Electronic Colloquium on Computational Complexity, Report No. 85 (2018)

XOR Codes and Sparse Random Linear Equations with Noise

Andrej Bogdanov[*] Manuel Sabin[†] Prashant Nalini Vasudevan[‡]

Abstract

A k-LIN instance is a system of m equations over n variables of the form $s_{i_1}+\cdots+s_{i_k} = 0$ or 1 modulo 2 (each involving k variables). We consider two distributions on instances in which the variables are chosen independently and uniformly but the right-hand sides are different. In a noisy planted instance, the right-hand side is obtained by evaluating the system on a random planted solution and adding independent noise with some constant bias to each equation; whereas in a random instance, the right-hand side is uniformly random. Alekhnovich (FOCS 2003) conjectured that the two are hard to distinguish when $k = 3$ and $m = O(n)$.

We give a sample-efficient reduction from solving noisy planted k-LIN instances to distinguishing them from random instances. Suppose that m-equation, n-variable instances of the two types are efficiently distinguishable with advantage ε. We show that $O(m \cdot (m/\varepsilon)^{2/k})$-equation, n-variable noisy planted k-LIN instances are efficiently solvable with probability $\exp -\widetilde{O}((m/\varepsilon)^{6/k})$. Our solver has worse success probability but better sample complexity than Applebaum's (SICOMP 2013).

The solver is based on a new approximate local list-decoding algorithm for the k-XOR code at large distances. The k-XOR encoding of a function $F\colon \Sigma \to \{-1, 1\}$ is its k-th tensor power $F^k(x_1, \dots, x_k) = F(x_1)\cdots F(x_k)$. Given oracle access to a function G that μ-correlates with F^k, our algorithm outputs the description of a message that $(\mu^{1/k} - \varepsilon)$-correlates with F with probability $\exp -\widetilde{O}(k^2\mu^{-2/k}\varepsilon^{-2})$. Previous decoders have a worse dependence on μ (Levin, Combinatorica 1987) or do not apply to subconstant $\mu^{1/k}$. We also prove a new XOR lemma for this parameter regime.

The decoder and its analysis rely on a new structure-versus-randomness dichotomy for Boolean-valued functions over product sets.

1 Introduction

XOR lemmas [Yao82] are statements that relate the average-case hardness of a Boolean function $F\colon \Sigma \to \{-1, 1\}$ to that of its k-XOR encoding $F^k\colon \Sigma^k \to \{-1, 1\}$ given by

$$F^k(x_1, \dots, x_k) = F(x_1)\cdots F(x_k).$$

If some algorithm A of low complexity computes F on a $(1 + \mu^{1/k})/2$-fraction of inputs under the uniform distribution over Σ, then the algorithm $A'(x_1, \dots, x_k) = A(x_1)\cdots A(x_k)$ computes F^k on

[*] andrejb@cse.cuhk.edu.hk. Department of Computer Science and Engineering and Instutite for Theoretical Computer Science and Communications, Chinese University of Hong Kong.

[†] msabin@berkeley.edu. Computer Science Division, UC Berkeley. Work partially done while visiting CUHK.

[‡] prashvas@mit.edu. Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology. Work partially done while visiting CUHK.

ISSN 1433-8092

a $(1+\mu)/2$-fraction of inputs. XOR lemmas formalize the intuition that this is essentially the best possible average-case algorithm for F^k.

In computational complexity, XOR lemmas are used for hardness amplification. Typically Σ is the Boolean cube $\{0,1\}^n$, and the function F is mildly hard to compute in the sense that every circuit of a given size fails on some small fraction of $\{0,1\}^n$. The objective is to show that F^k is very hard to compute, namely no circuit of somewhat smaller size can compute F on more than $(1+\mu)/2$ of the inputs.

In coding-theoretic language, XOR lemmas are approximate local list-decoding algorithms for the k-XOR code [STV01]. The function F represents a binary message of length $|\Sigma|$ and F^k is its encoding. List-decoding is the task of finding all the codewords F^k that have relative agreement at least $(1+\mu)/2$ with a given corrupted codeword $G\colon \Sigma^k \to \{-1,1\}$. The number of such codewords can be exponentially large in $|\Sigma|$ which severely restricts the utility of exact list-decoding. It is common to study the following relaxation.[1]

Definition 1. Let $\mu, \alpha \in [0,1]$ be parameters. A binary code is $(1-\alpha)/2$-*approximately list-decodable* for error rate $(1-\mu)/2$ with *list size* $\ell(\mu,\alpha)$ if for every corrupted codeword G there exists a list of codewords $C_1,\dots,C_\ell$ such that for any codeword C that is $(1-\mu)/2$-close to G, C is $(1-\alpha)/2$-close to C_i for some i.

When $\alpha > \mu$ the list size is still exponential in Σ (see Proposition 29), so the regime of interest is $\alpha \le \mu$. In the case of the k-XOR code, two codewords F^k, F'^k are at distance $(1-\alpha)/2$ if and only if the corresponding messages F, F' are at distance $(1-\alpha^{1/k})/2$. Most XOR lemmas [Imp95,GNW11, IJK09,IJKW10] study the algorithmic aspects of approximate list-decodability in the regime where μ is close to zero and $\alpha^{1/k}$ is close to one. Impagliazzo, Jaiswal, Kabanets, and Wigderson [IJKW10] give a local decoding algorithm with list size $O(1/\mu)$ assuming the *approximation error* $(1-\alpha^{1/k})/2$ is at most $O((\log 1/\mu)/k)$. Their algorithm, as well as most others, does not address the regime in which k is smaller than $\log 1/\mu$. One notable exception is Levin's XOR lemma [Lev87], which achieves approximation $\alpha^{1/k} = \mu^{1/k} - \varepsilon$ for arbitrary $\varepsilon > 0$ but with list size that grows at least exponentially in $1/\mu^2$ (see discussion below).

In this work study approximate list-decoding XOR codes in the regime $\mu = o(2^{-k})$ and even $\mu = o(1/|\Sigma|)$. Our main motivation for investigating this extreme parameter setting is the following connection between the XOR code and the hardness of noisy systems of sparse random linear equations over $GF(2)$.

Sparse random linear equations We are given a system of linear equations of the form $x_{i_1} + \cdots + x_{i_k} = 0$ or 1, where the variables in each equation and the left-hand sides of the different equations are independent and identically distributed, and the addition is modulo 2. Such a system has the form $Ax = b$ where A is a random $m \times n$ matrix with sparse and independent rows. We are interested in the following two problems:

Solving a planted instance: Given A and $As + e$, where $s \sim \{0,1\}^n$ is a random planted solution and e is a vector of random i.i.d. $\{0,1\}$-entries where each entry is 1 with constant probability η, find s.

Distinguishing planted from random instances: Distinguish the distribution $(A, As+e)$ from (A, r), where $r \sim \{0,1\}^m$ is independent of A.

[1] More precisely, the list can be of size $2^{h((1-\mu^{1/k})/2)|\Sigma|}$, where h is the binary entropy function.

The distinguishing variant was introduced by Alekhnovich [Ale11]. He conjectured that when $k = 3$ and $m = O(n)$ distinguishing with advantage substantially better than $1/n$ is intractable. This is a generalization of Feige's conjecture [Fei02] from which several hardness of approximation results are derived.

For the solving variant, Applebaum [App16] describes an efficient solver in the regime $m = \omega_k(n^{k/2})$ by a reduction to a 2CSP instance, the application of a suitable approximation algorithm [GW95, CW04], and some additional post-processing work.[2]

Distinguishing algorithms have received considerable attention in the regime where their advantage is very close to one. A *refutation algorithm* must accept all planted instances in which the error rate is less than some threshold, say 2η, and reject almost all random instances. Polynomial-time refutation algorithms are known for random k-XOR instances provided the number of clauses m exceeds $\omega_k(n^{k/2})$ [AOW15,BM16] and are conjectured not to exist when $m = o(n^{k/2})$ [ABW10,BM16]. In the latter regime, refutation in time $\exp \tilde{O}(n^{\delta})$ is possible if $m = \tilde{\omega}_k(n^{k/2-\delta(k/2-1)})$ [RRS17].

On the negative side, Feldman, Perkins, and Vempala [FPV15] describe a statistical model in which efficient search is possible when $m = \omega(n^{k/2}\log^2 n)$, but distinguishing isn't when $m = o((n/\log n)^{k/2})$. Kothari et al. [KMOW17] show that the refutation algorithms of [AOW15,BM16, RRS17] are optimal among a wide class of semidefinite programs.

A distinguishing advantage of $\Omega(\binom{m}{2}/\binom{n}{k})$ can be attained by a simple collision-finder: The distinguisher looks for two appearances of the same equation, accepts if the right-hand sides are equal, and rejects otherwise.

It is a curious coincidence that the threshold $m \approx n^{k/2}$ arises as a common barrier for solving and distinguishing.

We are interested in the relationship between solving planted systems of this type and distinguishing planted and random instances. In one direction, a solver that works on an α-fraction of instances can be used to distinguish with advantage at least $\alpha - 2^{-H(2\eta)m+n}$, indicating that solving should be harder than distinguishing. In the other direction, Applebaum [App13] gives an efficient reduction from solving a constant fraction of instances of size m to that of distinguishing instances of size $(\varepsilon^2 m/\log n)^{1/3}$, where ε is the advantage of the distinguisher.

The starting point of Applebaum's analysis is an application of Yao's distinguishing-to-prediction reduction [Yao82]. In this context the reduction turns a distinguisher for planted instances with m equations and advantage ε into a predictor that guesses the value of any given k-XOR equation (evaluated on the planted solution) with advantage ε/m, given $m-1$ planted equations as "training data".

Our reduction leverages the fact that the truth-table of the predictor is precisely a corrupted k-XOR encoding of the planted solution, so recovering the solution amounts to decoding the predictor. In this setting, the correlation ε/m between the predictor and the actual codeword is smaller than the inverse of the message length n. None of the known list-decoding algorithms address this range of parameters.

This perspective has been applied successfully in many cryptographic settings, including analysis of hard-core predicates [GL89, AGS03, App17] and sample-preserving distinguishing-to-search reductions for learning with errors[3] [MM11, BLRL$^+$18] and learning with rounding [BGM$^+$16]. All these applications rely on the availability of *exact* list-decoding algorithms for the underlying code.

[2] This approach applies more generally to equations with adversarial noise.

[3] Learning with errors instances over $GF(2)$ are precisely dense variants of noisy k-XOR and amount to list-decoding the "dense XOR code", i.e. the Hadamard code.

1.1 Our results

Our first result is a new approximate local list-decoding algorithm for the k-XOR code at very large distances. We use the notation $\mathrm{E}[A \cdot B]$ for the product of $A(x)$ and $B(x)$ averaged over their inputs, i.e., $\mathrm{E}[A \cdot B] = \mathrm{E}_x[A(x) \cdot B(x)]$.

Definition 2. A $(\mu, \alpha^{1/k})$ *approximate list-decoder* for the k-XOR code is an algorithm that, given a corrupted codeword G such that $\mathrm{E}[G \cdot F^k] \geq \mu$ for some message F, outputs a message $\hat{F}$ such that $\mathrm{E}[F \cdot \hat{F}] \geq \alpha^{1/k}$ with some success probability p.

The approximate list-decoder is *local* if G is provided as an oracle, and its output $\hat{F}$ is a circuit (which on input x calculates $\hat{F}(x)$), and *uniform* if its dependence on the parameters μ, $|\Sigma|$, k, and α is uniform (that is, it uses no non-uniform advice).

Theorem 3. *There is a uniform local $(\mu, \mu^{1/k} - \varepsilon)$-approximate list decoder that succeeds with probability at least $\Omega(\varepsilon)^{O(k^2/\mu^{2/k}\varepsilon^2)}$ and runs in time $\tilde{O}(k^k \mu^{-2} \varepsilon^{-2k} \log|\Sigma|)$.*

In contrast, the algorithm of Impagliazzo et al. [IJKW10] assumes that $\mu^{1/k}$ is lower-bounded by a constant, while the algorithm implicit in Levin's XOR lemma [Lev87] succeeds with probability exponential in μ^{-2}.

We apply Theorem 3 to derive the following search-to-decision reduction for noisy linear equations.

Theorem 4. *Suppose that m-equation, n-variable planted η-noisy kLIN instances are distinguishable from random ones in time t with advantage ε, where $\eta < 1/2$. Then, planted instances with $O(m \cdot (m/\varepsilon)^{2/k} + 2^{2k} n \log n/k)$ equations and n variables can be solved in time polynomial in t, m, n, and $1/\varepsilon$, with probability at least $(\varepsilon/m)^{O(k(m/\varepsilon)^{6/k})}$ over the choice of the instance and the randomness of the solver.*

In contrast, the solver in Applebaum's reduction (see Proposition 23) requires more than m^3/ε^2 equations but succeeds with high probability. It is possible to obtain other tradeoffs between the sample complexity and the success probability of the solver.

By combining Theorem 4 with the refutation algorithm of Raghavendra, Rao, and Schramm, it follows that for constant k and constant noise rate, for every constant $2/3 < \delta < 1$, m-equation, n-variable planted noisy k-LIN instances can be solved with probability $2^{-\tilde{O}(m^{6/k})}$ in time $2^{\tilde{O}(n^{\delta})}$ as long as $m = \tilde{\Omega}(n^{(1-\delta)k/2+1+2\delta/k})$.

Other consequences As a corollary of Theorem 3 we obtain an upper bound on the list size for k-XOR codes at high error rates.

Corollary 5. *For $\alpha = (\mu^{1/k} - \varepsilon)^k$, $\ell(\mu, \alpha) = O(|\Sigma|/\varepsilon)^{O(k^2/\mu^{2/k}\varepsilon^2)}$.*

The value of α in Corollary 5 is close to optimal. Proposition 29 shows that when $\alpha > \mu$ the list size becomes exponential in $|\Sigma|$. We do not know, however, if the list size has to be exponentially large in $\mu^{\Theta(1/k)}$ when $\alpha \leq \mu$. Proposition 30 proves the lower bound $\ell = \Omega(\alpha^{2/k}\mu^{-2})$ for all α, assuming $\mu \geq |\Sigma|^{-1/2}$. Proposition 28 gives a much tighter non-constructive upper bound when $\alpha < \mu^2$. All of these bounds are proved in Section 4.

We also obtain the following consequence for non-uniform hardness amplification in the low-correlation regime. Corollary 6 improves the amount of advice in Levin's proof [Lev87] from linear in μ^{-2} to linear in $\mu^{-O(1/k)}$.

Corollary 6. *There is a log-time uniform oracle circuit L with $O(k\log(1/\varepsilon)(n+\mu^{-2/k}\varepsilon^{-2}))$ bits of advice and size $\tilde{O}(k^k\mu^{-2}\varepsilon^{-2k})$ such that, if G predicts F^k with advantage μ, then for some setting of the advice L^G predicts $F\colon \{0,1\}^n \to \{-1,1\}$ with advantage at least $\mu^{1/k}-\varepsilon$.*

Corollary 6 is proved in Section 2.4.

1.2 Techniques for list-decoding the XOR code

Our proof of Theorem 3 is a derandomization of Levin's proof [Lev87] (see also [GNW11]). We begin with a short outline of his proof and point out its limitations with respect to list size. This motivates the two main innovations introduced in our work: A new notion of regularity for functions over product sets, and an analysis of a natural sampler for regular functions.

In the ensuing discussion we ignore the locality of the list-decoder, so the concepts are introduced in less general form than in Section 2.

Levin's XOR lemma Here is an outline of Levin's proof for $k=2$. The correlation assumption $\mathrm{E}[G\cdot F^2]\geq\mu$ can be written in the form

$$\mathrm{E}_x\big[F(x)\cdot \mathrm{E}_y[F(y)G(x,y)]\big]\geq\mu.$$

One case is that the inner expectation is at least $\sqrt{\mu}$ in absolute value for some x. Then the column function $G_x(y)=G(x,y)$ predicts F up to sign with advantage $\sqrt{\mu}$. Otherwise, all inner expectations are bounded by $\sqrt{\mu}$. Then the function $\mu^{-1/2}\tilde{F}$, where

$$\tilde{F}(x)=\mathrm{E}_y[F(y)G(x,y)]$$

is $[-1,1]$-bounded and predicts F with advantage $\sqrt{\mu}$. The function $\mu^{-1/2}\tilde{F}$ can be estimated to within ε pointwise from $\tilde{\Theta}(1/\varepsilon^2\mu)$ samples $F(y)G(x,y)$ for random y. Then F can be predicted with advantage $\sqrt{\mu}-\varepsilon$ given $\tilde{O}(1/\varepsilon^2\mu)$ pairs $(y,F(y))$ as advice.

More generally, given a correlation assumption of the form $\mathrm{E}[A(x)B(y)G(x,y)]\geq\alpha\beta$, either some column of G predicts B up to sign with advantage β, or else the empirical average $\beta^{-1}\,\mathrm{E}[G(x,y)B(y)]$ taken over $\tilde{\Theta}(1/\varepsilon^2\beta^2)$ samples usually predicts A with advantage $\alpha-\varepsilon$. Since ε must be less than α, the number of required samples grows at least quadratically in the inverse of the advantage $1/\alpha\beta$.

Levin's k-XOR lemma is proved by applying this proposition inductively. By setting $A=F^i,\alpha=\mu^{i/k}$ and $B=F^{k-i},\beta=\mu^{(k-i)/k}$, proving a k-XOR lemma is reduced to proving an i-XOR lemma and a $(k-i)$-XOR lemma. Even though different choices of the parameter i lead to different proofs, the resulting decoder always requires at least $\tilde{\Omega}(1/\alpha^2\beta^2)=\tilde{\Theta}(1/\mu^2)$ values of F as advice.

Our derandomized XOR lemma We illustrate our improvement on the list size for the 3-XOR code when $\varepsilon=\mu^{1/3}/2$. For these parameters Theorem 3 gives a list of size $\exp\tilde{O}(1/\mu^{4/3})$, which improves upon Levin's $\exp\tilde{\Theta}(1/\mu^2)$.

Assume $\mathrm{E}[F(x)F(y)F(z)G(x,y,z)]\geq\mu$. In case $\mathrm{E}[F(y)F(z)G(x,y,z)]$ is at least $\mu^{2/3}$ in magnitude for some x, we apply Levin's 2-XOR lemma to the function $G_x(y,z)=G(x,y,z)$ to obtain a list of size $\tilde{O}(1/\mu^{4/3})$. Otherwise, we may assume that the function

$$\tilde{F}(x)=\mathrm{E}_{y,z}[H_x(y,z)],\qquad\text{where } H_x(y,z)=F(y)F(z)G_x(y,z)$$

is bounded in magnitude by $\mu^{2/3}$ for all x. Levin's proof proceeds by estimating $\mu^{-2/3}\tilde{F}$ pointwise with precision $\mu^{1/3}/2$, or equivalently estimating $\tilde{F}$ pointwise with precision $\mu/2$. This requires $\tilde{\Theta}(1/\mu^2)$ samples of the form $F(y)F(z)$ coming from a set S of *independent* random pairs (y, z).

The source of our improvement in list size is an emulation of the random set S by a (small number of) random *product* set(s) $S_Y \times S_Z$, $|S_Y| = |S_Z|$ of comparable size. Since the list size is exponential in the advice length, this effectively reduces it from from $\exp(|S|)$ bits to $\exp(|S_Y|+|S_Z|)$ bits.

In general, random product sets are poor samplers. For example, if $H_x(y, z)$ happens to be a dictator in y (i.e. independent of z), then a random sample of size s^2 would produce a $\Theta(s)$-biased estimate. A random product sample of the same size would yield a $\Theta(\sqrt{s})$-biased estimate, wiping away any potential savings. But then $|\mathrm{E}_z[H_x(y_0, z)]|$ equals one for any dictator value y_0, so $G_x(y_0, z)$ is an exact decoding of $F(z)$ up to sign.

Our approximate list-decoder for the k-XOR code is based on a structure versus randomness dichotomy: Either the function H_x is "regular", in which case the product sampler accurately emulates a truly random sampler, or else one of the rows or columns of H_x is "structured", in which case the problem reduces to approximately list-decoding the $(k-1)$-XOR code.

Sampling regular functions Let $H(y, z)$ be a function with $|\mathrm{E}[H]| = \mu^{2/3}$. We call H *regular* if all rows and columns of H are pseudorandom in the sense that $|\mathrm{E}_y[H(y, z)]| \leq \mu^{1/3}$ for all z and $|\mathrm{E}_z[H(y, z)]| \leq \mu^{1/3}$ for all y. If one of the functions H_x is not regular, then one of the columns or rows of G_x already predicts F with advantage $\mu^{1/3}$ up to sign.

Our main technical result is Lemma 8, which shows that if H is regular, then a product sampler of complexity $|S_Y| = |S_Z| = \tilde{O}(\mu^{-4/3})$ estimates $\mathrm{E}[H]$ to within $\mu/2$ with constant probability. If all H_x are regular then F can be predicted with $\tilde{O}(\mu^{-4/3})$ bits of advice, giving the desired list size.

The product sampler is an unbiased estimator of $\mathrm{E}[H]$. Lemma 8 is proved by upper bounding its variance for regular functions by $o(\mu^2)$. This amounts to comparing the bias of the product and random samplers on a typical pair of samples (y, z) and (y', z'). The only difference is that the pairs (y, y') and (z, z') have a higher collision probability in the product sampler. Conditioned on neither of these pairs colliding, (y, z) and (y', z') are identically distributed for both samplers.

For the variance analysis, the product sampler is therefore modeled by the following process: With probability $1 - o(\mu^{4/3})$ emulate the random sampler, with probability $o(\mu^{4/3})$ fix $y = y'$ to a random value and emulate the random sampler for the function $H_y(z) = H(y, z)$, with probability $o(\mu^{4/3})$ do the same with the roles of the two coordinates reversed, and with probability $o(\mu^{8/3})$ fix both $y = y'$ and $z = z'$ to random values and output the constant $H_{yz} = H(y, z)$. By the regularity assumption, each of these cases contributes $o(\mu^2)$ to the variance, giving the desired conclusion.

1.3 Techniques for hardness versus randomness of noisy linear equations

The proof of Theorem 4 is based on the paradigm of Goldreich and Levin [GL89] for converting hardness into pseudorandomness in cryptographic settings. Yao's reduction [Yao82] is first applied to convert the distinguisher into a predictor. The truth-table of the predictor is then viewed as a corrupted codeword with respect to a suitable encoding of the planted solution. A decoding algorithm is then used to recover the solution.

In the setting of noisy random k-LIN instances, the predictor is a function that, given $m-1$ equations from the planted distribution as "training data", produces a guess for the value of the m-th equation. Given good training data, the truth-table of the predictor is therefore a corrupted codeword of the k-XOR code. A distinguisher with advantage ε yields a predictor with advantage $\mu = \varepsilon/(1-2\eta)m$ in expectation over the choice of the training data. This step of the reduction is also carried out (in greater generality) in the work of Applebaum [App13]. He then amplifies the advantage of the predictor by using independent samples of the training data up to the point where the solution can be uniquely extracted.

To avoid the increase in sample complexity, we instead apply our list-decoding algorithm for the k-XOR code to the predictor. With noticeable probability, the list-decoder outputs an approximate solution $\hat{s}$ that $\mu^{1/k}/2$-correlates with the planted solution s. In other words, the output of the list-decoder predicts the value of s_i for a random index i with advantage $\mu^{1/k}/2$. Our main insight is Claim 26 which shows that, owing to the symmetries of the k-LIN instance, the same advantage can be attained for an arbitrary i. This allows the advantage to be amplified by repetition (see Claim 27). Once it is sufficiently large, the solution can be extracted using a technique of Bogdanov and Qiao [BQ12].

1.4 Organization

In Section 2 we describe and analyze our approximate list-decoding algorithm for the k-XOR code and prove Theorem 3 and Corollary 6. In Section 3 we describe the reduction from distinguishing to solving noisy random k-LIN instances and prove Theorem 4. In Section 4 we prove Corollary 5 and some additional upper and lower bounds for the approximate list size of the k-XOR code.

2 Approximately list-decoding the XOR code

In this section we prove Theorem 3. Section 2.1 introduces the notion of regularity for product functions and analyzes the variance of product samplers. Section 2.2 describes and analyzes the list-decoding algorithm for a function G under the assumption that most of the functions $G(x_1, \dots, x_{k-1}, a)F(x_1)\cdots F(x_{k-1})$ are regular. Section 2.3 describes the list-decoder for general functions and proves Theorem 3.

2.1 Regularity and product samplers

Suppose $R \colon \Sigma^k \to \{-1,1\}$ is a random function each of whose entries are i.i.d. with some unknown bias that we are interested in estimating up to precision μ. Chebyshev's inequality guarantees that about $1/\mu^2$ samples are sufficient to produce an accurate estimate with constant probability. When R is random, it is irrelevant how the samples are chosen as long as they are all distinct. In particular they can be chosen from a *product sample* of the form $S_1 \times \cdots \times S_k$ where $|S_1| = \cdots = |S_k| = 1/\mu^{2/k}$.

For general functions, however, product samplers produce substantially poorer estimates than random samplers of the same size. For example, if $H \colon \Sigma^k \to \{-1,1\}$ is a dictator (that is, fully determined by one of its inputs) then the accuracy of the product sampler drops to $O((1/\mu)^{2/k})$.

Regularity is a pseudorandomness property of bounded functions over product sets that guarantees the product sampler has about the same accuracy as for a random function. In this context, the crucial property of the random function turns out to be its "closure" under input restriction: If some subset I of inputs is restricted, the product sampler on the remaining inputs has standard deviation $\mu^{2(k-|I|)/k}$. This motivates the following definition of regularity.

Definition 7. A function $H\colon \Sigma^k \to \{-1,1\}$ is (μ,λ)-*regular* if for all nonempty $I \subseteq [k]$,

$$\mathrm{E}[H(x_1,\dots,x_k)H(x'_1,\dots,x'_k) \mid x_I = x'_I] \leq \mu^2\lambda^{-|I|}.$$

H is *strongly* (μ,λ)-*regular* if the inequality also holds for $I = \varnothing$.

(The notation $x_I = x'_I$ is shorthand for "$x_i = x'_i$ for all $i \in I$.")

The regularity requirement is worst-case in the sense that it must hold for all subsets of coordinates, but average-case in the sense that once the coordinates of the input variables to be restricted are fixed, the deviation need only be small on average over the choice of the restricted values.

The parameter setting that is consistent with the above discussion is $\lambda = \mu^{2/k}$. For our intended application it is convenient to allow for a small deviation from this value and so the definition is stated in this more general form.

The main result of this section is the following lemma which bounds the variance of the product sampler with respect to regular functions.

Lemma 8. *If H is (μ,λ)-regular and $S_1,\dots,S_k$ are mutually independent and individually pairwise independent subsets of Σ of size s each, then*

$$\mathrm{Var}_{S_1,\dots,S_k}\,\mathrm{E}[H(x_1,\dots,x_k) \mid x_i \in S_i \textit{ for all } i] \leq \left(\left(1+\frac{1}{\lambda s}\right)^k - 1\right)\cdot\mu^2.$$

An interesting setting of parameters is $\lambda = \mu^{2/k}$ and $s = O(k\cdot\mu^{-2/k})$. The variance of the product sampler is then bounded by μ^2 just like for a random function at a (small) multiplicative cost of $O(k)$ in the sizes of the sets $S_1,\dots,S_k$.

Proof. Let $\mathbf{S} = (S_1,\dots,S_k)$, $\mathbf{x} = (x_1,\dots,x_k)$ and $\mathbf{x}' = (x'_1,\dots,x'_k)$. In this notation, the variance of interest equals

$$\begin{aligned}\mathrm{Var}_{\mathbf{S}}\,\mathrm{E}_{\mathbf{x},\mathbf{x}'}[H(\mathbf{x}) \mid \mathbf{x}\in\mathbf{S}] &= \mathrm{E}_{\mathbf{S}}\big[\mathrm{E}[H(\mathbf{x}) \mid \mathbf{x}\in\mathbf{S}]^2\big] - \mathrm{E}[H]^2\\ &= \mathrm{E}_{\mathbf{S},\mathbf{x},\mathbf{x}'}[H(\mathbf{x})H(\mathbf{x}') \mid \mathbf{x},\mathbf{x}'\in\mathbf{S}] - \mathrm{E}[H]^2. \qquad (1)\end{aligned}$$

The triples $(S_1,x_1,x'_1),\dots,(S_k,x_k,x'_k)$ in the first term are independent. Moreover, the induced marginal distribution on every pair (x_i,x'_i) is

$$(x_i,x'_i) \sim \begin{cases}\text{identical uniformly random element in } \Sigma, & \text{with probability } 1/s,\\ \text{uniformly random pair of distinct elements in } \Sigma, & \text{with probability } 1-1/s.\end{cases}$$

This distribution has the following alternative description: Flip a coin C_i with probability of heads $p = (1/s - 1/|\Sigma|)/(1-1/|\Sigma|) \leq 1/s$ and sample

$$(x_i,x'_i) \sim \begin{cases}\text{identical uniformly random element in } \Sigma, & \text{if } C_i \text{ is heads},\\ \text{independent uniformly random pair in } \Sigma\times\Sigma, & \text{if } C_i \text{ is tails}.\end{cases}$$

Letting $I \subseteq [k]$ denote the set of those i for which C_i came out heads we can write

$$\begin{aligned}\mathrm{E}_{\mathbf{S},\mathbf{x},\mathbf{x}'}[H(\mathbf{x})H(\mathbf{x}') \mid \mathbf{x},\mathbf{x}'\in\mathbf{S}] &= \mathrm{E}_{I,\mathbf{x},\mathbf{x}'}[H(\mathbf{x})H(\mathbf{x}') \mid x_I = x'_I]\\ &= \sum_{I\subseteq[k]} p^{|I|}(1-p)^{k-|I|}\,\mathrm{E}_{\mathbf{x},\mathbf{x}'}[H(\mathbf{x})H(\mathbf{x}') \mid x_I = x'_I]\\ &\leq E[H]^2 + \sum_{I\subset[k]}\left(\frac{1}{s}\right)^{|I|}\cdot\mu^2\lambda^{-|I|}.\end{aligned}$$

Plugging into (1) it follows that

$$\begin{aligned}\mathrm{Var}_{\mathbf{S}}\, \mathrm{E}_{\mathbf{x},\mathbf{x}'}[H(\mathbf{x}) \mid \mathbf{x} \in \mathbf{S}] &\le \sum_{I \subset [k]} \Big(\frac{1}{s}\Big)^{|I|} \cdot \mu^2 \lambda^{-|I|} \\ &\le \mu^2 \sum_{I \subset [k]} \Big(\frac{1}{\lambda s}\Big)^{|I|} \\ &= \mu^2 \bigg(\Big(1 + \frac{1}{\lambda s}\Big)^k - 1\bigg). \end{aligned}$$ □

The success probability of the sampler can be increased by taking the median run of several independent repetitions.

Repeated product sampler S^H:

```
Choose independent sets $S_{ij}$, $1 \le i \le k$, $1 \le j \le t$ of size $s$ each.
Output the median value of $\mathrm{E}[H(x) \mid x_i \in S_{ij}$ for all $i]$ among all $t$ such values.
```

The following claim states the effectiveness of the product sampler. The additional parameter θ controls the tradeoff between the accuracy of the estimate and the product sample size and can be initially thought of as a small constant.

Claim 9. *Assuming H is (μ, λ)-regular, $s \ge k/\theta\lambda$, and $t \ge 8 \log 1/\eta$, with probability at least $1 - \eta$, $|S^H - \mathrm{E}[H]| \le 2\sqrt{\theta/(1-\theta)} \cdot \mu$.*

Proof. By Chebyshev's inequality, for any j, the probability that the estimator

$$E_j = \mathrm{E}[H(x) \mid x_i \in S_i \text{ for all } i]$$

deviates by more than two standard deviations from its mean $\mathrm{E}[H]$ is at most $1/4$. By Lemma 8 and the choice of parameters, the standard deviation is at most $\sqrt{(1+\theta/k)^k - 1} \cdot \mu \le \sqrt{\theta/(1-\theta)} \cdot \mu$.

Since the estimators E_j are independent and each one falls within two standard deviations of $\mathrm{E}[H]$ with probability at least $3/4$, by a large deviation bound the probability that more than half of them fall outside this range is less than $2^{-t/8} \le \eta$. □

2.2 Approximately list-decoding product-sampleable functions

In this section we describe and analyze the list-decoder assuming the correlation function $H = G \cdot F^k$ is "product-sampleable", meaning that most restrictions to the last coordinate yield a regular function. The argument follows Levin's proof of the XOR lemma, except that we apply the product sampler from Section 2.1 in lieu of Levin's random sampler.

Definition 10. $H \colon \Sigma^k \to \{-1, 1\}$ is *product-sampleable* with error ε if for all but an ε-fraction of inputs $a \in \Sigma$, the functions $H_a(x_1, \dots, x_{k-1}) = H(x_1, \dots, x_{k-1}, a)$ are all strongly $(\mu^{(k-1)/k}, \frac{1}{2}\mu^{2/k})$-regular, where $\mu = |\mathrm{E}[H]|$.

Lemma 11. *Assume $k \ge 2$. There is a uniform local $(\underline{\mu}, \underline{\mu}^{1/k} - \varepsilon)$ list decoder that succeeds with probability at least $(\varepsilon/80)^{O((k-1)^2/\underline{\mu}^{2/k}\varepsilon^2)}$ and runs in time $k^2 \log|\Sigma| \cdot \mathrm{poly}(\underline{\mu}^{-1/k}, \varepsilon^{-1})$ on input G, assuming $H = G \cdot F^k$ is product-sampleable with error at most $\varepsilon\mu^{(k-1)/k}/80$, where $\mu = |\mathrm{E}[H]| \ge \underline{\mu}$.*

Approximate list-decoder $LPS^G(k, \underline{\mu}, |\Sigma|, \varepsilon)$:

```
1   Set $s = \lceil 514(k-1)/\underline{\mu}^{2/k}\varepsilon^2 \rceil$ and $t = \lceil 8 \log(80/\varepsilon) \rceil$.
2   Choose independent sets $S_{ij} \subseteq \Sigma$, $1 \le i \le k-1$, $1 \le j \le t$ of size $s$ each.
3   Guess the values $F(x)$ at random for all $x \in S_{ij}$.
4   For every $a$ in $\Sigma$:
5       Let $G_a \colon \Sigma^{k-1} \to \{-1, 1\}$ be the function $G(x_1, \dots, x_{k-1}, a)F(x_1) \cdots F(x_{k-1})$.
6       Let $\tilde{F}(a)$ be the median value of $\mathrm{E}[G_a(x) \mid x_i \in S_{ij}$ for all $i]$.
7   Choose a uniformly random $B$ from the range $[-1, 1]$ within $\lceil \log(4/\varepsilon) \rceil$ bits of precision.
8   Let $\tilde{F}_B(a) = 1$ if $[\![\mu^{-(k-1)/k}\tilde{F}(a)]\!] \ge B$ and $-1$ if not.
9   Output $\pm\tilde{F}_B$ where the sign is chosen at random.
```

In step 8, $[\![\cdot]\!] \colon \mathbb{R} \to [-1, 1]$ denotes the rounding function

$$[\![t]\!] = \begin{cases} 1, & \text{if } t > 1, \\ t, & \text{if } -1 \le t \le 1, \\ -1, & \text{if } t < -1. \end{cases}$$

Steps 2 to 6 implement the product sampler. The output of this sampler produces real-valued estimates $\hat{F}(a)$ of the message bits $F(a)$. The accuracy of the product sampler guarantees that when G_a is regular, $[\![\hat{F}]\!]$ is likely to significantly correlate with F. In order to extract a $\{-1, 1\}$-valued codeword from $\hat{F}$, steps 7 and 8 rounds its values with respect to the random threshold B. The rounding preserves the correlation in expectation. The expectation can be turned into a noticeable probability at a small price in accuracy.

Fact 12. *$[\![\cdot]\!]$ is a contraction, i.e., $|[\![s]\!] - [\![t]\!]| \le |s - t|$ for all s and t.*

Proof of Lemma 11. Let R be the set of all a for which the function H_a is strongly $(\mu^{(k-1)/k}, \frac{1}{2}\mu^{2/k})$-regular, where $\mu = |\mathrm{E}[H]|$. By assumption, $\overline{R}$ has measure at most $\varepsilon\mu^{(k-1)/k}/80$. For every $a \in N$, the function

$$G_a(x_1, \dots, x_{k-1}) = H_a(x_1, \dots, x_{k-1})F(a) = G(x_1, \dots, x_{k-1}, a)F(x_1) \cdots F(x_{k-1}).$$

is also strongly $(\mu^{(k-1)/k}, \frac{1}{2}\mu^{2/k})$-regular, as G_a and H_a may differ only in sign.

By Claim 9 with parameters $\theta = \varepsilon^2/257$ and $\eta = 80/\varepsilon$, for all but at most $\varepsilon/80$ of those a that are in R,

$$\left|\tilde{F}(a) - \mathrm{E}[G_a(x)]\right| \le \mu^{(k-1)/k} \cdot \frac{\varepsilon}{8} \tag{2}$$

with probability at least $1 - \varepsilon/80$ over the random choices in step 2. Let $A \subseteq R$ be the set of those a's for which inequality (2) holds. Then A has expected measure at least $1 - \varepsilon/80 - \varepsilon\mu^{(k-1)/k}/80 \ge 1 - \varepsilon/40$. By Markov's inequality, A has measure at least $1 - \varepsilon/20$ with probability at least 1/2 (3).

If a is in A, then by (2),

$$\left|\mu^{-(k-1)/k}\tilde{F}(a) - \mu^{-(k-1)/k}\,\mathrm{E}[G_a(x)]\right| \le \frac{\varepsilon}{8}.$$

By the strong regularity of G_a, $|\mu^{-(k-1)/k}\,\mathrm{E}[G_a(x)]| \le 1$. Since $[\![\cdot]\!]$ is a contraction,

$$\left|[\![\mu^{-(k-1)/k}\tilde{F}(a)]\!] - \mu^{-(k-1)/k}\,\mathrm{E}[G_a(x)]\right| \le \frac{\varepsilon}{8}.$$

If a is in R (but not in A), then strong regularity still holds and

$$\bigl|\llbracket \mu^{-(k-1)/k}\tilde{F}(a)\rrbracket - \mu^{-(k-1)/k}\,\mathrm{E}[G_a(x)]\bigr| \leq 2,$$

as both terms take values between -1 and 1. Finally, if a is not in R then

$$\bigl|\llbracket \mu^{-(k-1)/k}\tilde{F}(a)\rrbracket - \mu^{-(k-1)/k}\,\mathrm{E}[G_a(x)]\bigr| \leq 1 + \mu^{-(k-1)/k} \leq 2\mu^{-(k-1)/k}.$$

Therefore

$$\begin{aligned} \bigl|\mathrm{E}_a\bigl[F(a)\cdot(\llbracket \mu^{-(k-1)/k}\tilde{F}(a)\rrbracket - \mu^{-(k-1)/k}\,\mathrm{E}[\tilde{G}_a(x)])\bigr]\bigr| &\leq \frac{\varepsilon}{8}\cdot\Pr[a\in A] + 2\cdot\Pr[a\in R\setminus A] \\ &\quad + 2\mu^{-(k-1)/k}\cdot\Pr[a\notin R] \\ &\leq \frac{\varepsilon}{8} + 2\cdot\frac{\varepsilon}{20} + 2\mu^{-(k-1)/k}\cdot\frac{\varepsilon\mu^{(k-1)/k}}{80} \\ &\leq \frac{\varepsilon}{4}. \end{aligned} \tag{4}$$

By the definition of μ,

$$\bigl|\mathrm{E}_a\bigl[F(a)\cdot\mu^{-(k-1)/k}\,\mathrm{E}[G_a(x)]\bigr]\bigr| = \mu^{-(k-1)/k}\bigl|\mathrm{E}_{x,a}\bigl[G(x,a)\cdot F(x_1)\cdots F(x_{k-1})F(a)\bigr]\bigr| = \mu^{1/k}. \tag{5}$$

From (4), (5) and the triangle inequality it follows that

$$\bigl|\mathrm{E}_a\bigl[F(a)\cdot\llbracket \mu^{-(k-1)/k}\tilde{F}(a)\rrbracket\bigr]\bigr| \geq \mu^{1/k} - \frac{\varepsilon}{4}. \tag{6}$$

If B was a uniform $[-1,1]$ random variable, $\mathrm{E}_B[\tilde{F}_B(a)]$ would equal $\llbracket \mu^{-(k-1)/k}\tilde{F}(a)\rrbracket$. As B is precision-bounded, we have the weaker guarantee

$$\bigl|\mathrm{E}_B[\tilde{F}_B(a)] - \llbracket \mu^{-(k-1)/k}\tilde{F}(a)\rrbracket\bigr| \leq \frac{\varepsilon}{4} \tag{7}$$

for every $a\in\Sigma$. From (6) and (7) it follows that

$$\bigl|\mathrm{E}_{B,a}[F(a)\cdot\tilde{F}_B(a)]\bigr| \geq \mu^{1/k} - \frac{\varepsilon}{2}.$$

By Markov's inequality, the inequality

$$\bigl|\mathrm{E}_a[F(a)\cdot\tilde{F}_B(a)]\bigr| \geq \mu^{1/k} - \varepsilon.$$

must hold for at least a $\varepsilon/2$ (8) fraction of B's. If such a B is chosen, the correlation between the output and F is at least $\mu^{1/k} - \varepsilon$ with probability 1/2 (9) over the choice of sign in step 9.

To summarize, conditioned on events (3), (8), and (9) occurring and the guesses in step 3 of the algorithm being correct, the output of the algorithm has the desired correlation with F. As step 3 involves guessing at most $(k-1)st$ boolean values, the algorithm succeeds with probability at least

$$\frac{\varepsilon}{8}\cdot 2^{-(k-1)st} \geq \Bigl(\frac{\varepsilon}{80}\Bigr)^{O((k-1)^2/\underline{\mu}^{2/k}\varepsilon^2)}$$

by our choice of parameters. □

2.3 Proof of Theorem 3

The approximate list-decoder L guesses whether the function $H = G \cdot F^k$ is product-sampleable. If its guess is positive it runs the list-decoding algorithm LPS for product-sampleable functions from Section 2.2. Definitions 10 and 7 ensure that if H is not product-sampleable then a noticeable fraction of its restrictions have large bias. In this case, L guesses the suitable restriction and runs recursively on it.

The following specification is obtained by unwinding the recursion with uniform guessing probabilities. This choice turns out to be convenient for the analysis. We use the notation a_I to describe a partial assignment restricted to the subset of indices $I \subseteq [k]$, and a_i as a shorthand for $a_{\{i\}}$.

Algorithm $L^G(k, \underline{\mu}, |\Sigma|, \varepsilon)$**:**

```
1   Choose a random subset $R \subseteq [k]$ and a random partial assignment $a_R \sim \Sigma^R$.
2   Let $G' \colon \Sigma^{\overline{R}} \to \{-1, 1\}$ to be the function $G$ restricted to $x_R = a_R$.
3   If $|\overline{R}| = 0$, fail.
4   If $|\overline{R}| = 1$, output $G'$ or $-G'$ with equal probability.
5   Otherwise, output $LPS^{G'}(|\overline{R}|, \underline{\mu}^{|\overline{R}|/k}, |\Sigma|, \varepsilon)$.
```

To prove Theorem 3, we will show by strong induction on k that L is a $(\underline{\mu}, \underline{\mu}^{1/k} - \varepsilon)$-approximate list decoder with success probability at least

$$p(k, \underline{\mu}) = 2^{-k-1} \cdot (\varepsilon/80)^{C(k-1)^2/\underline{\mu}^{2/k}\varepsilon^2}$$

where C is a sufficiently large constant.

Base case $k = 1$**:** R is non-empty with probability $1/2$. In this case the larger one of $\mathrm{E}[F \cdot G]$ and $\mathrm{E}[F \cdot (-G)]$ is $\mu \geq \underline{\mu} - \varepsilon$, so the output of L has correlation at least $\underline{\mu} - \varepsilon$ with F with probability $1/4$, which is larger than $p(1, \underline{\mu})$.

Inductive step: Assume $k \geq 2$ and the claim holds up to $k-1$. To prove it holds for k we consider two cases. Let $H = G \cdot F^k$ and $\mu = |\mathrm{E}[H]|$.

If H is product-sampleable with error at most $\varepsilon\mu^{(k-1)/k}/80$, then in step 1 the empty set is chosen with probability 2^{-k}, in which case step 5 is triggered with $G' = G$. By Lemma 11, the output of LPS^G $(\underline{\mu}^{1/k} - \varepsilon)$-correlates with F with probability at least $(\varepsilon/80)^{C(k-1)^2/\underline{\mu}^{2/k}\varepsilon^2}$, so the overall success probability exceeds $p(k, \underline{\mu})$ as desired.

The following claim summarizes the irregularity of functions that are not product-sampleable.

Claim 13. *If H is not product-sampleable then with probability at least $\varepsilon\mu^{(k-1)/k}/80$ over the choice of a_k there exists a proper subset $I \subset [k]$ with $k \in I$ for which with probability at least $\mu^{2|\overline{I}|/k}$ over the choice of $a_{I\setminus\{k\}}$, $|\mathrm{E}[H(x) \mid x_I = a_I]| \geq \mu^{|\overline{I}|/k}$.*

Proof. By Definition 10, for more then a $\varepsilon\mu^{(k-1)/k}/80$ fraction of a_k there exists a subset $I \subseteq [k]$, $k \in I$ that violates strong regularity in the following sense:

$$\mathrm{E}[H(x)H(x') \mid x_I = x'_I, x_k = x'_k = a_k] > (\mu^{(k-1)/k})^2 \cdot (\tfrac{1}{2}\mu^{2/k})^{-|I\setminus\{k\}|} = 2^{|I|-1} \cdot \mu^{2|\overline{I}|/k}. \tag{10}$$

The subset I must be proper because when $|I| = k$, the right-hand side exceeds one. Fix such an a_k and let a_I be a random extension of a_k. By the independence of $x_{\overline{I}}$ and $x'_{\overline{I}}$,

$$\mathrm{E}_{a_I}\big[\mathrm{E}[H(x) \mid x_I = a_I]^2\big] = \mathrm{E}[H(x)H(x') \mid x_I = x'_I, x_k = x'_k = a_k]. \tag{11}$$

When $I = \{k\}$, the claim follows from (10) and (11) after taking square roots. Otherwise, by Markov's inequality $\mathrm{E}[H(x) \mid x_I = a_I]^2 \geq \mu^{2|\overline{I}|/k}$ with probability at least $(2^{|I|-1} - 1)\mu^{2|\overline{I}|/k}$ over the choice of a_I. Because $2^{|I|-1} - 1$ is at least 1, the claim follows again after taking square roots. □

Let U be the event "$|\mathrm{E}[H(x) \mid x_I = a_I]| \geq \mu^{|\overline{I}|/k}$ and $I \subseteq R$." By Claim 13 and the uniform choice of R, U has probability at least

$$\Pr[U] \geq 2^{-|I|} \cdot \frac{\varepsilon\mu^{(k-1)/k}}{80} \cdot \mu^{2|\overline{I}|/k} \geq \frac{(\varepsilon/80)\mu^{3(k-1)/k}}{2^{|I|}}. \tag{12}$$

Claim 14. *Conditioned on U, the output $L^G(k, \underline{\mu})$ has correlation at least $\underline{\mu}^{1/k} - \varepsilon$ with F with probability at least $p(|\overline{I}|, \underline{\mu}^{|\overline{I}|/k})$.*

Proof. Conditioned on U, the view of $L(k, \underline{\mu})$ when querying the oracle G is identical to the view of $L(|\overline{I}|, \underline{\mu}^{|\overline{I}|/k})$ when querying the oracle $\hat{G}(x_{\overline{I}}) = G(x_{\overline{I}}, a_I)$. Also conditioned on U, the function $\hat{H}(x_{\overline{I}}) = H(x_{\overline{I}}, a_I)$ is $\mu^{|\overline{I}|/k}$-biased. The function $\hat{H}$ equals

$$\hat{H}(x_{\overline{I}}) = \sigma \cdot \hat{G}(x_{\overline{I}}) \cdot \prod\nolimits_{i \in \overline{I}} F(x_i)$$

where $\sigma = \prod_{i \in I} F(a_i)$ is a possible change of sign. By inductive assumption, $L^{\hat{G}}(|\overline{I}|, \underline{\mu}^{|\overline{I}|/k})$ then has correlation at least $(\underline{\mu}^{|\overline{I}|/k})^{1/|\overline{I}|} - \varepsilon = \underline{\mu}^{1/k} - \varepsilon$ with F with the desired probability. □

From (12) and Claim 14 it follows that $L^G(k, \underline{\mu})$ succeeds with probability at least

$$\begin{aligned}
\Pr[U] \cdot p(|\overline{I}|, \underline{\mu}^{|\overline{I}|/k}) &\geq \frac{(\varepsilon/80)\mu^{3(k-1)/k}}{2^{|I|}} \cdot 2^{-|\overline{I}|-1} \cdot (\varepsilon/80)^{C(|\overline{I}|-1)^2/\underline{\mu}^{2/|\overline{I}|}\varepsilon^2} \\
&\geq 2^{-k-1} \cdot (\varepsilon/80)^{3k-2} \cdot (\varepsilon/80)^{C(k-2)^2/\underline{\mu}^{2/k}\varepsilon^2} \\
&\geq 2^{-k-1} \cdot (\varepsilon/80)^{C(k-1)^2/\underline{\mu}^{2/k}\varepsilon^2} \\
&= p(k, \underline{\mu})
\end{aligned}$$

assuming $\varepsilon \leq \mu^{1/k}$ in the second inequality and $C \geq 4$ in the third one. This completes the inductive step and the proof of Theorem 3.

2.4 Proof of Corollary 6

Let $\Sigma = \{0, 1\}^n$. The proof of Theorem 3 shows that the circuit L^G computes a function that predicts F with advantage $\mu^{1/k} - \varepsilon$ with positive probability. In particular, the prediction succeeds for some fixed choice of the randomness. The amount of advice is therefore upper bounded by the randomness complexity of L^G.

The randomness complexity of the list-decoder LPS is governed by the $(k-1)t$ choices of the sets S_{ij} chosen in step 2 and the $(k-1)st$ values of F guessed in step 3. If the elements of each set S_{ij} are chosen in a pairwise independent manner, step 2 can be performed using at most $2ktn$ bits of randomness. Plugging in the parameters for s and t we conclude that LPS has randomness complexity $O(k \log(1/\varepsilon)(n + \mu^{-2/k}\varepsilon^{-2}))$.

In step 5 of the list-decoder L the randomness complexity of the call to LPS is maximized when R is the empty set. In step 1, L requires $k(n+1)$ additional bits of randomness, so the randomness complexity of L is also $O(k \log(1/\varepsilon)(n + \mu^{-2/k}\varepsilon^{-2}))$ as desired.

3 From Distinguishing to Solving Random Noisy Linear Equations

In this section, we show how to use an approximate list-decoder of the k-XOR code to reduce solving random planted k-LIN instances to distinguishing them from completely random instances.

Definition 15 (k-LIN)**.** Let $k, m, n \in \mathbb{N}$, and $\eta \in [0, 1/2)$. An m-equation n-variable *k-LIN instance* is a pair (A, b), where $A \in \{0,1\}^{m\times n}$ is such that each row a_i of A has at most k non-zero entries, and $b \in \{0,1\}^m$. A *random planted η-noisy* k-LIN instance is such a pair $(A, As+e)$ where:

- Each row of $A \in \{0,1\}^{m\times n}$ is sampled independently from the *row distribution* $\mathcal{R}_{n,k}$, which is the modulo 2 sum of k independent random indicator vectors in $\{0,1\}^n$ (i.e. vectors of the form $(0,\ldots,0,1,0,\ldots,0)$).
- $s \in \{0,1\}^n$, called the *planted solution*, is chosen uniformly at random.
- $e \in \{0,1\}^m$, called the *noise vector*, is chosen such that each bit in it is 1 independently with probability η.

In the rest of this section, unless specified otherwise, the number of equations in a k-LIN instance is to be taken to be m, the number of variables to be n, and the noise to be η.

Remark 1. Another natural distribution on the rows of A is the uniform distribution on strings on Hamming weight k. Our results and analysis can be modified to apply to this distribution as well.

Definition 16 (Solving and Distinguishing k-LIN)**.** We define the following two operations for random k-LIN instances:

- An algorithm S is said to *solve* planted η-noisy k-LIN instances with *success probability* p if, given a random planted η-noisy k-LIN instance $(A, As+e)$, it outputs s with probability p (over the randomness of A, e and S itself).

- An algorithm D is said to *distinguish* planted η-noisy k-LIN instances from random with *advantage* ε if it distinguishes, with advantage ε, between a random planted η-noisy k-LIN instance $(A, As+e)$ and (A, r), where A is chosen as in k-LIN, but $r \sim \{0,1\}^m$ is chosen at uniform independently of A. That is,

$$\mathrm{E}_{A,s,e}[\mathsf{D}(A, As+e)] - \mathrm{E}_{A,r}[\mathsf{D}(A,r)] \geq \varepsilon$$

We will be reducing the task of solving a k-LIN instance with m'-equations to that of distinguishing instances with m equations from random (with advantage ε) for some $m' > m$. Our objective here is to keep m' as small as possible in relation to m. It is already known (from [App13]) how to perform such a reduction with $m' = \tilde{\Theta}(m^3/\varepsilon^2)$ that results in constant success probability for the solver (see Proposition 23). Using the approximate list-decoder constructed in Section 2, we are able to bring m' down significantly at the cost of lower success probability.

Theorem 17 (Refined Theorem 4)**.** *Suppose that m-equation n-variable planted η-noisy kLIN instances are distinguishable from random in time t with advantage ε. Then planted η-noisy kLIN instances with m' equations and n variables can be solved in time polynomial in t, m, n, $1/\varepsilon$, and $1/(1-2\eta)$ with probability at least p where:*

$m' = O\big((1-2\eta)^{2/k}\cdot m\cdot (m/\varepsilon)^{2/k} + 2^{2k}\; n\log n/k(1-2\eta)^2\big)$ *and* $p = (\varepsilon/(1-2\eta)m)^{O(k((1-2\eta)m/\varepsilon)^{6/k})}$.

The rest of this section is dedicated to the proof of this theorem. Our approach for getting a solver for k-LIN from a distinguisher is broadly divided into the following three parts, each of which we describe briefly below:

1. Using the distinguisher to get a predictor.
2. Using the predictor to get an approximate-solver.
3. Using the approximate-solver to get an actual solver.

The first step is to show (Lemma 19 in Section 3.1) that a distinguisher for k-LIN can be used to construct a predictor that, given a noisy k-LIN instance $(A, As + e)$, has a small advantage in predicting the answers to random equations – that is, the value of $\langle a, s\rangle$ for random a from the row distribution $\mathcal{R}_{n,k}$. This operation is defined as below.

Definition 18. An algorithm P is called a *predictor* for η-noisy k-LIN with *advantage* δ if, when given a random planted η-noisy k-LIN instance $(A, As + e)$ and a random "row" a from the row distribution $\mathcal{R}_{n,k}$, predicts $\langle a, s\rangle$ with advantage δ. That is,

$$\mathrm{E}_{A,s,e,a}[\mathsf{P}(a; A, As + e) \cdot (-1)^{\langle a,s\rangle}] \geq \delta$$

We say that such a predictor P *predicts* s with advantage δ from the *training data* $(A, As + e)$.

The predictor is constructed from the distinguisher using standard hybrid arguments. The following lemma is proven in Section 3.1.

Lemma 19. *Suppose there is an algorithm that distinguishes m-equation n-variable planted η-noisy k-LIN instances from random with advantage ε and runs in time t. Then, there is a predictor for m-equation n-variable η-noisy k-LIN that also runs in time t and has advantage $\varepsilon/(1 - 2\eta)m$.*

Once we have such a predictor, we then use it to solve k-LIN "approximately". That is, given a planted instance $(A, As + e)$, we recover an $\tilde{s}$ that correlates well with s. We use the following shorthand for the measure of correlation between two binary strings. Given $s, \tilde{s} \in \{0, 1\}^n$,

$$s \cdot \tilde{s} = \mathrm{E}_i[(-1)^{s[i]}(-1)^{\tilde{s}[i]}]$$

where the expectation is over i drawn at random from $[n]$. Note that this quantity is contained in $[-1, 1]$, and $s \cdot \tilde{s} = \gamma$ is the same as saying that s and $\tilde{s}$ agree on a $(1+\gamma)/2$ fraction of coordinates. We also overload this notation to handle the case where s (or $\tilde{s}$) is a $\{-1, 1\}$-string, in which case $(-1)^{s[i]}$ in the expression above is to be replaced with $s[i]$.

The operation of approximately solving k-LIN instances is now defined as below.

Definition 20 (Approximately Solving k-LIN)**.** An algorithm $\tilde{\mathsf{S}}$ is said to γ*-approximately solve* planted η-noisy k-LIN instances with *success probability* p if, given a random planted η-noisy k-LIN instance $(A, As + e)$, with probability p it outputs some $\tilde{s}$ such that $s \cdot \tilde{s} \geq \gamma$.

To construct an approximate solver for k-LIN from a predictor, we use the approximate list-decoder for the k-XOR code. Given a k-LIN instance $(A, As + e)$ as training data, we view the "truth-table" of the predictor $\mathsf{P}(\cdot; A, As+e)$ as a corrupt codeword of the k-XOR code. Intuitively, the correctness of the predictor should say that this codeword is not too far from the k-XOR encoding of s. We are unable to decode or list-decode this codeword, however, as the noise in the codeword is too high. Instead, we use the approximate list-decoder and obtain, as one of the elements in the list, an $\tilde{s}$ that is noticeably correlated with s. We then amplify this correlation by exploiting certain symmetries of k-LIN. See Section 3.2 for a more thorough exposition of the intuition behind this approach and the proof of the following lemma that states our results in this respect.

Lemma 21. *Let $\mu, \alpha, \gamma \in [0,1]$, and $k \in \mathbb{N}$ be a constant. For some $m, n \in \mathbb{N}$, suppose:*

1. *There is a predictor for m-equation n-variable η-noisy k-LIN that runs in time t_1 and has advantage δ.*
2. *There is a $(\mu, \alpha^{1/k})$ approximate list-decoder for the k-XOR code with messages of length n that runs in time t_2 and has success probability p.*

Let $r = 8\log(8/(1-\gamma))/\alpha^{2/k}$. Then, there is an algorithm that γ-approximately solves (mr)-equation n-variable planted η-noisy k-LIN instances that runs in time $\widetilde{O}(r(t_1 + t_2 + mn))$, and has success probability $\frac{3}{4}\left[p\left(\delta - \mu\right)\right]^r$.

The final step in our reduction is to convert the approximate solution produced by the approximate solver above into an actual solution. To do this, we employ a technique of Bogdanov and Qiao [BQ12]. In brief, given an approximate solution $\tilde{s}$, to recover the first bit $s[1]$ of the actual solution, we first find a number of equations where the first bit is involved. In each of these equations, we pretend that $\tilde{s}$ is correct about the values of the remaining bits and solve for $s[1]$, and finally set $s[1]$ to be the majority answer. This is repeated for each bit of s and, if enough equations are used, all the bits are recovered correctly. The end result in our case is stated in the following lemma.

Lemma 22. *Assuming $m \geq 40\ n\log n/k(1-2\eta)^2\gamma^{2(k-1)}$, there is a $O(mn^2)$-time algorithm that, given a m-equation, n-variable planted noisy k-LIN instance $(A, As+e)$ and a γ-approximate solution $\hat{s}$ that is independent of A and e, outputs s with probability $1 - o(1)$.*

We finish our proof by putting the above lemmas together with the approximate list-decoder for the k-XOR code from Section 2.

Proof of Theorem 17. The hypothesis of the theorem promises a k-LIN distinguisher that runs in time t and has advantage ε. Lemma 19 now immediately implies the existence of a k-LIN predictor P that runs in time t and has advantage $\delta = (\varepsilon/(1-2\eta)m)$.

Set $\mu = \delta/2$ and α to be such that $\alpha^{1/k} = \mu^{1/k}/2$ $(= \mu^{1/k} - \mu^{1/k}/2)$. Theorem 3 implies a $(\mu, \alpha^{1/k})$ approximate list-decoder for the k-XOR code that runs in time $\widetilde{O}(\mu^{-4}\log n)$, and has success probability $\Omega(\mu)^{O(k/\mu^{4/k})}$.

Set $\gamma = 1/2$. Along with the above predictor and list-decoder, Lemma 21 now implies an algorithm that γ-approximately solves m'-equation n-variable planted η-noisy k-LIN instances, where m' is equal to:

$$10 \cdot \frac{32\ m}{\alpha^{2/k}} = 1280 \cdot \frac{m}{\mu^{2/k}} \leq 2560 \cdot \frac{m}{\delta^{2/k}} = 2560 \cdot (1-2\eta)^{2/k} \cdot m \cdot \left(\frac{m}{\varepsilon}\right)^{2/k}$$

This approximate solver runs in time on the order of:

$$\frac{1}{\alpha^{2/k}} \cdot (t + \mu^{-4} \log n + mn)) = \mathrm{poly}(t, (1-2\eta), m, n, 1/\varepsilon)$$

It has success probability at least:

$$\begin{aligned}\frac{3}{4}\left[(\mu)^{O(k/\mu^{4/k})} \cdot (\delta - \mu)\right]^{32/\alpha^{2/k}} &\geq \mu^{O(k/\mu^{4/k}\alpha^{2/k})} \\ &\geq \left(\frac{\varepsilon}{(1-2\eta)m}\right)^{O\left(k((1-2\eta)m/\varepsilon)^{6/k}\right)}\end{aligned}$$

With the above probability, we have a γ-approximate solution. In order to recover the actual solution, we apply Lemma 22 with this approximate solution and a fresh set of m'' equations with the same planted solution, where $m'' = 40 \cdot 2^{2(k-1)}\ n \log n/k(1-2\eta)^2$. This gives us the actual solution, incurs an additional running time of $O(mn^2)$ and the final success probability becomes the above multiplied by $(1-o(1))$. This completes the proof of the theorem. □

In the rest of this section, we prove Lemmas 19, 21 and 22 (in Sections 3.1, 3.2, and 3.3, respectively).

3.1 From Distinguishing to Prediction: Proof of Lemma 19

Our proof of Lemma 19 uses standard hybrid arguments and distinguishing-to-prediction reductions [Yao82]. The same technique was applied in the more general setting of Goldreich's one-way function by Applebaum [App13].

Proof of Lemma 19. Recall that we consider the distribution on k-LIN instances $(A, As+e)$, where $s \sim \{0,1\}^n$ is random, each row of $A \in \{0,1\}^{m\times n}$ is drawn from $\mathcal{R}_{n,k}$, and the entries of e are independent $\{0,1\}$-bits each equal to 1 with probability η. Let D be the distinguisher promised by the hypothesis. Then, taking $r \in \{0,1\}^m$ to be uniformly random, we have the following:

$$\mathrm{E}_{A,s,e}[\mathsf{D}(A, As+e)] - \mathrm{E}_{A,r}[\mathsf{D}(A,r)] \geq \varepsilon, \tag{13}$$

Denote the rows of A by $a_1, \ldots, a_m$. Consider the following hybrid distributions (for all $i \in [m]$):

$$H_i = (A, \langle a_1, s\rangle + e_1, \ldots, \langle a_i, s\rangle + e_i, r_{i+1}, \ldots, r_m)$$

By (13), for some i we must have:

$$\mathrm{E}[\mathsf{D}(H_i)] - \mathrm{E}[\mathsf{D}(H_{i-1})] \geq \varepsilon/m$$

Without loss of generality, assume $i = m$. Let A' be the first $m-1$ rows of A and u be a random bit. We can write the above expression as:

$$\mathrm{E}_{A',a_m,s,e',e_m}[D(A', a_m, A's+e', \langle a, s\rangle + e_m)] - \mathrm{E}_{A',a_m,e',s,u}[D(A', a_m, A's+e', u)] \geq \varepsilon/m$$

Given a k-LIN instance (A', b') and a row a to predict for, the predictor P chooses a random $u \sim \{0,1\}$ and outputs $(-1)^u \cdot \mathsf{D}(A', a, b', u)$. The advantage of P as a predictor for k-LIN is:

$$\mathrm{E}_{A',a,s,e'}[\mathsf{P}(a; A', A's+e') \cdot (-1)^{\langle a,s\rangle}] = \mathrm{E}_{A',a,s,e',\tilde{e}}[\mathsf{P}(a; A', A's+e') \cdot (-1)^{\langle a,s\rangle + \tilde{e}}] \cdot \mathrm{E}_e[(-1)^{\tilde{e}}]$$

where $\tilde{e}$ is a bit set to be 1 with probability η independent of everything else. We can now bound the first term in the above product as:

$$\begin{aligned}
&\mathrm{E}_{A',a,s,e',\tilde{e}}[\mathsf{P}(a; A', a, A's+e')\cdot(-1)^{\langle a,s\rangle+\tilde{e}}] \\
&\qquad= \mathrm{E}_{A',a,s,e',\tilde{e}}\left[\frac{1}{2}\Big(\mathsf{D}(A',a,A's+e',\langle a,s\rangle+\tilde{e}) - \mathsf{D}(A',a,A's+e',\overline{\langle a,s\rangle+\tilde{e}})\Big)\right] \\
&\qquad= \mathrm{E}_{A',a,s,e',\tilde{e}}[\mathsf{D}(A',a,A's+e',\langle a,s\rangle+\tilde{e})] - \mathrm{E}_{A',a,s,e',u}[\mathsf{D}(A',a,A's+e',u)] \\
&\qquad\geq \varepsilon/m.
\end{aligned}$$

As the bias of $\tilde{e}$ is equal to $(1-2\eta)$, which is positive, the above two expressions imply that:

$$\mathrm{E}_{A',a,s,e'}[\mathsf{P}(a; A', A's+e')\cdot(-1)^{\langle a,s\rangle}] \geq \frac{\varepsilon}{(1-2\eta)m}. \tag{14}$$

This proves the lemma, and P is the necessary predictor. □

We note that Applebaum's reduction [App13] proceeds by amplifying the advantage of the predictor via repetition, giving the following conclusion.

Proposition 23. *Suppose that m-equation, n-variable planted η-noisy k-LIN instances are distinguishable from random in time t with advantage ε. Then there is a solver for random planted η-noisy n-variable k-LIN instances with $O(m\cdot(m/\varepsilon)^2\log n)$ equations that runs in time polynomial in the running time of the distinguisher and has constant success probability.*

3.2 From Prediction to Approximate Solving: Proof of Lemma 21

Given a predictor with advantage δ, that is

$$\mathrm{E}_{A,a,s,e}[\mathsf{P}(a; A, As+e)\cdot(-1)^{\langle a,s\rangle}] \geq \delta$$

we aim to recover an approximate solution $\hat{s}$ that correlates with s. By Markov's inequality, the function $G(a) = \mathsf{P}(a; A, As+e)$ as a function of a has correlation at least $\delta/2$ with $\langle a,s\rangle$ for at least a $\delta/2$-fraction of the choices of A, s, and e. The function $\langle a,s\rangle$ is the k-XOR encoding s^k of s (under a reduced representation), so our list-decoder from Section 2 outputs an approximate codeword $\tilde{s}$ that, say, $\delta^{1/k}/2$-correlates with s with noticeable probability. This is the solver $\widetilde{\mathsf{S}}$ we obtain in Claim 25.

It remains to amplify this correlation to a sufficiently large threshold so that the solution can be fully recovered by the correction procedure in Section 3.3. A natural idea is to run $\widetilde{\mathsf{S}}$ several times on independent training data and take pointwise majorities of its outputs. However, this transformation may not be effective. It could be the case that the output of $\widetilde{\mathsf{S}}$ completely reveals the first $\delta^{1/k}n/2$ bits of s but provides no information about the rest. Then additional runs of $\widetilde{\mathsf{S}}$ wouldn't yield any additional information about the planted solution s.

To rule out this scenario, we first apply a random self-reduction to the instance $(A, b = As+e)$. More precisely, we run the approximate solver $\widetilde{\mathsf{S}}$ on the instance $(A', b') = (A\pi, b + A\pi s')$ where π is a random permutation (matrix) and $s' \sim \{0,1\}^n$ is a random shift and recover the answer via the transformation $\pi(\widetilde{\mathsf{S}} + s')$.

This transformation preserves the correlation between the output of the solver and the planted solution. On the other hand, since s is hidden and its coordinates are randomly permuted, the coordinates on which s and $\pi(\widetilde{\mathsf{S}} + s')$ agree are uniformly distributed (conditioned on the number of agreements). In particular, for any fixed coordinate i, whenever $\widetilde{\mathsf{S}}$ succeeds, s and $\pi(\widetilde{\mathsf{S}}+s')$ agree in position i with probability at least $1/2 + \delta^{1/k}/4$. Incidentally, the fact that s is hidden means that this agreement is as good for any planted solution s, which simplifies some of our technical arguments. We call this pair of properties *uniformity*.

Definition 24 (Uniform Solver)**.** A k-LIN solver $\widetilde{\mathsf{S}}$ is *uniform* if its advantage is uniformly distributed across all planted solutions and co-ordinates. To be precise, it satisfies the following conditions for any $\gamma \in [-1, 1]$:

- *Uniformity across solutions:* The probability that $\widetilde{\mathsf{S}}$ outputs a γ-approximate solution is independent of the planted solution. That is, for any $s \in \{0,1\}^n$,

$$\Pr_{A,e,\widetilde{\mathsf{S}}}\left[s \cdot \widetilde{\mathsf{S}}(A, As+e) = \gamma\right] = \Pr_{s',A,e,\widetilde{\mathsf{S}}}\left[s' \cdot \widetilde{\mathsf{S}}(A, As'+e) = \gamma\right]$$

- *Uniformity across co-ordinates:* For any s, the correlation of $\tilde{s} \leftarrow \widetilde{\mathsf{S}}(A, As+e)$ with s is distributed uniformly across all the co-ordinates. That is, for any $s \in \{0,1\}^n$ and $i \in [n]$,

$$\mathrm{E}_{\tilde{s}}[(-1)^{s[i]+\tilde{s}[i]} |\ s \cdot \tilde{s} = \gamma] = \gamma$$

Lemma 21 is a direct consequence of the following three claims.

Claim 25. *Under the assumptions of Lemma 21, there is an algorithm that $\alpha^{1/k}$-approximately solves m-equation n-variable planted η-noisy k-LIN instances that runs in time $O(t_1 + t_2)$, and has success probability at least $p\,(\delta - \mu)$.*

Claim 26. *Suppose there is an algorithm that γ-approximately solves m-equation n-variable planted η-noisy k-LIN instances that runs in time t, and has success probability p. Then, there is an algorithm that uniformly γ-approximately solves k-LIN with the same parameters, runs in time $t + \widetilde{O}(mn)$, and has success probability p.*

Claim 27. *Let $\gamma, \gamma' \in (0, 1]$ such that $\gamma' > \gamma$, and $r = 8 \cdot \log(8/(1-\gamma'))/\gamma^2$. Suppose there is an algorithm that uniformly γ-approximately solves m-equation n-variable planted η-noisy k-LIN instances that runs in time t, and has success probability p. Then, there is an algorithm that γ'-approximately solves (mr)-equation n-variable planted η-noisy k-LIN instances that runs in time $O(rt)$, and has success probability at least $3p^r/4$.*

Proof of Claim 25. Let P and LD denote the predictor and the approximate list-decoder, respectively. Let $G(i_1, \ldots, i_k)$ be the function $P([i_1, \ldots, i_k]; A, b)$, where $[\cdot]$ is a conversion to the format required by the predictor, i.e. as a vector in $\{0,1\}^n$ whose j-th bit is the parity of the number of appearances of i among $i_1, \ldots, i_k$. The approximate solver outputs LD^G. By assumption,

$$\mathrm{E}_{A,e,s,a}\left[\mathsf{P}(a; A, As+e) \cdot (-1)^{\langle a,s\rangle}\right] \geq \delta$$

For fixed A and b,

$$\mathrm{E}_{i_1,\ldots,i_k}[G(i_1, \ldots, i_k) \cdot (-1)^{s[i_1]+\cdots+s[i_k]}] = \mathrm{E}_a[\mathsf{P}(a; A, b) \cdot (-1)^{\langle a,s\rangle}]$$

by the definition of G and the change of representation convention. Combining these two equations (and abusing notation to treat G also as a string once its randomness is fixed), we conclude that $\mathrm{E}[G \cdot s^k] \geq \delta$, where the expectation is taken over the choice of the instance $A, b = Ax + e$, and the randomness of P. By Markov's inequality,

$$\Pr\left[G \cdot s^k < \mu\right] \leq \frac{1-\delta}{1-\mu} = 1 - \frac{\delta - \mu}{1-\mu} \leq 1 - (\delta - \mu)$$

By our assumption on the list decoder, whenever the above condition does not happen, LD^G outputs a message $\tilde{s}$ that $\alpha^{1/k}$-correlates with s with probability at least p. So the solver succeeds with probability at least $p(\delta - \mu)$ as desired. □

Notation. In the following proof, for a permutation $\pi : [n] \to [n]$, we also use π to denote the corresponding permutation matrix. For a matrix A with n columns, $A\pi$ denotes the multiplication of A by the matrix π, which gives the matrix resulting from applying the permutation π to the *columns* of A; and for a vector s, the matrix-vector product πs gives the vector resulting from applying π to the coordinates of s.

Proof of Claim 26. Let $\widetilde{\mathsf{S}}$ be the γ-approximate solver from the hypothesis. The following solver has the additional uniformity property:

The uniform approximate solver $\widetilde{\mathsf{S}}'$ on input $(A, b = As + e)$ works as follows:

```
Pick a random permutation $\pi : [n] \to [n]$ and $s' \sim \{0,1\}^n$. Set $A' = A\pi$ and $b' = b + A's'$.
Compute $\tilde{s} \leftarrow \widetilde{\mathsf{S}}(A', b')$, and set $\tilde{s}' = \pi(\tilde{s} + s')$.
Output $\tilde{s}'$.
```

The running time of $\widetilde{\mathsf{S}}'$ can be verified by inspection. We start by showing that $\widetilde{\mathsf{S}}'$ is uniform across solutions. For any $s \in \{0,1\}^n$ and $\gamma' \in [-1,1]$, we are concerned with the following probability:

$$\Pr_{\tilde{s}'}[s \cdot \tilde{s}' = \gamma'] = \Pr_{\tilde{s},\pi,s'}[s \cdot \pi(\tilde{s} + s') = \gamma'] = \Pr_{\tilde{s},\pi,s'}[(\pi^{-1}s + s') \cdot \tilde{s} = \gamma'] = \Pr_{\tilde{s},s''}[s'' \cdot \tilde{s} = \gamma']$$

where we have denoted $(\pi^{-1}s + s')$ by s'' – note that s'' is uniformly distributed, independently of s. The last expression is the probability that the output of $\widetilde{\mathsf{S}}$ has correlation γ' with s''. The input to $\widetilde{\mathsf{S}}$ is:

$$(A', b') = (A', As + e + A's') = (A', A'(\pi^{-1}s + s') + e) = (A', A's'' + e)$$

which is a random k-LIN instance independent of s. Thus, the last probability above, which only involves random variables independent of s, is also independent of s, and hence, so is the probability that $s \cdot \tilde{s}' = \gamma'$. This shows that $\widetilde{\mathsf{S}}'$ is uniform across solutions.

Next we show that $\widetilde{\mathsf{S}}'$ is indeed a γ-approximate solver with success probability p. This success probability, following the equalities above, is as follows:

$$\Pr_{s,\tilde{s}'}[s \cdot \tilde{s}' \geq \gamma] = \Pr_{s,\tilde{s},\pi,s'}[(\pi^{-1}s + s') \cdot \tilde{s} \geq \gamma] = \Pr_{\tilde{s},s''}[s'' \cdot \tilde{s} \geq \gamma]$$

where, as argued before, s'' is uniformly distributed, and $\tilde{s}$ is the output of $\widetilde{\mathsf{S}}$ on input $(A', A's'' + e)$. In other words, the input to $\widetilde{\mathsf{S}}$ is actually a random k-LIN instance with a random planted solution, and the last expression above is its success probability as a γ-approximate solver. This shows that $\widetilde{\mathsf{S}}'$ has the same success probability as $\widetilde{\mathsf{S}}$, which is p.

It remains to show that $\widetilde{\mathsf{S}}'$ is uniform across co-ordinates. For any $s \in \{0,1\}^n$, $i \in [n]$ and $\gamma' \in [-1,1]$, we are concerned with the following expectation, which we manipulate in the same way as the expressions above:

$$\begin{aligned}
\mathrm{E}_{\tilde{s}'}\left[(-1)^{s[i]+\tilde{s}'[i]} \;\middle|\; s\cdot\tilde{s}' = \gamma'\right] &= \mathrm{E}_{\tilde{s},\pi,s'}\left[(-1)^{s[i]+\pi(\tilde{s}+s')[i]} \;\middle|\; s\cdot\pi(\tilde{s}+s') = \gamma'\right] \\
&= \mathrm{E}_{\tilde{s},\pi,s'}\left[(-1)^{(\pi^{-1}s+s')[\pi^{-1}(i)]+\tilde{s}[\pi^{-1}(i)]} \;\middle|\; (\pi^{-1}s+s')\cdot\tilde{s} = \gamma'\right] \\
&= \mathrm{E}_{\tilde{s},\pi,s''}\left[(-1)^{s''[\pi^{-1}(i)]+\tilde{s}[\pi^{-1}(i)]} \;\middle|\; s''\cdot\tilde{s} = \gamma'\right]
\end{aligned}$$

This time, we use the fact that the distribution of s'' (and hence $\tilde{s}$) is independent of π. This implies that $\pi^{-1}(i)$ is distributed uniformly over $[n]$, independently of both s'' and $\tilde{s}$. Thus, we have:

$$\begin{aligned}
\mathrm{E}_{\tilde{s},\pi,s''}\left[(-1)^{s''[\pi^{-1}(i)]+\tilde{s}[\pi^{-1}(i)]} \;\middle|\; s''\cdot\tilde{s} = \gamma'\right] &= \mathrm{E}_{\tilde{s},s''}\left[\mathrm{E}_{\pi}\left[(-1)^{s''[\pi^{-1}(i)]+\tilde{s}[\pi^{-1}(i)]}\right] \;\middle|\; s''\cdot\tilde{s} = \gamma'\right] \\
&= \mathrm{E}_{\tilde{s},s''}\left[s''\cdot\tilde{s} \;\middle|\; s''\cdot\tilde{s} = \gamma'\right] \\
&= \gamma'
\end{aligned}$$

This shows that $\widetilde{\mathsf{S}}'$ is uniform across co-ordinates, and therefore that it is a uniform γ-approximate solver with success probability p. □

Proof of Claim 27. Let $\widetilde{\mathsf{S}}$ be the uniform γ-approximate solver for m-equation n-variable η-noisy k-LIN. We show how to amplify it into a γ'-approximate solver $\widetilde{\mathsf{S}}'$ for (rm)-equation k-LIN instances, where $r = 8\cdot\log(8/(1-\gamma'))/\gamma^2$.

The approximate solver $\widetilde{\mathsf{S}}'$ on input $(A, b = As + e)$ works as follows:

```
Divide $(A,b)$ into $r$ instances $(A_1,b_1),\dots,(A_r,b_r)$, where each $(A_j,b_j)$ has $m$ equations.
For each $j \in [r]$, compute $\tilde{s}_j \leftarrow \widetilde{\mathsf{S}}(A_j,b_j)$.
For each $i \in [n]$, set $\tilde{s}[i] = \mathrm{maj}_{j\in[r]}\tilde{s}_j[i]$.
Output $\tilde{s}$.
```

The γ-approximate solver $\widetilde{\mathsf{S}}$, owing to its uniformity over co-ordinates, gives us the following guarantee on each $\tilde{s}_j$ and any $i \in [n]$:

$$\Pr[\tilde{s}_j[i] = s[i] \mid s\cdot\tilde{s}_j \geq \gamma] \geq \frac{1}{2} + \frac{\gamma}{2}$$

For each $i \in [n]$, as $\tilde{s}[i] = \mathrm{maj}_{j\in[r]}\tilde{s}_j[i]$, by the above expression and the Chernoff bound, we can bound the probability that $\tilde{s}[i]$ is wrong, conditioned on all the $\tilde{s}_j$'s having γ-correlation with s as follows:

$$\Pr\left[\tilde{s}[i] \neq s[i] \;\middle|\; \forall j \in [r] : s\cdot\tilde{s}_j \geq \gamma\right] \leq e^{-r\gamma^2/8}$$

In our case, $r = 8\log(8/(1-\gamma'))/\gamma^2$. This gives us:

$$\Pr\left[\tilde{s}[i] \neq s[i] \;\middle|\; \forall j \in [r] : s \cdot \tilde{s}_j \geq \gamma\right] \leq \frac{1}{8} \cdot (1-\gamma')$$

Let $I_{\neq}$ be the set of $i \in [n]$ such that $\tilde{s}[i] \neq s[i]$. By linearity of expectation, we have:

$$\mathrm{E}\left[|I_{\neq}| \;\middle|\; \forall j \in [r] : s \cdot \tilde{s}_j \geq \gamma\right] \leq \frac{(1-\gamma')n}{8}$$

The correlation $s \cdot \tilde{s}$ is more than γ' exactly when $|I_{\neq}|$ is at most $n \cdot (1-\gamma')/2$. Thus, by the Markov bound, we have:

$$\Pr\left[s \cdot \tilde{s} \geq \gamma' \;\middle|\; \forall j \in [r] : s \cdot \tilde{s}_j \geq \gamma\right] = \Pr\left[|I_{\neq}| \leq n \cdot (1-\gamma')/2 \;\middle|\; \forall j \in [r] : s \cdot \tilde{s}_j \geq \gamma\right] \geq \frac{3}{4}$$

Accounting for the probability of the event being conditioned on above happening, we have the following success probability for $\widetilde{\mathsf{S}}'$:

$$\begin{aligned}\Pr[s \cdot \tilde{s} \geq \gamma'] &\geq \Pr\left[\forall j \in [r] : s \cdot \tilde{s}_j \geq \gamma\right] \cdot \Pr\left[s \cdot \tilde{s} \geq \gamma' \;\middle|\; \forall j \in [r] : s \cdot \tilde{s}_j \geq \gamma\right] \\ &\geq p^r \cdot \frac{3}{4}\end{aligned}$$

where the bound on the first probability in the product above uses the fact that $\widetilde{\mathsf{S}}$ is uniform across solutions, and so has success probability p for any arbitrary planted solution s. This completes the proof of the claim.

□

3.3 Correcting an approximate solution: Proof of Lemma 22

Before we give the proof of Lemma 22 we need to address one technicality. The k variables in a random k-LIN equation are sampled independently and with repetition, so a variable can occur in an equation several times. The matrix A only records the parity of the number of occurrences of each variable. For example, when $n = 3$, the left-hand side of both equations $x_1 + x_3 + x_1 = 0$ and $x_3 + x_3 + x_3 = 0$ is represented by the same row vector $(0, 0, 1)$. The latter representation was useful when proving Claim 26 but it will now be useful to revert to the former one. This amounts to reverse-sampling the missing variables in A and randomly reordering them. So we will assume, without loss of generality, that the k-LIN equation $x_{i_1} + \cdots + x_{i_k} = b$ is represented by the ordered tuple $(i_1, \ldots, i_k, b)$ where $i_1, \ldots, i_k$ are independent and uniform indices from $[n]$.

Proof of Lemma 22. To predict $s[1]$, the algorithm collects all equations in its input system in which variable x_1 appears. Such equations have the form $(i_1, \ldots, i_j = 1, \ldots, i_k, b)$, where j is the index of a random occurrence of variable x_1 (in case there are several). Using each such equation, the value of $s[1]$ is estimated by

$$\tilde{s}[1] = b + \hat{s}[i_1] + \cdots + \hat{s}[i_{j-1}] + \hat{s}[i_{j+1}] + \cdots + \hat{s}[i_k]$$

where $\hat{s}$ is the given approximate solution, and the majority value of these estimates is output. The procedure is repeated for s_2 up to s_n.

Conditioned on $i_j = 1$, the other indices are independent and uniform. Assuming b is a noisy solution, that is b equals $s[i_1] + \cdots + s[i_k]$ plus a noise bit e, the correlation between $s[1]$ and its estimator $\tilde{s}[1]$ is given by

$$\mathrm{E}[(-1)^{s[1]+\tilde{s}[1]}] = \mathrm{E}[(-1)^{(s[i_1]+\hat{s}[i_1])+\cdots+(s[i_{j-1}]+\hat{s}[i_{j-1}])+(s[i_{j+1}]+\hat{s}[i_{j+1}])+\cdots+(s[i_k]+\hat{s}[i_k])+e}]$$

where the second expression excludes i_j. By independence, this can be written as a product of expectations

$$\mathrm{E}[(-1)^{s[i_1]+\hat{s}[i_1]}] \cdots \mathrm{E}[(-1)^{s[i_{j-1}]+\hat{s}[i_{j-1}]}] \cdot \mathrm{E}[(-1)^{s[i_{j+1}]+\hat{s}[i_{j+1}]}] \cdots \mathrm{E}[(-1)^{s[i_k]+\hat{s}[i_k]}] \cdot \mathrm{E}[(-1)^e].$$

Each of the first $k-1$ terms equals γ and the noise contributes $1-2\eta$, so $s[1]$ and $\tilde{s}[1]$ have correlation $\kappa = (1-2\eta)\gamma^{k-1}$.

The probability that x_1 appears in any given equation is at least $1-(1-1/n)^k \geq k/2n$, so by a Chernoff bound the probability that x_1 appears in fewer than $km/4n$ of the equations is at most $\exp(-km/16n) = o(1/n)$. Assuming this isn't the case, the estimates arising from the different equations containing x_1 are independent, so as $km/4n \geq 10 \log n/\kappa^2$, by another Chernoff bound the probability that the majority estimate for $s[1]$ is incorrect is $o(1/n)$. By a union bound, the majority estimates for all n co-ordinates of s are correct except with probability $o(1)$. □

4 Bounds on list size

In this section we state and prove upper and lower bounds on the list size $\ell(\mu, \alpha)$. The upper bound in Corollary 5 follows from Theorem 3 and a counting argument. Proposition 28 gives a substantially tighter non-constructive upper bound in the regime $\alpha < \mu^2$. The lower bound of Proposition 29 in the regime $\alpha > \mu$ is proved by a volume argument. The lower bound in Proposition 30, which applies to the whole range of parameters, is obtained by analyzing a specific corrupted codeword.

4.1 Proof of Corollary 5

We show that the existence of an approximate list-decoder for a code of message length $|\Sigma|$ that succeeds with probability at least p implies $\ell \leq \ln 2 \cdot |\Sigma|/p$. Plugging in the value of p from Theorem 3 then gives Corollary 5.

Let *list* be the collection of outputs generated by $\ln 2 \cdot |\Sigma|/p$ independent runs of the approximate list-decoder L^G. If a codeword μ-correlates with G, the probability that it doesn't α-correlate with anything in *list* is at most $(1-p)^{\ln 2 \cdot |\Sigma|/p} < 2^{-|\Sigma|}$. Since there are at most $2^{|\Sigma|}$ codewords that μ-correlate with G, by a union bound there is a positive probability that *list* covers all of them.

4.2 Non-constructive upper bound in the regime $\alpha < \mu^2$

The following lemma, which is essentially the proof of the Johnson bound, gives a much tighter upper bound on list size than Corollary 5 in the regime $\alpha < \mu^2$.

Proposition 28. *For every $0 < \alpha < \mu^2$ and every binary code, $\ell(\mu, \alpha) \leq (1-\alpha)/(\mu^2 - \alpha)$.*

For example, $\ell(\mu, \mu^2/2) \leq 4/\mu^2$. In the case of the k-XOR code, Proposition 28 shows the existence of a list $F_1, \ldots, F_\ell$ of *messages* such that $\mathrm{E}[G \cdot F^k] \geq \mu$ implies $|\mathrm{E}[F \cdot F_i]| \geq \alpha^{1/k}$ for some $i \in [\ell]$ (since $\mathrm{E}[F^k \cdot F_i^k] = \mathrm{E}[F \cdot F_i]^k$).

Proof. Let ℓ be the maximal value for which there exists a list $C_1, \dots, C_\ell$ such that $\mathrm{E}[G \cdot C_i] \geq \mu$ for all $1 \leq i \leq \ell$ and $\mathrm{E}[C_i \cdot C_j] \leq \alpha$ for all $i \neq j$. Then for every $t \geq 0$,

$$\begin{aligned} 0 &\leq \mathrm{E}\big[(C_1 + \dots + C_\ell - tG)^2\big] \\ &\leq \sum_{i=1}^{\ell} \mathrm{E}[C_i^2] + \sum_{i \neq j} \mathrm{E}[C_i \cdot C_j] - 2t \sum_{i=1}^{\ell} \mathrm{E}[C_i \cdot G] + t^2 \,\mathrm{E}[G^2] \\ &\leq \ell + \ell(\ell-1)\delta - 2\ell\mu t + t^2 \end{aligned}$$

This is only possible if the discriminant $4\ell^2\mu^2 - 4(\ell + (\ell^2 - \ell)\alpha)$ (of the quadratic in t) is nonnegative, implying that $\delta \geq \mu^2$ or $\ell \leq (1-\alpha)/(\mu^2 - \alpha)$.

If $\mathrm{E}[G \cdot C] \geq \mu$ then $\mathrm{E}[C \cdot C_i^k]$ must be greater than α for some i, for otherwise the maximality of ℓ would be contradicted. □

4.3 Lower bound in the regime $\alpha > \mu$

Proposition 29. *For the k-XOR code, when $\mu < \alpha < 1$,*

$$\ell(\mu, \alpha) \geq \frac{4}{e\sqrt{1-\mu^{2/k}} \cdot |\Sigma|} \cdot \exp\big(\tfrac{1}{2}(\alpha^{2/k} - \mu^{2/k})|\Sigma|\big).$$

Let h denote the binary entropy function and $0 \leq \delta \leq 1$. We will use the following bounds on the volume of Hamming balls:

$$\binom{N}{\leq (1-\delta)N/2} \leq 2^{Nh(\frac{1-\delta}{2})} \tag{15}$$

$$\binom{N}{(1-\delta)N/2} \geq \frac{4}{e\sqrt{1-\delta^2 N}} 2^{Nh(\frac{1-\delta}{2})} \tag{16}$$

The following Taylor expansion is valid for all $-1 \leq \delta \leq 1$:

$$h\Big(\frac{1-\delta}{2}\Big) = 1 - \frac{1}{2\ln 2} \cdot \sum_{i=1}^{\infty} \frac{\delta^{2i}}{i(2i-1)}. \tag{17}$$

Proof of Proposition 29. Let $N = |\Sigma|$ and G be the constant function 1. The codewords $\mathcal{C}$ that are $(1-\mu)/2$-close to G are exactly those that encode messages of relative Hamming weight at most $(1-\mu^{1/k})/2$, so the number of such codewords is at least

$$|\mathcal{C}| \geq \binom{N}{(1-\mu^{1/k})N/2} \geq \frac{4}{e\sqrt{1-\mu^{2/k}} \cdot N} \cdot 2^{h\left(\frac{1-\mu^{1/k}}{2}\right)\cdot N},$$

by (16). On the other hand, the codewords that are $(1-\alpha)/2$-close to any given codeword $C_i = F_i^k$ are those that encode messages within Hamming distance at most $(1-\alpha^{1/k})/2$ from F_i, so there are at most $2^{h((1-\alpha^{1/k})/2)\cdot N}$ of them by (15). Therefore covering $\mathcal{C}$ requires a list of size

$$\frac{4}{e\sqrt{1-\mu^{2/k}} \cdot N} \cdot 2^{h\left(\frac{1-\mu^{1/k}}{2}\right)\cdot N - h\left(\frac{1-\alpha^{1/k}}{2}\right)\cdot N}.$$

Using the Taylor expansion (17), we can lower bound $h((1-\mu^{1/k})/2) - h((1-\alpha^{1/k})/2)$ by the difference of the leading terms in the summation, which equals $(\alpha^{2/k} - \mu^{2/k})/2\ln 2$, completing the proof. □

4.4 A general lower bound

Proposition 30. *For the k-XOR code, when $\mu \geq |\Sigma|^{-1/2}$, $\ell(\mu, \alpha) \geq \Omega(\alpha^{2/k}\mu^{-2})$, assuming $|\Sigma|$ is a power of two.*

Proof. We first assume that μ is equal to $|\Sigma|^{-1/2}$. Let Σ be a $\mathbb{F}_2$-vector space and $\mathcal{H} \subseteq \{-1, 1\}^{\Sigma}$ be the Hadamard code. Its codewords are the functions $H(x) = (-1)^{\langle a,x \rangle}$. The codewords of the k-wise tensor product $\mathcal{H}^k$ of $\mathcal{H}$ are given by $H^k(x_1, \dots, x_k) = H(x_1) \cdots H(x_k) = H(x_1 + \cdots + x_k)$. Thus the code $\mathcal{H}^k$ is isomorphic to $\mathcal{H}$ as a linear space.

Let G be the corrupted codeword $G(x_1, \dots, x_k) = B(x_1 + \cdots + x_k)$, where B is the bent function

$$B(z) = (-1)^{z_1 z_2 + \cdots + z_{t-1} z_t}, \qquad t = \log|\Sigma|.$$

For every codeword H of $\mathcal{H}$, $\mathrm{E}[GH^k] = \mathrm{E}[BH]$. The correlation of B with every linear function is identical up to sign, so by Parseval's identity $\mathrm{E}[BH]$ always equals $|\Sigma|^{-1/2}$ or $-|\Sigma|^{-1/2}$. After a possible change of sign in G we may assume that $\mathrm{E}[GH^k] \geq |\Sigma|^{-1/2}$ for at least half the codewords in $\mathcal{H}^k$. Since all these codewords also belong to the k-XOR code, there must exist a list $X_1^k, \dots, X_\ell^k$ of k-XOR codewords such that half the codewords in $\mathcal{H}$ $\alpha^{1/k}$-correlate to some X_i. Viewed as vectors in $\mathbb{R}^{\Sigma}$, the elements of $\mathcal{H}$ are orthonormal. By Pythagoras' theorem any X_i can $\alpha^{1/k}$-correlate with at most $\alpha^{-2/k}$ of them. It follows that $\ell = \Omega(|\Sigma| \cdot \alpha^{2/k})$.

When $\mu > |\Sigma|^{-1/2}$ we apply the argument to a dimension-$\lceil \log 1/\mu^2 \rceil$ quotient of the Hadamard code. □

5 Open Questions

The main coding-theoretic question left open is the dependence of the list size on the correlation μ at large distances for the k-XOR code. The upper bound in Corollary 5 is exponential in $\mu^{\Theta(1/k)}$, while the lower bound in Proposition 30 is proportional to $1/\mu^2$. A tensoring argument shows that $\ell_{2k}(\mu^k, \alpha^k) > \ell_2(\mu, \alpha)$, where ℓ_k is the list size for the k-XOR code. If, say, $\ell_2(\mu, \mu/2)$ were lower bounded by an exponential in $1/\mu$, an exponential lower bound certifying the optimality of Corollary 5 would follow. On the other hand, any improvement in the success probability in the decoder (and therefore the list size) would improve the success probability of our k-LIN reduction.

Regarding hardness versus randomness of k-LIN instances, one natural question is whether our sample-efficient reduction can be carried out for general k-CSP predicates. Such a reduction would relate the pseudorandomness and one-wayness of efficient local functions with small output length. Most of the techniques developed in this work apply to the more general setting. The only exception is Claim 26, which exploits the symmetry of the XOR predicate. Can this "uniformisation" of solvers be done for general predicates?

While our reduction is sample-efficient, it still incurs a loss of $O(m^{2/k})$. Is it possible to reduce this loss by reusing training data in different uses of the predictor? One intriguing possibility is suggested by our product sampler for regular functions. When predicting the answer to a fixed equation a, the predictor P takes as input m samples $(a_1, b_1), \dots, (a_m, b_m)$, and the bias of the predictor over all these samples is towards $\langle a, s \rangle$. So amplifying the probability of successful prediction is the same as estimating the bias of P. And if P were a regular function, we would be able to use our product sampler to reuse samples during amplification. While there is no reason to expect an arbitrary predictor to be regular, it might be possible to convert it into a regular one.

Finally, the success probability of the solver produced by our reduction becomes trivial for small values of k (that is, if $k \leq 6$ and $m = \Omega(n)$). Is it possible to perform meaningful solving-to-distinguishing reductions for k-LIN for such small values of k?

Acknowledgments

Andrej Bogdanov is supported by RGC GRF grant CUHK14238716. Manuel Sabin is supported by the NSF GRFP under Grant #DGE-1106400. Prashant Nalini Vasudevan was also supported by NSF Grant CNS-1350619, and by DARPA (IBM – DARPA Prime) under Agreement #4915012803.

Andrej thanks Siyao Guo, Sebastian Muller, Luca Trevisan, and Osamu Watanabe for helpful discussions.

References

[ABW10] Benny Applebaum, Boaz Barak, and Avi Wigderson. Public-key cryptography from different assumptions. In Leonard J. Schulman, editor, *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 171–180. ACM, 2010.

[AGS03] Adi Akavia, Shafi Goldwasser, and Shmuel Safra. Proving hard-core predicates using list decoding. In *44th Symposium on Foundations of Computer Science (FOCS 2003), 11-14 October 2003, Cambridge, MA, USA, Proceedings*, pages 146–157. IEEE Computer Society, 2003.

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

[AOW15] Sarah R. Allen, Ryan O'Donnell, and David Witmer. How to refute a random CSP. In Venkatesan Guruswami, editor, *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.

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

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

[App17] B. Applebaum. Exponentially-hard gap-csp and local prg via local hardcore functions. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, volume 00, pages 836–847, Oct. 2017.

[BGM$^+$16] Andrej Bogdanov, Siyao Guo, Daniel Masny, Silas Richelson, and Alon Rosen. On the hardness of learning with rounding over small modulus. In *Proceedings of the 13th Theory of Cryptography Conference (TCC)*, 2016. To appear.

[BLRL$^+$18] Shi Bai, Tancrède Lepoint, Adeline Roux-Langlois, Amin Sakzad, Damien Stehlé, and Ron Steinfeld. Improved security proofs in lattice-based cryptography: Using the rényi divergence rather than the statistical distance. *Journal of Cryptology*, 31(2):610–640, Apr 2018.

[BM16] Boaz Barak and Ankur Moitra. Noisy tensor completion via the sum-of-squares hierarchy. In Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir, editors, *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.

[BQ12] Andrej Bogdanov and Youming Qiao. On the security of Goldreich's one-way function. *Computational Complexity*, 21(1):83–127, 2012.

[CW04] Moses Charikar and Anthony Wirth. Maximizing quadratic programs: Extending grothendieck's inequality. In *45th Symposium on Foundations of Computer Science (FOCS 2004), 17-19 October 2004, Rome, Italy, Proceedings*, pages 54–60, 2004.

[Fei02] Uriel Feige. Relations between average case complexity and approximation complexity. In *Proceedings of the Thiry-fourth Annual ACM Symposium on Theory of Computing*, STOC '02, pages 534–543, New York, NY, USA, 2002. ACM.

[FPV15] Vitaly Feldman, Will Perkins, and Santosh Vempala. On the complexity of random satisfiability problems with planted solutions. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 77–86. ACM, 2015.

[GL89] Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In David S. Johnson, editor, *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washigton, USA*, pages 25–32. ACM, 1989.

[GNW11] Oded Goldreich, Noam Nisan, and Avi Wigderson. *On Yao's XOR-Lemma*, pages 273–301. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[GW95] Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42(6):1115–1145, November 1995.

[IJK09] Russell Impagliazzo, Ragesh Jaiswal, and Valentine Kabanets. Approximate list-decoding of direct product codes and uniform hardness amplification. *SIAM J. Comput.*, 39(2):564–605, 2009.

[IJKW10] Russell Impagliazzo, Ragesh Jaiswal, Valentine Kabanets, and Avi Wigderson. Uniform direct product theorems: Simplified, optimized, and derandomized. *SIAM J. Comput.*, 39(4):1637–1665, 2010.

[Imp95] Russell Impagliazzo. Hard-core distributions for somewhat hard problems. In *In 36th Annual Symposium on Foundations of Computer Science*, pages 538–545. IEEE, 1995.

[KMOW17] 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, 2017.

[Lev87] Leonid A. Levin. One way functions and pseudorandom generators. *Combinatorica*, 7(4):357–363, Dec 1987.

[MM11] Daniele Micciancio and Petros Mol. Pseudorandom knapsacks and the sample complexity of LWE search-to-decision reductions. In Phillip Rogaway, editor, *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings*, volume 6841 of *Lecture Notes in Computer Science*, pages 465–484. Springer, 2011.

[RRS17] 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, 2017.

[STV01] Madhu Sudan, Luca Trevisan, and Salil P. Vadhan. Pseudorandom generators without the XOR lemma. *J. Comput. Syst. Sci.*, 62(2):236–266, 2001.

[Yao82] Andrew Chi-Chih Yao. Theory and applications of trapdoor functions (extended abstract). In *23rd Annual Symposium on Foundations of Computer Science, Chicago, Illinois, USA, 3-5 November 1982*, pages 80–91. IEEE Computer Society, 1982.