, we propagate in a breadth-first search manner: for any edge (C, C') in S_u where z_C is determined, set $z_{C'} = z_C \cdot \xi_u(C)\xi_u(C')$. We repeat this process until we have labeled all of S_u. Notice that as S_u is a connected component, fixing z_{C_0} for any $C_0 \in S_u$ uniquely determines the assignment of all S_u, thus we have obtained $z_C = \xi_u(C)$ up to a global sign.

Now, we observe that S_u does not depend on the noise in ψ. Indeed, this is because the pruning and expander decomposition (and thus the graph G_u) depends solely on the constraint graph G of the instance ϕ, and not on the right-hand sides of the constraints. The following lemma thus shows that with high probability over the noise, the number of $C \in S_u$ where $\xi_u(C) = -1$ is strictly less than $1/2|S_u|$. Hence, in Step (2d), by picking the assignment $\pm z$ that minimizes the number of $C \in S_u$ with $\xi_u(C) = -1$, we see that $\mathcal{A}_u^{(2)} = \{C \in S_u : z_C = -1\} = \{C \in S_u : \xi_u(C) = -1\}$.

Claim VI.11. Let $\eta \in (0, 1/2)$ be the corruption probability, and assume that $p \leq n^k$ and $\frac{m}{p} \geq \frac{24k}{(1-2\eta)^2} \log n$. With probability $1 - n^{-k}$ over the noise in ψ, it holds that for each $u \in [p]$ with $\varepsilon_u < 1/3$, $|\{C \in S_u : \xi_u(C) = -1\}| < \frac{1}{2}|S_u|$.

We postpone the proof of Claim VI.11, and finish the proof of Lemma VI.9. We next bound $\sum_{u \in [p]} |\mathcal{A}_u^{(1)}|$. By Eq. (7) we have that $\frac{1}{p}\sum_u \varepsilon_u \leq \varepsilon$. Thus,

$$\sum_{u:\varepsilon_u \geq 1/3} |\mathcal{H}_u| \leq \frac{m}{p} \sum_{u:\varepsilon_u \geq 1/3} 3\varepsilon_u \leq 3\varepsilon m\,.$$

Moreover, by Claim VI.10 we have $|S_u| \geq (1-\varepsilon_u)|\mathcal{H}_u| = (1-\varepsilon_u)\frac{m}{p}$. Thus,

$$\sum_{u:\varepsilon_u < 1/3} |\mathcal{H}_u \setminus S_u| \leq \sum_{u:\varepsilon_u < 1/3} \varepsilon_u \cdot \frac{m}{p} \leq \varepsilon m\,.$$

Therefore, combining the two,

$$\sum_{u\in[p]} |\mathcal{A}_u^{(1)}| = \sum_{u:\varepsilon_u < 1/3} |\mathcal{H}_u \setminus S_u| + \sum_{u:\varepsilon_u \geq 1/3} |\mathcal{H}_u| \leq 4\varepsilon m\,,$$

which finishes the proof of Lemma VI.9. □

In the following, we prove Claims VI.10 and VI.11.

Proof of Claim VI.10. Let $S_1, \dots, S_t$ be the connected components of G. Let $\ell_i = |S_i \cap L|$ and $r_i = |S_i \cap R|$. The number of edges in G is at most $\sum_{i=1}^t \ell_i r_i$.

Now, suppose that the largest connected component of G has size at most M. Then, we have that $\ell_i + r_i \leq M$ for all $i \in [t]$. Notice that the number of edges deleted from $K_{n,n}$ to produce G must be at least $n^2 - \sum_{i=1}^t \ell_i r_i$, and this is at most εn^2. Hence, by maximizing the quantity $\sum_{i=1}^t \ell_i r_i$ subject to $\ell_i + r_i \leq M$ for all $i \in [t]$ and $\sum_{i=1}^t \ell_i + r_i = 2n$, we can obtain a lower bound on the number of edges deleted from $K_{n,n}$ in order for the largest connected component of G to have size at most M. We have that

$$\sum_{i=1}^t \ell_i r_i \leq \sum_{i=1}^t \left(\frac{\ell_i + r_i}{2}\right)^2 \leq \frac{M}{2} \cdot \sum_{i=1}^t \frac{\ell_i + r_i}{2} = \frac{nM}{2}\,,$$

where the first inequality is by the AM-GM inequality. Thus,

$$\varepsilon n^2 \geq n^2 - \frac{nM}{2} \implies M \geq 2n(1-\varepsilon)\,,$$

which finishes the proof. □

Proof of Claim VI.11. Let u be such that $\varepsilon_u < 1/3$, and let S_u be the largest connected component in G_u. Observe that S_u is determined solely by the constraint graph of ϕ, and in particular does not depend on the noise in ϕ (and hence on the noise in ψ). As $p \leq n^k$ by assumption, it thus suffices to show that for each $u \in [p]$, with probability $1 - n^{-2k}$ it holds that $|\{C \in S_u : \xi_u(C) = -1\}| < \frac{1}{2}|S_u|$. Notice that $|\{C \in S_u : \xi_u(C) = -1\}|$ is simply the sum of $|S_u|$ Bernoulli(η) random variables. By Hoeffding's inequality, with probability $\geq 1 - \exp(-2\delta^2|S_u|)$ it holds that $|\{C \in S_u : \xi_u(C) = -1\}| \leq (\eta+\delta)|S_u|$. We choose $\delta = \frac{1}{2}(\frac{1}{2} - \eta)$ such that $\eta + \delta < \frac{1}{2}$ for $\eta \in (0, \frac{1}{2})$. Then, by noting that $2\delta^2|S_u| \geq 2\delta^2(1-\varepsilon_u)|\mathcal{H}_u| \geq \frac{1}{2}(\frac{1}{2}-\eta)^2 \cdot \frac{2}{3} \cdot \frac{m}{p} \geq 2k\log n$ since $\frac{m}{p} \geq \frac{24k}{(1-2\eta)^2}\log n$, Claim VI.11 follows. □

F. Finishing the proof of Lemma V.2

Proof of Lemma V.2. We are given an τ-spread p-bipartite k-XOR instance ψ with constraint graph $\mathcal{H} = \{\mathcal{H}_u\}_{u\in[p]}$, where we recall from Definition V.1 that (1) $m = |\mathcal{H}|$ and each $|\mathcal{H}_u| = \frac{m}{p} \geq 2\lfloor\frac{1}{2\tau^2}\rfloor$ and $\frac{m}{p}$ is even, and (2) for any $Q \subseteq [n]$, $\deg_u(Q) \leq \frac{1}{\tau^2}\max(1, n^{\frac{k}{2}-1-|Q|})$. For convenience, let $m \geq n^{\frac{k-1}{2}}\sqrt{p}\cdot\beta$ where $\beta := C \cdot \frac{(k\log n)^{3/2}}{\tau\gamma^2\varepsilon^{3/2}}$ and $\gamma := 1 - 2\eta \in (0,1]$ since $\eta \in [0, \frac{1}{2})$.

First, we construct the 2-XOR instance ϕ defined in Definition VI.1. As stated in Observation VI.2, the average degree is at least $d := \frac{1}{4}\beta^2$, and furthermore, by Lemma VI.5, the maximum degree of $G_{u,C}^{(L)}(\phi)$ and $G_{u,C'}^{(R)}(\phi)$ for any $u \in [p]$, $C \in \mathcal{H}_u^{(L)}$ and $C' \in \mathcal{H}_u^{(R)}$ is bounded by $\Delta := 1/\tau^2$. The algorithm then follows the steps outlined in Section VI-B.

Step 1. We apply graph pruning and expander decomposition (Lemma VI.6) with parameter $\varepsilon' := \frac{1}{4}\varepsilon$, which decomposes ϕ into $\phi_1, \dots, \phi_T$ such that they contain $1 - \varepsilon'$ fraction of the constraints in ϕ, and their constraint graphs (after adding some self-loops due to expander decomposition) have minimum degree $d_{\min} \geq \frac{1}{3}\varepsilon' d = \frac{1}{48}\varepsilon\beta^2$ and spectral gap $\lambda \geq \Omega(\varepsilon'^2/\log^2 m) = \Omega(\varepsilon^2/(k^2\log^2 n))$.

Step 2. We solve the SDP relaxation for each subinstance ϕ_i. Let G be the constraint graph of ϕ_i (with at most $N \leq n^{k-1}$ vertices) and H be the corrupted edges of G. We apply the relative spectral approximation result (Lemma VI.7) with $\xi_1^{(1)}, \dots, \xi_{m/2p}^{(1)}$ (resp. $\xi_1^{(2)}, \dots, \xi_{m/2p}^{(2)}$) being $\{-1, 1\}$ random variables indicating whether each $C \in \mathcal{H}_u^{(L)}$ (resp. $C' \in \mathcal{H}_u^{(R)}$) is corrupted. Moreover, the subgraphs $G_i^{(1)}$ and $G_j^{(2)}$ in Lemma VI.7 (which are simply subgraphs of $G_{u,C}^{(L)}(\phi)$ and $G_{u,C'}^{(R)}(\phi)$) have maximum degree $\leq \Delta = 1/\tau^2$. Thus, we have that with probability $1 - O(N^{-2})$,

$$L_H \preceq \max\left((1+\delta)\cdot 2\eta(1-\eta),\ \frac{1}{3}\right) \cdot L_G$$

where $\delta = \sqrt{\frac{B\Delta\log N}{d_{\min}\lambda}} \leq O\Big(\sqrt{\frac{k^3\log^3 n}{\tau^2\varepsilon^3\beta^2}}\Big)$. Plugging in β (for large enough C), we get that $\delta \leq \gamma^2 = 1 - 4\eta(1-\eta)$. Therefore, we have $(1+\delta)\cdot 2\eta(1-\eta) \leq (1+\gamma^2)\cdot\frac{1}{2}(1-\gamma^2) < \frac{1}{2}$, hence $L_H \prec \frac{1}{2}L_G$. By union bound over all $T \leq N$ subinstances, this holds for all subinstances ϕ_i with probability $1 - \frac{1}{\mathrm{poly}(n)}$ over the randomness of the noise.

Then, by Lemma II.5, the SDP relaxation has a unique optimum which is the planted assignment. Thus, we can identify the set of corrupted edges in each ϕ_i.

Step 3. So far we have identified, for $\geq 1 - \varepsilon'$ fraction of all $\{(u, C, C') : u \in [p], C \in \mathcal{H}_u^{(L)}, C' \in \mathcal{H}_u^{(R)}\}$, the product $\xi_u(C)\cdot\xi_u(C')$, where $\xi_u(C) = -1$ if (u, C) is corrupted in ψ, and $+1$ otherwise. Let $P_u \subseteq \{(C, C') : C \in \mathcal{H}_u^{(L)}, C' \in \mathcal{H}_u^{(R)}\}$ be such pairs for each $u \in [p]$, and let $P = \cup_{u\in[p]} P_u$. Note that $|P| \geq (1-\varepsilon')\frac{m^2}{4p}$ and P depends only on $\mathcal{H}$ and not on the noise.

We then run Algorithm VI.8. By the assumption that $\tau \leq \frac{c\gamma}{\sqrt{k\log n}}$ for a small enough c, we have $|\mathcal{H}_u| = \frac{m}{p} \geq 2\lfloor\frac{1}{2\tau^2}\rfloor \geq$

$\frac{24k}{(1-2\eta)^2}$, which is the condition we need in Lemma VI.9. Thus, with probability $1-n^{-k}$, Algorithm VI.8 outputs (1) $\mathcal{A}_1 \subseteq \mathcal{H}$ which only depends on $\mathcal{H}$ and such that $|\mathcal{A}_1| \leq 4\varepsilon' m = \varepsilon m$, and (2) $\mathcal{A}_2 \subseteq \mathcal{H}$, the set of corrupted constraints in $\mathcal{H} \setminus \mathcal{A}_1$. This completes the proof of Lemma V.2. □

References

[1] Jackson Abascal, Venkatesan Guruswami, and Pravesh K. Kothari. Strongly refuting all semi-random Boolean CSPs. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, pages 454–472. SIAM, 2021.

[2] Emmanuel Abbe and Colin Sandon. Detection in the stochastic block model with multiple clusters: proof of the achievability conjectures, acyclic BP, and the information-computation gap. *arXiv preprint arXiv:1512.09080*, 2015.

[3] Dimitris Achlioptas, Arthur Chtcherba, Gabriel Istrate, and Cristopher Moore. The phase transition in 1-in-k SAT and NAE 3-SAT. In *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, pages 721–722, 2001.

[4] Sarah R. Allen, Ryan O'Donnell, and David Witmer. How to Refute a Random CSP. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 689–708. IEEE Computer Society, 2015.

[5] Gunnar Andersson and Lars Engebretsen. Better approximation algorithms for Set splitting and Not-All-Equal SAT. *Information Processing Letters*, 65(6):305–311, 1998.

[6] Benny Applebaum. Cryptographic Hardness of Random Local Functions: Survey. *Computational complexity*, 25:667–722, 2016.

[7] Sanjeev Arora, David R. Karger, and Marek Karpinski. Polynomial time approximation schemes for dense instances of *NP*-hard problems. In *Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing, 29 May-1 June 1995, Las Vegas, Nevada, USA*, pages 284–293. ACM, 1995.

[8] Boaz Barak, Siu On Chan, and Pravesh K. Kothari. Sum of Squares Lower Bounds from Pairwise Independence. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 97–106. ACM, 2015.

[9] Boaz Barak and Ankur Moitra. Noisy Tensor Completion via the Sum-of-Squares Hierarchy. In *Proceedings of the 29th Conference on Learning Theory, COLT 2016, New York, USA, June 23-26, 2016*, volume 49 of *JMLR Workshop and Conference Proceedings*, pages 417–445. JMLR.org, 2016.

[10] Boaz Barak and David Steurer. Sum-of-squares proofs and the quest toward optimal algorithms. *CoRR*, abs/1404.5236, 2014.

[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] Mihir Bellare, Shafi Goldwasser, Carsten Lund, and Alexander Russell. Efficient probabilistically checkable proofs and applications to approximations. In *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, pages 294–304, 1993.

[13] Siavosh Benabbas, Konstantinos Georgiou, Avner Magen, and Madhur Tulsiani. SDP gaps from pairwise independence. *Theory of Computing*, 8(1):269–289, 2012.

[14] Avrim Blum and Joel Spencer. Coloring Random and Semi-Random k-Colorable Graphs. *J. Algorithms*, 19(2):204–234, 1995.

[15] Andrej Bogdanov and Youming Qiao. On the security of Goldreich's one-way function. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques: 12th International Workshop, APPROX 2009*, pages 392–405. Springer, 2009.

[16] Rares-Darius Buhai, Pravesh K Kothari, and David Steurer. Algorithms approaching the threshold for semi-random planted clique. In *Proceedings of the 55th Annual ACM SIGACT Symposium on Theory of Computing*, 2022.

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

[18] Amin Coja-Oghlan, Andreas Goerdt, and André Lanka. Strong refutation heuristics for random k-SAT. *Combinatorics, Probability & Computing*, 16(1):5, 2007.

[19] Jian Ding, Allan Sly, and Nike Sun. Satisfiability threshold for random regular NAE-SAT. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 814–822, 2014.

[20] 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, 2002.

[21] Uriel Feige. Refuting Smoothed 3CNF Formulas. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007), October 20-23, 2007, Providence, RI, USA, Proceedings*, pages 407–417. IEEE Computer Society, 2007.

[22] Uriel Feige and Joe Kilian. Heuristics for semirandom graph problems. *J. Comput. Syst. Sci.*, 63(4):639–671, 2001.

[23] Vitaly Feldman, Will Perkins, and Santosh S. Vempala. Subsampled Power Iteration: a Unified Algorithm for Block Models and Planted CSP's. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2836–2844, 2015.

[24] Vitaly Feldman, Will Perkins, and Santosh S. Vempala. On the Complexity of Random Satisfiability Problems with Planted Solutions. *SIAM Journal on Computing*, 47(4):1294–1338, 2018.

[25] Noah Fleming, Pravesh Kothari, and Toniann Pitassi. Semialgebraic Proofs and Efficient Algorithm Design. *Foundations and Trends® in Theoretical Computer Science*, 14(1-2):1–221, 2019.

[26] Dimitris Fotakis, Michael Lampis, and Vangelis Th. Paschos. Subexponential Approximation Schemes for CSPs: From Dense to Almost Sparse. In *33rd Symposium on Theoretical Aspects of Computer Science, STACS 2016, February 17-20, 2016, Orléans, France*, volume 47 of *LIPIcs*, pages 37:1–37:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016.

[27] Andreas Goerdt and André Lanka. Recognizing more random unsatisfiable 3-sat instances efficiently. *Electron. Notes Discret. Math.*, 16:21–46, 2003.

[28] Oded Goldreich. Candidate One-Way Functions Based on Expander Graphs. *Electron. Colloquium Comput. Complex.*, 2000.

[29] Venkatesan Guruswami, Pravesh K. Kothari, and Peter Manohar. Algorithms and certificates for Boolean CSP refutation: smoothed is no harder than random. In *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 678–689. ACM, 2022.

[30] Johan Håstad. Some optimal inapproximability results. *Journal of the ACM (JACM)*, 48(4):798–859, 2001.

[31] Jun-Ting Hsieh, Pravesh K. Kothari, and Sidhanth Mohanty. A simple and sharper proof of the hypergraph Moore bound. In *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 2324–2344. SIAM, 2023.

[32] Russell Impagliazzo and Ramamohan Paturi. On the Complexity of k-SAT. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001.

[33] Haixia Jia, Cristopher Moore, and Doug Strain. Generating Hard Satisfiable Formulas by Hiding Solutions Deceptively. *Journal of Artificial Intelligence Research*, 28:107–118, 2007.

[34] Ravi Kannan, Santosh Vempala, and Adrian Vetta. On clusterings: Good, bad and spectral. *Journal of the ACM (JACM)*, 51(3):497–515, 2004.

[35] David R Karger. Random sampling in cut, flow, and network design problems. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pages 648–657, 1994.

[36] Pravesh K. Kothari, Ryuhei Mori, Ryan O'Donnell, and David Witmer. Sum of squares lower bounds for refuting any CSP. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 132–145. ACM, 2017.

[37] Florent Krzakala, Marc Mézard, and Lenka Zdeborová. Reweighted Belief Propagation and Quiet Planting for Random k-SAT. *Journal on Satisfiability, Boolean Modeling and Computation*, 8(3-4):149–171, 2012.

[38] Florent Krzakala and Lenka Zdeborová. Hiding Quiet Solutions in Random Constraint Satisfaction Problems. *Physical review letters*, 102(23):238701, 2009.

[39] Ryuhei Mori and David Witmer. Lower Bounds for CSP Refutation by SDP Hierarchies. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2016, September 7-9, 2016, Paris, France*, volume 60 of *LIPIcs*, pages 41:1–41:30, 2016.

[40] Dana Moshkovitz. The Projection Games Conjecture and the NP-Hardness of $\ln n$-Approximating Set-Cover. *Theory Comput.*, 11:221–235, 2015.

[41] Dana Moshkovitz and Ran Raz. Two-query PCP with subconstant error. *J. ACM*, 57(5):29:1–29:29, 2010.

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

[43] Ryan O'Donnell and David Witmer. Goldreich's PRG: evidence for near-optimal polynomial stretch. In *2014 IEEE 29th Conference on Computational Complexity (CCC)*, pages 1–12. IEEE, 2014.

[44] Prasad Raghavendra, Satish Rao, and Tselil Schramm. Strongly refuting random CSPs below the spectral threshold. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 121–131. ACM, 2017.

[45] Thatchaphol Saranurak and Di Wang. Expander decomposition and pruning: Faster, stronger, and simpler. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2616–2635. SIAM, 2019.

[46] Daniel A Spielman and Shang-Hua Teng. Spectral sparsification of graphs. *SIAM Journal on Computing*, 40(4):981–1025, 2011.

[47] Joel A Tropp. An introduction to matrix concentration inequalities. *Foundations and Trends® in Machine Learning*, 8(1-2):1–230, 2015.

[48] Alexander S. Wein, Ahmed El Alaoui, and Cristopher Moore. The Kikuchi Hierarchy and Tensor PCA. In David Zuckerman, editor, *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, pages 1446–1468. IEEE Computer Society, 2019.

[49] Christian Wulff-Nilsen. Fully-dynamic minimum spanning forest with improved worst-case update time. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1130–1143, 2017.

Appendix A
Notions of relative approximation

In this paper, we have encountered several notions of relative graph approximations. Let G be an n-vertex graph, and let H be a random subgraph of G by selecting each edge with a fixed probability $\eta \in (0,1)$. We are interested in the sufficient conditions on G for each of the following to hold with probability $1-o(1)$ (for some $\delta = o(1)$):

(1) **Relative cut approximation**: $x^\top L_H x \leq (1+\delta)\eta \cdot x^\top L_G x$ for all $x \in \{-1,1\}^n$.

(2) **Relative SDP approximation**: $\langle X, L_H\rangle \leq (1+\delta)\eta \cdot \langle X, L_G\rangle$ for all symmetric matrices $X \succeq 0$ with $\mathrm{diag}(X) = \mathbb{I}$.

(3) **Relative spectral approximation:** $L_H \preceq (1+\delta)\eta \cdot L_G$.

Here, we only state one-sided inequalities, as solving noisy XOR requires only an upper bound on L_H. Note also that the above is in increasing order: relative spectral approximation implies relative SDP approximation, which in turn implies relative cut approximation.

Recall from Lemma II.4 that a lower bound on the min-cut of G suffices for cut approximation to hold, while Lemma II.6 shows that lower bounds on the minimum degree and spectral gap of G suffice for spectral approximation to hold. It is natural to wonder whether a min-cut lower bound is sufficient for SDP approximation as well, since it allows us to efficiently recover the planted assignment in a noisy planted 2-XOR via solving an SDP relaxation (see Lemma II.5). Unfortunately, there is a counterexample.

Separation of cut and SDP approximation. The example is the same graph that separates cut and spectral approximation described in [46]. Let n be even and $k = k(n)$. Define $G = (V,E)$ be a graph on $N = nk$ vertices where $V = \{0,1,\dots,n-1\} \times \{1,\dots,k\}$ and $(u,i), (v,j) \in V$ are connected if $v = u \pm 1 \mod n$. Moreover, there is one additional edge e^* between $(0,1)$ and $(n/2,1)$. In other words, G consists of n clusters of vertices of size k, where the clusters form a ring with a complete bipartite graph between adjacent clusters, along with a special edge e^* in the middle.

Clearly, the minimum cut of G is $2k$, which means that cut approximation holds. Essentially, the special edge e^* does not play a role here.

However, we will show that e^* breaks SDP approximation. Define vector $x_0 \in \mathbb{R}^V$ such that the (u,i) entry is

$$x_0(u,i) = \min(u, n-u)\,,$$

and vectors $x_1,\dots,x_{n-1}$ to be cyclic shifts of x_0: for $w \in \{0,1,\dots,n-1\}$,

$$x_w(u,i) = x_0(u-w \ (\mathrm{mod}\ n),\ i)\,.$$

We note that x_0 is the vector shown in [46] that breaks spectral approximation. We now show that $X = \sum_{w=0}^{n-1} x_w x_w^\top$ (scaled so that X has all 1s on the diagonal) breaks SDP approximation.

First, it is easy to see that the diagonal entries of X are all equal due to symmetry. Thus, for some scaling c, $cX \succeq 0$ and $\mathrm{diag}(cX) = \mathbb{I}$.

Observe that for $w \leq \frac{n}{2}-1$, $x_w(0,1) = w$ and $x_w(\frac{n}{2},1) = \frac{n}{2} - w$. For $w \geq \frac{n}{2}$, $x_w(0,1) = n-w$ and $x_w(\frac{n}{2},1) = w - \frac{n}{2}$. Thus, as $x_w^\top L_{e^*} x_w = \left(x_w(0,1) - x_w(\frac{n}{2},1)\right)^2$,

$$\begin{aligned}\langle X, L_{e^*}\rangle &= \sum_{w=0}^{n-1} x_w^\top L_{e^*} x_w \\ &= \sum_{w=0}^{\frac{n}{2}-1} \left(\frac{n}{2} - 2w\right)^2 + \sum_{w=\frac{n}{2}}^{n-1} \left(\frac{3n}{2} - 2w\right)^2 = \Theta(n^3)\,.\end{aligned}$$

On the other hand, $x_w^\top L_{G\setminus e^*} x_w = nk^2$ for any w, thus $\langle X, L_{G\setminus e^*}\rangle = n^2k^2$. This is $o(n^3)$, i.e. dominated by $\langle X, L_{e^*}\rangle$, when $k = o(\sqrt{n})$. Since e^* is selected in H with probability η, we have that with probability η,

$$\langle X, L_H\rangle \geq \langle X, L_{e^*}\rangle \geq (1-o(1)) \cdot \langle X, L_G\rangle\,,$$

which violates the desired SDP approximation.

Appendix B
Hypergraph Decomposition

In this section, we describe the hypergraph decomposition algorithm used in Section V (for the proof of Theorem 3). This algorithm is nearly identical to the hypergraph decomposition step of [29, Section 4].

Algorithm B.1.

Given: A semirandom (with noise η) k-XOR instance ψ with constraint hypergraph $\mathcal{H}$ over n vertices, and a spread parameter $\tau \in (0,1)$.

Output: For each $t = 2,\dots,k$, a semirandom

```
        (with noise $\eta$) planted $\tau$-spread $p^{(t)}$-bipartite $t$-XOR instance $\psi^{(t)}$ with constraint hypergraph $\{\mathcal{H}_u^{(t)}\}_{u\in[p^{(t)}]}$, along with "discarded" hyperedges $\mathcal{H}^{(1)}$.
Operation:
    1) Initialize: $\psi^{(t)}$ to the empty instance, and $p^{(t)} = 0$ for $t = 2, \dots, k$.
    2) Fix violations greedily:
        a) Find a maximal nonempty violating $Q$. That is, find $Q \subseteq [n]$ of size $1 \leq |Q| \leq k-1$ such that $\deg(Q) = |\{C \in \mathcal{H} : Q \subseteq C\}| > \frac{1}{\tau^2}\max(1, n^{\frac{k}{2}-|Q|})$, and $\deg(Q') \leq \frac{1}{\tau^2}\max(1, n^{\frac{k}{2}-|Q'|})$ for all $Q' \supsetneq Q$.
        b) Let $q = |Q|$. Let $u = 1 + p^{(k+1-q)}$ be a new "label", and define $\mathcal{H}_u^{(k+1-q)}$ to be an arbitrary subset of $\{C \setminus Q : C \in \mathcal{H}, Q \subseteq C\}$ of size exactly $2 \cdot \lfloor \frac{1}{2\tau^2}\max(1, n^{\frac{k}{2}-q})\rfloor$.
        c) Set $p^{(k+1-q)} \leftarrow 1 + p^{(k+1-q)}$, and $\mathcal{H} \leftarrow \mathcal{H} \setminus \mathcal{H}_u^{(k+1-q)}$.
    3) If no such $Q$ exists, then put the remaining hyperedges in $\mathcal{H}^{(1)}$.
```

Lemma B.2. *Algorithm B.1 has the following guarantees:*

(1) The runtime is $n^{O(k)}$,

(2) The number of "discarded" hyperedges is $m^{(1)} := |\mathcal{H}^{(1)}| \leq \frac{1}{k\tau^2} n^{\frac{k}{2}}$,

(3) For each $t \in \{2, \dots, k\}$ and $u \in [p^{(t)}]$, $|\mathcal{H}_u^{(t)}| = \frac{m^{(t)}}{p^{(t)}} = 2\lfloor \frac{1}{2\tau^2}\max(1, n^{t-\frac{k}{2}-1})\rfloor$,

(4) For each $t = 2, \dots, k$, the instance $\psi^{(t)}$ is τ-spread.

Proof. The runtime of Algorithm B.1 is obvious. We now argue that $m^{(1)}$ is small. By construction, $\mathcal{H}^{(1)}$ is the set of remaining hyperedges when the inner loop terminates, and so we must have $\deg(\{i\}) \leq \frac{1}{\tau^2}\max(1, n^{\frac{k}{2}-1}) = \frac{1}{\tau^2} n^{\frac{k}{2}-1}$ for every $i \in [n]$; here, deg only counts hyperedges remaining in $\mathcal{H}$. We then have $\sum_{i\in[n]} \deg(\{i\}) = k|\mathcal{H}^{(1)}|$, as every $C \in \mathcal{H}^{(1)}$ is counted exactly k times in the sum. Hence, $m^{(1)} \leq \frac{1}{k\tau^2} n^{\frac{k}{2}}$.

Next, for each $t \in \{2, \dots, k\}$, by construction (Step (2b)) each $\mathcal{H}_u^{(t)}$ has the same size, namely $2\lfloor \frac{1}{2\tau^2}\max(1, n^{t-\frac{k}{2}-1})\rfloor$. It then follows that $m^{(t)} := \sum_{u\in[p^{(t)}]} |\mathcal{H}_u^{(t)}| = p^{(t)} \cdot 2\lfloor \frac{1}{2\tau^2}\max(1, n^{t-\frac{k}{2}-1})\rfloor$, and so $|\mathcal{H}_u^{(t)}| = \frac{m^{(t)}}{p^{(t)}}$. We also note that $m^{(t)}/p^{(t)}$ is clearly even.

We now argue that for each t, the instance $\psi^{(t)}$ is τ-spread. From Definition V.1, we need to prove that for each $u \in [p^{(t)}]$ and $Q \subseteq [n]$, $\deg_u(Q) \leq \frac{1}{\tau^2}\max(1, n^{\frac{k}{2}-1-|Q|})$. To see this, let $u \in [p^{(t)}]$, and let Q_u be the set "associated" with the label u, i.e., the set picked in Step (2a) of Algorithm B.1 when the label u is added in Step (2b). Note that we must have $|Q_u| = k+1-t$. Let $\mathcal{H}'$ denote the set of constraints in $\mathcal{H}$ at the time when u and $\mathcal{H}_u^{(t)}$ is added to $\psi^{(t)}$. Namely, we have that for every $C \in \mathcal{H}_u^{(t)}$, $Q_u \cup C \in \mathcal{H}'$, and Q_u, C are disjoint. Now, let $R \subseteq [n]$ be a nonempty set of size at most $t-1$. First, observe that if $R \cap Q_u$ is nonempty, then we must have $\deg_u(R) = 0$ (this degree is in the hypergraph $\mathcal{H}_u^{(t)}$). Indeed, this is because $C \cap Q_u = \emptyset$ for all $C \in \mathcal{H}_u^{(t)}$. So, we can assume that $R \cap Q_u = \emptyset$. Next, we see that $\deg_u(R) \leq \deg_{\mathcal{H}'}(Q_u \cup R)$ (where $\deg_{\mathcal{H}'}$ is the degree in $\mathcal{H}'$), as $Q_u \cup C \in \mathcal{H}'$ for every $C \in \mathcal{H}_u^{(t)}$. Because Q_u was maximal whenever it was processed in our decomposition algorithm and $Q_u \subsetneq Q_u \cup R$ as R is nonempty and $R \cap Q_u = \emptyset$, it follows that

$$\begin{aligned}\deg_{\mathcal{H}'}(Q_u \cup R) &\leq \frac{1}{\tau^2}\max(1, n^{\frac{k}{2}-|Q_u\cup R|}) = \frac{1}{\tau^2}\max(1, n^{\frac{k}{2}-|Q_u|-|R|}) \\ &= \frac{1}{\tau^2}\max(1, n^{t-\frac{k}{2}-1-|R|}) \leq \frac{1}{\tau^2}\max(1, n^{\frac{t}{2}-1-|R|})\,,\end{aligned}$$

where the last inequality follows because $t - \frac{k}{2} - 1 - |R| \leq \frac{t}{2} - 1 - |R|$ always holds, as $t \leq k$.

Finally, when $R = \emptyset$, we trivially have $\deg_u(\emptyset) = |\mathcal{H}_u^{(t)}| = 2\lfloor \frac{1}{2\tau^2}\max(1, n^{t-\frac{k}{2}-1})\rfloor \leq \frac{1}{\tau^2}\max(1, n^{t-\frac{k}{2}-1}) \leq \frac{1}{\tau^2}\max(1, n^{\frac{t}{2}-1})$, where we use again that $t - \frac{k}{2} \leq \frac{t}{2}$ as $t \leq k$. This finishes the proof. □

Appendix C
Theorem 3 when $k = 1$

In this section, we state and prove a variant of Theorem 3 for the degenerate case of $k = 1$. The algorithm here is straightforward, and we include it only for completeness.

Lemma C.1 (Algorithm for noisy 1-XOR). *Let $\eta \in (0, 1/2)$ be a constant. Let $n \in \mathbb{N}$ and $\varepsilon \in (0, 1)$, and let $m \geq O(n \log n/\varepsilon)$. There is a polynomial-time algorithm $\mathcal{A}$ that takes as input a 1-XOR instance ψ with constraint hypergraph $\mathcal{H}$ and outputs two disjoint sets $\mathcal{A}_1(\mathcal{H}), \mathcal{A}_2(\psi) \subseteq \mathcal{H}$ with the following guarantees: (1) for any instance ψ with m constraints, $|\mathcal{A}_1(\mathcal{H})| \leq \varepsilon m$ and $\mathcal{A}_1(\mathcal{H})$ only depends on $\mathcal{H}$, and (2) for any $x^* \in \{-1, 1\}^n$ and any k-uniform hypergraph $\mathcal{H}$ with at least m hyperedges, with high probability over $\psi \leftarrow \psi(\mathcal{H}, x^*, \eta)$, it holds that $\mathcal{A}_2(\psi) = \mathcal{E}_\psi \cap (\mathcal{H} \setminus \mathcal{A}_1(\mathcal{H}))$.*

Proof. First, observe that a 1-XOR instance is a degenerate case where $\mathcal{H}$ is a multiset of $[n]$ of size m. Let $S \subseteq [n]$ denote the set of $i \in [n]$ where i appears in $\mathcal{H}$ with multiplicity $\leq c \log n$, where c is a constant to be determined later. Let $\mathcal{A}_1(\mathcal{H})$ denote $\mathcal{H} \cap S$, i.e., the set of elements in $\mathcal{H}$ that are in S. We clearly have that $|\mathcal{A}_1(\mathcal{H})| \leq cn \log n \leq \varepsilon m$.

Now, let $i \notin S$. Observe that for each occurrence of i in $\mathcal{H}$, we have a corresponding *independent* right-hand side $b \in \{-1, 1\}$ where $b = x_i^*$ with probability $1 - \eta$ and $-x_i^*$ with probability η. Thus, by taking the majority, we can with high probability decode x_i^* and thus determine the corrupted constraints. It thus remains to show that with probability $\geq 1 - 1/\mathrm{poly}(n)$, the fraction of corrupted right-hand sides for i is $< \frac{1}{2}$. Indeed, by a Chernoff bound, with probability $\geq 1 - \exp(-2\delta^2 c \log n)$, it holds that the fraction of corrupted right-hand sides is at most $(\eta + \delta)$. By choosing $\delta = \frac{1}{2}(\frac{1}{2} - \eta)$ and c to be a sufficiently large constant, Lemma C.1 follows. □