Deterministic Approximate Counting for Degree-2 Polynomial Threshold Functions

Anindya De*
Institute for Advanced Study

Ilias Diakonikolas†
University of Edinburgh

Rocco A. Servedio‡
Columbia University

Abstract

We give a *deterministic* algorithm for approximately computing the fraction of Boolean assignments that satisfy a degree-2 polynomial threshold function. Given a degree-2 input polynomial $p(x_1, \ldots, x_n)$ and a parameter $\epsilon > 0$, the algorithm approximates

$$\mathbf{Pr}_{x \sim \{-1,1\}^n}[p(x) \geq 0]$$

to within an additive $\pm\epsilon$ in time $\mathrm{poly}(n, 2^{\mathrm{poly}(1/\epsilon)})$. Note that it is NP-hard to determine whether the above probability is nonzero, so any sort of multiplicative approximation is almost certainly impossible even for efficient randomized algorithms. This is the first deterministic algorithm for this counting problem in which the running time is polynomial in n for $\epsilon = o(1)$. For "regular" polynomials p (those in which no individual variable's influence is large compared to the sum of all n variable influences) our algorithm runs in $\mathrm{poly}(n, 1/\epsilon)$ time. The algorithm also runs in $\mathrm{poly}(n, 1/\epsilon)$ time to approximate $\mathbf{Pr}_{x \sim N(0,1)^n}[p(x) \geq 0]$ to within an additive $\pm\epsilon$, for any degree-2 polynomial p.

As an application of our counting result, we give a deterministic FPT multiplicative $(1 \pm \epsilon)$-approximation algorithm to approximate the k-th absolute moment $\mathbf{E}_{x \sim \{-1,1\}^n}[|p(x)^k|]$ of a degree-2 polynomial. The algorithm's running time is of the form $\mathrm{poly}(n) \cdot f(k, 1/\epsilon)$.

*anindya@math.ias.edu. Research supported by Templeton Foundation Grant 21674 and NSF CCF-1149843.
†ilias.d@ed.ac.uk. Some of this work was done while the author was at UC Berkeley supported by a Simons Fellowship.
‡rocco@cs.columbia.edu. Supported by NSF grants CNS-0716245, CCF-0915929, and CCF-1115703.

1 Introduction

A *degree-d polynomial threshold function* (PTF) is a Boolean function $f : \{-1,1\}^n \to \{-1,1\}$ defined by $f(x) = \mathrm{sign}(p(x))$ where $p : \{-1,1\}^n \to \mathbb{R}$ is a degree-d polynomial. In the special case where $d = 1$, degree-d PTFs are often referred to as *linear threshold functions* (LTFs) or *halfspaces*. While LTFs and low-degree PTFs have been researched for decades (see e.g., [MK61, MTT61, MP68, Mur71, GHR92, Orp92, Hås94, Pod09] and many other works) their study has recently received new impetus as they have played important roles in complexity theory [She08, She09, DHK+10, Kan10, Kan12c, Kan12a, KRS12], learning theory [KKMS08, SSSS11, DOSW11, DDFS12], voting theory [APL07, DDS12] and other areas.

An important problem associated with LTFs and PTFs is that of deterministically estimating the fraction of assignments that satisfy a given LTF or PTF over $\{-1,1\}^n$. In particular, in this paper we are interested in deterministically estimating the fraction of satisfying assignments for PTFs of degree $d = 2$. This problem is motivated by the long line of work on *deterministic approximate counting algorithms*, starting with the seminal work of Ajtai and Wigderson [AW85] who gave non-trivial deterministic counting algorithms for constant-depth circuits. Since then much progress has been made on the design of deterministic counting algorithms for other classes of Boolean functions like DNFs, low-degree $GF[2]$ polynomials and LTFs [LV96, GMR13, Vio09, GKM+11]. Problems of this sort can be quite challenging; after close to three decades of effort, deterministic polynomial time counting algorithms are not yet known for a simple class like polynomial-size DNFs.

Looking beyond Boolean functions, there has been significant work on obtaining deterministic approximate counting algorithms for combinatorial problems using ideas and techniques from statistical physics. This includes work on counting matchings [BGK+07], independent sets [Wei06], proper colorings [LLY13] and other problems in statistical physics [BG08]. We note that there is interest in obtaining such deterministic algorithms despite the fact that in some of these cases an optimal randomized algorithm is already known (e.g., for counting matchings [JSV01]) and the performance of the corresponding deterministic algorithm is significantly worse [BGK+07]. For this paper, the most relevant prior work are the results of Gopalan *et al.* and Stefankovic *et al.* [GKM+11] who independently obtained deterministic $\mathrm{poly}(n, 1/\epsilon)$ time algorithms for counting the satisfying assignments of an LTF up to $(1 \pm \epsilon)$ multiplicative error. (As we discuss later, in contrast with LTFs it is NP-hard to count the satisfying assignments of a degree-d PTF for any $d > 1$ up to any multiplicative factor. Thus, the right notion of approximation for degree-2 PTFs is additive error.)

There has recently been significant work in the literature on *unconditional derandomization* of LTFs and PTFs. The starting point of these works are the results of Rabani and Shpilka [RS09] and Diakonikolas et al [DGJ+09] who gave explicit constructions of polynomial-sized hitting sets and pseudorandom generators for LTFs. Building on these works, Meka and Zuckerman [MZ10] and subsequently Kane [Kan11a, Kan11b, Kan12b] constructed polynomial-sized PRGs for degree-d PTFs for $d > 1$. These PRGs trivially imply deterministic polynomial-time counting algorithms for any fixed d and fixed $\epsilon > 0$. While there has been significant research on improving the dependence of the size of these PRGs on ϵ, the best construction in the current state of the art is due to Kane [Kan12c] who gave an explicit PRG for degree-d polynomial threshold functions over $\{-1,1\}^n$ of size $n^{O_d(1)\cdot \mathrm{poly}(1/\epsilon)}$. (In a related but different line of work [Kan11a, Kan11b, Kan12b] focusing on PRGs for degree-d PTFs over the Gaussian $\mathcal{N}(0,1)^n$ distribution, the strongest result to date is that of [Kan12b] which for any constant degree d gives an explicit PRG of size $n^{f_d(1/\epsilon)}$ for degree-d PTFs; here $f_d(1/\epsilon)$ is a slightly sub-polynomial function of $1/\epsilon$, even for $d = 2$). As a consequence, the resulting deterministic counting algorithms have a running time which is at least $n^{O_d(1)\cdot \mathrm{poly}(1/\epsilon)}$ and thus the running time of these algorithms is not a *fixed* polynomial in n. In particular, for any $\epsilon = o(1)$, the running time of these algorithms is super-polynomial in n.

1.1 Our contributions. In this work we give the first *fixed* polynomial time deterministic algorithm for a PTF problem of this sort. As our main result, for all $\epsilon > 0$ we give a fixed $\mathrm{poly}(n)$-time algorithm to

deterministically $\pm\epsilon$-approximately count the number of satisfying assignments to a degree-2 PTF:

Theorem 1. *[Deterministic approximate counting of degree-2 PTFs over $\{-1,1\}^n$, informal statement] There is a deterministic algorithm which, given a degree-2 polynomial $q(x_1,\ldots,x_n)$ and $\epsilon > 0$ as input, runs in time $\mathrm{poly}(n, 2^{\tilde{O}(1/\epsilon^9)})$ and outputs a value $v \in [0,1]$ such that $\left|\mathbf{Pr}_{x\in\{-1,1\}^n}[q(x) \geq 0] - v\right| \leq \epsilon$.*

Note that, as a consequence of this theorem, we get a $\mathrm{poly}(n)$ time deterministic algorithm to count the fraction of satisfying assignments of a degree-2 PTF over $\{-1,1\}^n$ up to error $\epsilon = \tilde{O}(\log^{-1/9} n)$.

The *influence* of variable i on a polynomial $p : \{-1,1\}^n \to \mathbb{R}$, denoted $\mathrm{Inf}_i(p)$, is the sum of squares of all coefficients of p that are on monomials containing x_i; it is a measure of how much "effect" the variable i has on the outcome of p. Following previous work [DHK+10] we say that a polynomial p is *ϵ-regular* if $\max_{i\in[n]} \mathrm{Inf}_i(p) \leq \epsilon \cdot \sum_{j=1}^n \mathrm{Inf}_j(p)$. For sufficiently regular polynomials, our algorithm runs in fully polynomial time $\mathrm{poly}(n, 1/\epsilon)$:

Theorem 2. *[Deterministic approximate counting of regular degree-2 PTFs over $\{-1,1\}^n$, informal statement] Given $\epsilon > 0$ and an $O(\epsilon^9)$-regular degree-2 polynomial $q(x_1,\ldots,x_n)$ our algorithm runs (deterministically) in time $\mathrm{poly}(n, 1/\epsilon)$ and outputs a value $v \in [0,1]$ such that $\left|\mathbf{Pr}_{x\in\{-1,1\}^n}[q(x) \geq 0] - v\right| \leq \epsilon$.*

We note that the regular case has been a bottleneck in all known constructions of explicit PRGs for PTFs; the seed-length of known generators for regular PTFs is no better than for general PTFs. Given Theorem 2, the only obstacle to improved running times for deterministic approximate counting algorithms is improving the parameters of the "regularity lemma" which we use. [1]

Discussion. Our counting results described above give deterministic *additive* approximations to the desired probabilities. While additive approximation is not as strong as multiplicative approximation, we recall that the problem of determining whether $\mathbf{Pr}_{x\in\{-1,1\}^n}[q(x) \geq 0]$ is nonzero is well-known to be NP-hard for degree-2 polynomials even if all nonconstant monomials in q are restricted to have coefficient 1 (this follows by a simple reduction from Max-Cut, see the polynomial $q_{G,\mathrm{CUT}}$ defined below). Thus, no efficient algorithm, even allowing randomness, can give any multiplicative approximation to $\mathbf{Pr}_{x\sim\{-1,1\}^n}[q(x) \geq 0]$ unless $\mathbf{NP} \subseteq \mathbf{RP}$. Given this, it is natural to consider additive approximation.

Our results for degree-2 PTFs yield efficient deterministic algorithms for a range of natural problems. As a simple example, consider the following problem: Given an undirected n-node graph $G = ([n], E)$ and a size parameter k, the goal is to estimate the fraction of all 2^{n-1} cuts that contain at least k edges. (Recall that *exactly* counting the number of cuts of at least a given size is known to be #P-hard [Pap94].) We remark that a simple sampling-based approach yields an efficient *randomized* $\pm\epsilon$-additive approximation algorithm for this problem. Now note that the value of the polynomial $q_{G,\mathrm{CUT}}(x) = (|E| - \sum_{\{i,j\}\in E} x_i x_j)/2$ on input $x \in \{-1,1\}^n$ equals the number of edges in the cut corresponding to x (where vertices i such that $x_i = 1$ are on one side of the cut and vertices i such that $x_i = -1$ are on the other side). It is easy to see that if $|E| \geq C^9 n$ then $q_{G,\mathrm{CUT}}(x)$ must be $(1/C^9)$-regular. Theorem 2 thus provides a *deterministic* $\mathrm{poly}(n, 1/C)$-time algorithm that gives an $\pm O(1/C)$-additive approximation to the fraction of all cuts that have size at least k in n-node graphs with at least $C^9 n$ edges, and Theorem 1 gives a deterministic $\mathrm{poly}(n, 2^{\tilde{O}(1/\epsilon^9)})$-time $\pm\epsilon$-approximation algorithm for all n-node graphs.

As another example, consider the polynomial $q_{G,\mathrm{INDUCED}}(x) = \sum_{\{i,j\}\in E} \frac{1+x_i}{2} \cdot \frac{1+x_j}{2}$. In this case, we have that $q_{G,\mathrm{INDUCED}}(x)$ equals the number of edges in the subgraph of G that is induced by vertex set $\{i : x_i = 1\}$. Similarly to the example of the previous paragraph, Theorem 2 yields a deterministic $\mathrm{poly}(n, 1/C)$-time algorithm that gives a $\pm O(1/C)$-additive approximation to the fraction of all induced subgraphs that have at least k edges in n-node graphs with at least $C^9 n$ edges, and Theorem 1 gives a deterministic $\mathrm{poly}(n, 2^{\tilde{O}(1/\epsilon^9)})$-time $\pm\epsilon$-additive approximation algorithm for any graph.

[1] Indeed, Kane [Kan13] has suggested that using the notions of regularity and invariance from [Kan12c] may result in an improved, though still $2^{\mathrm{poly}(1/\epsilon)}$, running time for our approach; we have not explored that in this work.

Estimating moments. Our results also imply deterministic fixed-parameter tractable (FPT) algorithms for approximately computing moments of degree-2 polynomials. Consider the following computational problem ABSOLUTE-MOMENT-OF-QUADRATIC: given as input a degree-2 polynomial $q(x_1, \ldots, x_n)$ and an integer parameter $k \geq 1$, output the value $\mathbf{E}_{x \in \{-1,1\}^n}[|q(x)|^k]$. It is clear that the *raw* moment $\mathbf{E}[q(x)^k]$ can be computed exactly in $n^{O(k)}$ time by expanding out the polynomial $q(x)^k$, performing multilinear reduction, and outputting the constant term. Since the k-th raw moment equals the k-th absolute moment when k is even, this gives an $n^{O(k)}$ time algorithm for ABSOLUTE-MOMENT-OF-QUADRATIC for even k. However, for any fixed odd $k \geq 1$ the ABSOLUTE-MOMENT-OF-QUADRATIC problem is #P-hard (see Appendix B). Thus, it is natural to seek approximation algorithms for this problem.

Using the hyper-contractive inequality [Bon70, Bec75] it can be shown that the natural randomized algorithm – draw uniform points from $\{-1, 1\}^n$ and use them to empirically estimate $\mathbf{E}_{x \in \{-1,1\}^n}[|q(x)|^k]$ – with high probability gives a $(1 \pm \epsilon)$-accurate estimate of the k-th absolute moment of q in $\mathrm{poly}(n, 2^{k \log k}, 1/\epsilon)$ time. Using Theorem 1 we are able to derandomize this algorithm and obtain a *deterministic* FPT $(1 \pm \epsilon)$-multiplicative approximation algorithm for ABSOLUTE-MOMENT-OF-QUADRATIC:

Theorem 3. *There is a* deterministic *algorithm which, given any degree-*2 *polynomial $q(x_1, \ldots, x_n)$ with b-bit integer coefficients, any integer $k \geq 1$, and any $\epsilon \in (0, 1)$, runs in* $\mathrm{poly}\left(n, b, 2^{\tilde{O}((k \log k \log(1/\epsilon))^{9k}/\epsilon^9)}\right)$ *time and outputs a value $v \in \left[(1-\epsilon)\,\mathbf{E}_{x \in \{-1,1\}^n}[|q(x)|^k], (1+\epsilon)\,\mathbf{E}_{x \in \{-1,1\}^n}[|q(x)|^k]\right]$ that multiplicatively $(1 \pm \epsilon)$-approximates the k-th absolute moment of q.*

1.2 Techniques. The major technical work in this paper goes into proving Theorem 2. Given Theorem 2, we use a (deterministic) algorithmic version of the "regularity lemma for degree-d PTFs" from [DSTW10] to reduce the case of general degree-2 PTFs to that of regular degree-2 PTFs. (The regularity lemma that is implicit in [HKM09] can also be used for this purpose.)

As is usual in this line of work, we can use the invariance principle of Mossel *et al.* [MOO10] to show that for an $O(\epsilon^9)$-regular degree-2 polynomial $p : \mathbb{R}^n \to \mathbb{R}$, we have $\left|\mathbf{Pr}_{x \in \{-1,1\}^n}[p(x) \geq 0] - \mathbf{Pr}_{x \in \mathcal{N}(0,1)^n}[p(x) \geq 0]\right| \leq \epsilon$. Thus, to prove Theorem 2, we are left with the task of additively estimating $\mathbf{Pr}_{x \in \mathcal{N}(0,1)^n}[p(x) \geq 0]$.

The first conceptual idea towards achieving the aforementioned task is this: Since Gaussian distributions are invariant under rotations, computing the probability of interest $\mathbf{Pr}_{x \in \mathcal{N}(0,1)^n}[p(x) \geq 0]$ is equivalent to computing $\mathbf{Pr}_{x \in \mathcal{N}(0,1)^n}[\tilde{p}(x) \geq 0]$ for a "decoupled" polynomial $\tilde{p}$. More precisely, there exists a polynomial $\tilde{p} : \mathbb{R}^n \to \mathbb{R}$ of the form $\tilde{p}(x) = \sum_{i=1}^n \lambda_i x_i^2 + \sum_{i=1}^n \mu_i x_i + C$ such that the distributions of $p(x)$ and $\tilde{p}(x)$ (where $x \sim \mathcal{N}(0,1)^n$) are identical. Indeed, consider the symmetric matrix A associated with the quadratic part of $p(x)$ and let $Q^T \cdot A \cdot Q = \Lambda$ be the spectral decomposition of A. It is easy to show that $\tilde{p}(x) = p((Q \cdot x)_1, \ldots, (Q \cdot x)_n)$ is a decoupled polynomial with the same distribution as $p(x)$, $x \sim \mathcal{N}(0,1)^n$. The counting problem for $\tilde{p}$ should intuitively be significantly easier since there are no correlations between $\tilde{p}$'s monomials, and hence it would be useful if $\tilde{p}$ could be efficiently exactly obtained from p. Strictly speaking, this is not possible, as one cannot obtain the exact spectral decomposition of a symmetric matrix A (for example, A can have irrational eigenvalues). For the sake of this informal discussion, we assume that one can in fact obtain the exact decomposition and hence the polynomial $\tilde{p}(x)$.

Suppose we have obtained the decoupled polynomial $\tilde{p}(x)$. The second main idea in our approach is the following: We show that one can efficiently construct a t-variable "junta" polynomial $q : \mathbb{R}^t \to \mathbb{R}$, with $t = \mathrm{poly}(1/\epsilon)$, such that the distribution of $q(x)$ is $O(\epsilon)$-close to the distribution of $\tilde{p}(x)$ in Kolmogorov distance. (Recall that the Kolmogorov distance between two random variables is the maximum distance between their CDFs.) To prove this, we use a powerful recent result of Chatterjee [Cha09] (Theorem 42), proved using Stein's method, which provides a *central limit theorem* for functions of Gaussians. Informally, this CLT says that for any function $F : \mathbb{R}^n \to \mathbb{R}$ satisfying some technical conditions, if $g_1, \ldots, g_n$ are independent $\mathcal{N}(0,1)$ random variables, then $F(g_1, \ldots, g_n)$ is close in total variation distance (ℓ_1 distance between the pdfs) to a Gaussian distribution with the "right" mean and variance. (We refrain from giving a

more detailed description of the theorem here as the technical conditions stem from considering generators of the Ornstein-Uhlenbeck process, thus rendering it somewhat unsuitable for an intuitive discussion.) Using this result, we show that if $\max_i \lambda_i^2 \leq \epsilon^2 \cdot \mathrm{Var}(\tilde{p})$ (i.e., if the maximum magnitude eigenvalue of the symmetric matrix A corresponding to p is "small"), then the distribution of $\tilde{p}(x)$ (hence, also of $p(x)$) is $O(\epsilon)$-close to $\mathcal{N}(\mathbf{E}[\tilde{p}], \mathrm{Var}(\tilde{p}))$, and hence the one-variable polynomial $q(x) = \sqrt{\mathrm{Var}(\tilde{p})}x_1 + \mathbf{E}[\tilde{p}]$ is the desired junta. In the other case, i.e., the case that $\max_i \lambda_i^2 > \epsilon^2 \cdot \mathrm{Var}(\tilde{p})$, one must resort to a more involved approach as described below.

If $\max_i \lambda_i^2 > \epsilon^2 \cdot \mathrm{Var}(\tilde{p})$, we perform a "critical index based" case analysis (in the style of Servedio, see [Ser07]) appropriately tailored to the current setting. We remark that such analyses have been used several times in the study of linear and polynomial threshold functions (see e.g., [DGJ$^+$09, FGRW09, DHK$^+$10, DSTW10]). In all these previous works the critical index analysis was performed on *influences* of variables in the original polynomial (or linear form). Two novel aspects of the analysis in the current work are that (i) we must transform the polynomial from its original form into the "decoupled" version before carrying out the critical index analysis; and (ii) in contrast to previous works, we perform the critical index analysis not on the influences of variables, but rather on the *eigenvalues* of the quadratic part of the decoupled polynomial, i.e., on the values $(|\lambda_1|, \ldots, |\lambda_n|)$, ignoring the linear part of the decoupled polynomial. The following paragraph explains the situation in detail.

Suppose that the eigenvalues are ordered so that $|\lambda_1| \geq \ldots \geq |\lambda_n|$. Consider the polynomials $\tilde{p}_{H,i}(x) = C + \sum_{j \leq i}(\lambda_j x_j^2 + \mu_j x_j)$ (the "head part") and $\tilde{p}_{T,i}(x) = \sum_{j>i}(\lambda_j x_j^2 + \mu_j x_j)$ (the "tail part"). Define the τ-critical index as the minimum $\ell \in [n]$ such that $|\lambda_\ell| / \sqrt{\mathrm{Var}(\tilde{p}_{T,\ell-1})} \leq \tau$. Let $K_0 = \Theta(\tau^{-2}\log(1/\tau))$. If the τ-critical index is more than K_0 then we show that the "head part" $\tilde{p}_{H,K_0}(x)$ (appropriately shifted) captures the distribution of $\tilde{p}(x)$ up to a small error. In particular, the distribution of $q(x) = \tilde{p}_{H,K_0}(x) + \mathbf{E}[\tilde{p}_{T,K_0}(x)]$ is $O(\sqrt{\tau})$-close to that of $\tilde{p}$ in Kolmogorov distance. On the other hand, if the critical index is $K \leq K_0$, then it follows from Chatterjee's CLT that the polynomial $q(x) = \tilde{p}_{H,K}(x) + \sqrt{\mathrm{Var}(\tilde{p}_{T,K}(x))}x_{K+1} + \mathbf{E}[\tilde{p}_{T,K}(x)]$ is $O(\tau)$-close to $\tilde{p}$ in total variation distance (hence, also in Kolmogorov distance). Note that in both cases, $q(x)$ has at most $K_0 + 1$ variables and hence setting $\tau = \Theta(\epsilon^2)$, we obtain a polynomial $q(x)$ on $t \leq K_0 + 1 = \mathrm{poly}(1/\epsilon)$ variables whose distribution is Kolmogorov $O(\epsilon)$-close to that of $\tilde{p}(x)$.

Thus, we have effectively reduced our initial task to the deterministic approximate computation of $\mathbf{Pr}_{x \sim \mathcal{N}(0,1)^{K_0+1}}[q(x) \geq 0]$. This task can potentially be achieved in a number of different ways (see the discussion at the start of Section 2.4); with the aim of giving a self-contained and $\mathrm{poly}(1/\epsilon)$-time algorithm, we take a straightforward approach based on dynamic programming. To do this, we start by discretizing the random variable $\mathcal{N}(0,1)$ to obtain a distribution $D_{\mathcal{N}}$ (supported on $\mathrm{poly}(1/\epsilon)$ many points) which is such that $\left|\mathbf{Pr}_{x \sim \mathcal{N}(0,1)^{K_0+1}}[q(x) \geq 0] - \mathbf{Pr}_{x \sim D_{\mathcal{N}}^{K_0+1}}[q(x) \geq 0]\right| \leq \epsilon$. Since $q(x)$ is a decoupled polynomial, computing $\mathbf{Pr}_{x \sim D_{\mathcal{N}}^{K_0+1}}[q(x) \geq 0]$ can be reduced to the counting version of the knapsack problem where the weights are integers of magnitude $\mathrm{poly}(1/\epsilon)$, and therefore can be solved exactly in time $\mathrm{poly}(1/\epsilon)$ by standard dynamic programming.

Remark 4. *We note that the dynamic programming approach we employ could be used to do deterministic approximate counting for a decoupled n-variable Gaussian degree-2 polynomial $\tilde{p}(x)$ in $\mathrm{poly}(n, 1/\epsilon)$ time even without the junta condition. However, the fact that $\tilde{p}$ is Kolmogorov-close to a junta polynomial q is a structural insight which has already proved useful in followup work. Indeed, achieving a junta structure is absolutely crucial for recent extensions of this result [DDS13, DS13] which generalize the deterministic approximate counting algorithm presented here (to juntas of degree-2 PTFs in [DDS13] and to general degree-d PTFs in [DS13], respectively).*

Singular Value Decomposition: The above informal description glossed over the fact that given a matrix A, it is in general not possible to exactly represent the SVD of A using a finite number of bits (let alone to exactly compute the SVD in polynomial time). In our actual algorithm, we have to deal with the fact

that we can only achieve an "approximate" SVD. We define a notion of approximation that is sufficient for our purposes and show that such an approximation can be computed efficiently. Our basic algorithmic primitive is (a variant of the) well-known "powering method" (see [Vis13] for a nice overview). Recall that the powering method efficiently computes an approximation to the eigenvector corresponding to the highest magnitude eigenvalue. In particular, the method has the following guarantee: given that the largest-magnitude eigenvalue of A has absolute value $|\lambda_{\max}(A)|$, the powering method runs in time $\mathrm{poly}(n, 1/\kappa)$ and returns a unit vector w such that $\|A \cdot w\|_2 \geq |\lambda_{\max}(A)| \cdot (1 - \kappa)$.

For our purposes, we require an additional criterion: namely, that the vector $A \cdot w$ is almost parallel to w. (This corresponds to the notion of "decoupling" the polynomial discussed earlier.) It can be shown that if one naively applies the powering method, then it is not necessarily the case that the vector w it returns will also satisfy this requirement. To get around this, we modify the matrix A before applying the powering method and show that the vector w so returned provably satisfies the required criterion, i.e., $A \cdot w$ is almost parallel to w. An additional caveat is that the standard "textbook" version of the method is a randomized algorithm, and we of course need a deterministic algorithm. This can be handled by a straightforward derandomization, resulting in only a linear time overhead.

1.3 Organization. We record basic background facts from linear algebra, probability, and analysis in Appendix A, along with our new extended notion of the "critical index" of a pair of sequences. Section 2 establishes our main technical result – an algorithm for deterministically approximately counting satisfying assignments of a degree-2 PTF under the Gaussian $\mathcal{N}(0,1)^n$ distribution. Section 3 extends this result to satisfying assignments over $\{-1,1\}^n$. Finally, in Section 4 we give the application to deterministic approximation of absolute moments.

2 Deterministic approximate counting for Gaussian distributions

2.1 Intuition. Our goal is to compute, up to an additive $\pm\epsilon$, the probability $\mathbf{Pr}_{x\sim\mathcal{N}(0,1)^n}[p(x) \geq 0]$. The algorithm has two main stages. In the first stage (Section 2.3) we transform the original n-variable degree-2 polynomial p into an essentially equivalent polynomial q with a "small" number of variables – independent of n – and a nice special form (a degree-2 polynomial with no "cross terms"). The key algorithmic tool used in this transformation is the routine APPROXIMATE-DECOMPOSE which is described and analyzed in Section 2.2. In particular, suppose that the original degree–2 polynomial is of the form $p(x) = \sum_{i<j} a_{i,j}x_ix_j + \sum_i b_ix_i + C = x^TAx + b^Tx + C$. The first stage constructs a degree-2 "junta" polynomial $q(y_1, \ldots, y_K) : \mathbb{R}^K \to \mathbb{R}$ with no cross terms (that is, every non-constant monomial in q is either of the form y_i or y_i^2) where $K = \tilde{O}(1/\epsilon^4)$, such that $|\mathbf{Pr}_{x\sim\mathcal{N}(0,1)^n}[p(x) \geq 0] - \mathbf{Pr}_{y\sim\mathcal{N}(0,1)^K}[q(y) \geq 0]| \leq \epsilon$. Theorem 27 summarizes what is accomplished in the first stage. We view this stage as the main contribution of the paper.

In the second stage (Section 2.4) we give an efficient deterministic algorithm to approximately count the fraction of satisfying assignments for q. Our algorithm exploits both the fact that q depends on only $\mathrm{poly}(1/\epsilon)$ variables and the special form of q. Theorem 43 summarizes what is accomplished in the second stage. Theorem 50 combines these two results and gives our main result for deterministic approximate counting of Gaussian degree-2 PTFs.

The first stage: Constructing a degree-2 junta PTF. To implement the first step we take advantage of the fact that $x \sim \mathcal{N}(0,1)^n$ in order to "decouple" the variables. Suppose we have computed the spectral decomposition of A as $A = Q\Lambda Q^T$. (We remark that our algorithm does not compute this decomposition explicitly; rather, it iteratively approximates the eigenvector corresponding to the largest magnitude eigenvalue of A, as is described in detail in the pseudocode of algorithm `Construct-Junta-PTF`. For the

sake of this intuitive explanation, we assume that we construct the entire spectrum.) Then, we can write p as

$$p(y) = y^T \Lambda y + \mu^T y + C = \sum_{i=1}^{n} \lambda_i y_i^2 + \sum_{i=1}^{n} \mu_i y_i + C,$$

where $y = Q^T x$ and $\mu = Q^T b$. Since Q is orthonormal, it follows that $y \sim \mathcal{N}(0,1)^n$ and that the desired probability can be equivalently written as $\mathbf{Pr}_{y \sim \mathcal{N}(0,1)^n}[p(y) \geq 0]$.

At this point, let us arrange the variables in order, so that the sequence $|\lambda_1|, \ldots, |\lambda_n|$ is non-increasing. We now consider the ϵ-critical index of the pair of sequences $\{\lambda_i^2\}_{i=1}^n$ and $\{\mu_i^2\}_{i=1}^n$ (here $\{\mu_i^2\}_{i=1}^n$ is the "auxiliary sequence"see Definition 71). The starting point of our analysis is the following result.

Informal theorem: If the ϵ-critical index is zero, then the random variable $p(y)$, where $y \sim \mathcal{N}(0,1)^n$, is $O(\sqrt{\epsilon})$-close in total variation distance to $\mathcal{N}(\nu, \sigma^2)$ where $\nu = \mathbf{E}[p(y)]$ and $\sigma^2 = \mathrm{Var}[p(y)]$.

As mentioned earlier, the proof of the above theorem uses a recent result of Chatterjee [Cha09] (Theorem 42) which provides a central limit theorem for functions of Gaussians. With this as starting point, we consider a case analysis depending on the value of the ϵ-critical index of the pair of sequences $\{\lambda_i^2\}_{i=1}^n$ and $\{\mu_i^2\}_{i=1}^n$ ($\{\mu_i^2\}_{i=1}^n$ is the auxiliary sequence). Let K be the value of the the ϵ-critical index of the pair. If $K \leq K_0 \stackrel{\text{def}}{=} \tilde{O}(1/\epsilon^2)$, then the tail $p_{T,K}(y) = \sum_{j>K}(\lambda_j y_j^2 + \mu_j y_j)$ can be replaced by $\mathcal{N}(\nu_j, \sigma_j^2)$ where $\nu_j = \mathbf{E}[p_{T,K}(y)]$ and $\sigma^2 = \mathrm{Var}[p_{T,K}(y)]$. On the other hand, if $K \geq K_0$, then the distribution of $q(y) = \sum_{j \leq K_0}(\lambda_j y_j^2 + \mu_j y_j) + C$ differs from the distribution of $p(y)$ by $O(\sqrt{\epsilon})$ in Kolmogorov distance. In either case, we end up with a degree-2 polynomial on at most $K_0 + 1 = \tilde{O}(1/\epsilon^2)$ variables whose distribution is $O(\sqrt{\epsilon})$ close to the distribution of $p(y)$ in Kolmogorov distance.

The main difficulty in the real algorithm and analysis vis-a-vis the idealized version described above is that computationally, it is not possible to compute the exact spectral decomposition. Rather, what one can achieve is some sort of an approximate decomposition (we are deliberately being vague here about the exact nature of the approximation that can be achieved). Roughly speaking, at every stage of the algorithm constructing q several approximations are introduced and non-trivial technical work is required in bounding the corresponding error. See Sections 2.2 and 2.3 for the detailed analysis.

The second stage: Counting satisfying assignments of degree-2 juntas over Gaussian variables. We are now left with the task of (approximately) counting $\mathbf{Pr}[q(y) \geq 0]$. To do this we start by discretizing each normal random variable y_i to a sufficiently fine granularity – it turns out that a grid of size $\mathrm{poly}(1/\epsilon)$ suffices. Let us denote by $\tilde{y}_i$ the discretized approximation to y_i. We also round the coefficients of q to a suitable $\mathrm{poly}(\epsilon)$ granularity and denote by q' the rounded polynomial. It can be shown that $q(y)$ and $q'(\tilde{y})$ are ϵ-close in Kolmogorov distance. Finally, this reduces computing $\mathbf{Pr}[q(y) \geq 0]$ to computing $\mathbf{Pr}[q'(\tilde{y}) \geq 0]$. Since the terms in q' are decoupled (i.e., there are no cross terms) and have small integer coefficients, $q'(\tilde{y})$ can be expressed as a read-once branching program of size $\mathrm{poly}(1/\epsilon)$. Using dynamic programming, one can efficiently compute the exact probability $\mathbf{Pr}[q'(\tilde{y}) \geq 0]$ in time $\mathrm{poly}(1/\epsilon)$. See Section 2.4 for the details.

We note that alternative algorithmic approaches could potentially be used for this stage. We chose our approach of discretizing and using dynamic programming because we feel that it is intuitive and self-contained and because it easily gives a $\mathrm{poly}(1/\epsilon)$-time algorithm for this stage.

2.2 A useful algorithmic primitive. In this section we state and prove correctness of the main algorithmic primitive APPROXIMATE-DECOMPOSE that our procedure for constructing a degree-2 junta over Gaussian variables will use. This primitive partially "decouples" a given input degree-2 polynomial by transforming the polynomial into an (essentially equivalent) polynomial in which a new variable y (intuitively corresponding to the largest eigenvector of the input degree-2 polynomial's matrix) essentially does not appear together with any other variables in any monomials.

Theorem 6 gives a precise statement of APPROXIMATE-DECOMPOSE's performance. The reader who is eager to see how APPROXIMATE-DECOMPOSE is used may wish to proceed directly from the statement of Theorem 6 to Section 2.3.

We require the following definition to state Theorem 6. (Below a "normalized linear form" is an expression $\sum_{i=1}^{n} w_i x_i$ with $\sum_{i=1}^{n} w_i^2 = 1$.)

Definition 5. *Given a degree-2 polynomial $p : \mathbb{R}^n \to \mathbb{R}$ defined by $p(x) = \sum_{1 \le i \le j \le n} a_{ij} x_i x_j + \sum_{1 \le i \le n} b_i x_i + C$ and a normalized linear form $L_1(x)$, we define the* residue of p with respect to $L_1(x)$*, $Res(p, L_1(x))$, to be the polynomial obtained by the following process : For each $i \in [n]$, express x_i as $\alpha_{i1} L_1(x) + R_i(x)$ where $R_i(x)$ is orthogonal to the linear form $L_1(x)$. Now, consider the polynomial $q(y_1, x) = p(\alpha_{11} y_1 + R_1(x), \ldots, \alpha_{n1} y_1 + R_n(x))$. $Res(p, L_1(x))$ is defined as the homogenous multilinear degree-2 part of $q(y_1, x)$ which has the variable y_1 present in all its terms.*

Theorem 6. *Let $p : \mathbb{R}^n \to \mathbb{R}$ be a degree-2 polynomial (with constant term 0) whose entries are b-bit integers and let $\epsilon, \eta > 0$. There exists a deterministic algorithm APPROXIMATE-DECOMPOSE which on input an explicit description of p, ϵ and η runs in time $\mathrm{poly}(n, b, 1/\epsilon, 1/\eta)$ and has the following guarantee :*

(a) If $\lambda_{\max}(p) \ge \epsilon \sqrt{\mathrm{Var}(p)}$, then the algorithm outputs rational numbers λ_1, μ_1 and a degree-2 polynomial $r : \mathbb{R}^{n+1} \to \mathbb{R}$ with the following property: for $(y, x_1, \ldots, x_n) \sim \mathcal{N}(0,1)^{n+1}$, the two distributions $p(x_1, \ldots, x_n)$ and $q(y_1, x_1, \ldots, x_n)$ are identical, where $q(y_1, x_1, \ldots, x_n)$ equals $\lambda_1 y_1^2 + \mu_1 y_1 + r(y_1, x_1, \ldots, x_n)$. Further, $\mathrm{Var}(Res(r, y_1)) \le 4\eta^2 \mathrm{Var}(p)$ and $\mathrm{Var}(r) \le (1 - \epsilon^4/40) \cdot \mathrm{Var}(p)$.

(b) If $\lambda_{\max}(p) < \epsilon \sqrt{\mathrm{Var}(p)}$, then the algorithm either outputs "small max eigenvalue" or has the same guarantee as (a).

In the rest of Section 2.2 we prove Theorem 6, but first we give some high-level intuition. Recall from the introduction that we would like to compute the SVD of the symmetric matrix corresponding to the quadratic part of the degree-2 polynomial p, but the exact SVD is hard to compute. APPROXIMATE-DECOMPOSE works by computing an approximation to the largest eigenvalue-eigenvector pair, and using the approximate eigenvector to play the role of L_1 in Definition 5.

The case that is of most interest to us is when the largest eigenvalue has large magnitude compared to the square root of the variance of p (since we will use Chatterjee's theorem to deal with the complementary case) so we focus on this case below. For this case, part (a) of Theorem 6 says that the algorithm outputs a degree-2 polynomial $q(y_1, x_1, \ldots, x_n)$ with the same distribution as p. Crucially, in this polynomial q, the first variable y_1 is "approximately decoupled" from the rest of the polynomial, namely r (because $\mathrm{Var}(Res(r, y_1))$ is small), and moreover $\mathrm{Var}(r)$ is substantially smaller than $\mathrm{Var}(p)$ (this is important because intuitively it means we have "made progress" on the polynomial p). Note that if we were given the exact eigenvalue-eigenvector pair corresponding to the largest magnitude eigenvalue, it would be possible to meet the conditions of case (a) with $\eta = 0$.

While approximating the largest eigenvector is a well-studied problem, we could not find any off-the-shelf solution with the guarantees we required. APPROXIMATE-DECOMPOSE adapts the well-known powering method for finding the largest eigenvector to give the desired guarantees.

2.2.1 Decomposing a matrix. In order to describe the APPROXIMATE-DECOMPOSE algorithm we first need a more basic procedure which we call APPROXIMATE-LARGEST-EIGEN. Roughly speaking, given a real symmetric matrix A with a large-magnitude eigenvalue, the APPROXIMATE-LARGEST-EIGEN procedure outputs approximations of the largest-magnitude eigenvalue and its corresponding eigenvector. Theorem 7 gives a precise performance guarantee:

Theorem 7. *Let $A \in \mathbb{R}^{n\times n}$ be a symmetric matrix whose entries are b-bit integers (not all 0) and $\epsilon, \eta > 0$ be given rational numbers. There exists a deterministic algorithm* APPROXIMATE-LARGEST-EIGEN *which on input A, ϵ and η, runs in time $\mathrm{poly}(n, b, 1/\epsilon, 1/\eta)$ and has the following behavior:*

(a) If $|\lambda_{\max}(A)| \geq \epsilon\|A\|_F$, the algorithm outputs a number $\tilde{\lambda} \in \mathbb{R}_+$ and unit vector $\tilde{w} \in \mathbb{R}^n$ such that

(i) $(1-\eta)|\lambda_{\max}(A)| \leq |\tilde{\lambda}| \leq |\lambda_{\max}(A)|$;

(ii) the matrix $\tilde{B} = A - \tilde{\lambda}(\tilde{w}\tilde{w}^T)$ satisfies $\|\tilde{B}\tilde{w}\|_2 < \eta\|A\|_F$; and

(iii) $\|\tilde{B}\|_F \leq (1-\epsilon^2/40)\cdot\|A\|_F$.

(b) If $|\lambda_{\max}(A)| < \epsilon\|A\|_F$, the algorithm either outputs "small max eigenvalue" or behaves as in case (a).

Let us describe the APPROXIMATE-LARGEST-EIGEN algorithm. Let $2^{-m} \leq \epsilon \leq 2^{-m+1}$. The running time of the algorithm will have a polynomial dependence on 2^m . Without loss of generality, assume that $\lambda_{\max}(A)$ is a positive number. Instead of working directly with the matrix A, we will work with the matrix $A' = A + t\cdot I$ where $t = \lceil\|A\|_F\rceil$. Note that an eigevector-eigenvalue pair (v, λ) of A maps to the pair $(v, \lambda + t)$ for A'.

For $\delta = \min\{\epsilon^4/100, \eta^4/10^8\}$, the APPROXIMATE-LARGEST-EIGEN algorithm works as follows :

- For unit vectors $e_1, \ldots, e_n$ and $k = \lceil\frac{1}{2\delta}\cdot\log(9n/4)\rceil$, the algorithm computes

$$\mu_i = \frac{\|A'\cdot(A'^k\cdot e_i)\|_2^2}{\|(A'^k\cdot e_i)\|_2^2}.$$

- Let $i^* = \arg\max_{i\in[n]}\mu_i$, and define

$$w = \frac{A'^k\cdot e_{i^*}}{\|A'^k\cdot e_{i^*}\|_2}, \quad \lambda = \|A\cdot w\|_2.$$

  Note that since w can have irrational entries, exact computation of w and λ is not feasible. However, in time $\mathrm{poly}(1/\delta, b, n)$, we can compute a unit vector $\tilde{w}$ so that $\|w-\tilde{w}\|_2 \leq \mathrm{poly}(\delta, 1/b, 1/n)$. Define $\tilde{\lambda}$ as $\|A\cdot\tilde{w}\|_2$ rounded to a precision $\mathrm{poly}(\delta, 1/b, 1/n)$. It is easy to see that $|\tilde{\lambda}-\lambda| \leq \mathrm{poly}(\delta, 1/b, 1/n)$.

- If $\tilde{\lambda}^2 \geq (1-9\cdot\delta^{1/4})\cdot\epsilon^2\cdot\|A\|_F^2$, then output the pair $(\tilde{w}, \tilde{\lambda})$. Else, output "small max eigenvalue".

Proof of Theorem 7: We start with the following simple claim:

Claim 8. *If $|\lambda_{\max}(A)| \leq (\epsilon/2)\cdot\|A\|_F$, then* APPROXIMATE-LARGEST-EIGEN *outputs "small max eigenvalue".*

Proof. Note that if $|\lambda_{\max}(A)| \leq (\epsilon/2)\cdot\|A\|_F$, then $\tilde{\lambda}^2 \leq \epsilon^2\|A\|_F^2/4$. By our choice of δ we have that $\tilde{\lambda}^2 < (1-9\cdot\delta^{1/4})\cdot\epsilon^2\cdot\|A\|_F^2$, hence the algorithm will output "small max eigenvalue". □

Next let us recall the "powering method" to compute the largest eigenvalue of a symmetric matrix. See the monograph by Vishnoi [Vis13] (the following statement is implicit in Lemma 8.1).

Lemma 9. *Let $A \in \mathbb{R}^{n\times n}$ be a symmetric matrix, $\lambda_{\max}(A)$ be the largest magnitude eigenvalue of A and v be the corresponding eigenvector. Let w be any unit vector such that $|\langle v, w\rangle| \geq \frac{2}{3\sqrt{n}}$. Then, for $k > \frac{1}{2\kappa}\cdot\log(9n/4)$, $\|A\cdot(A^k\cdot v)\|_2 \geq |\lambda_{\max}(A)|\cdot(1-\kappa)\cdot\|(A^k\cdot v)\|_2$.*

Let $v_{\max}$ be the eigenvector corresponding to the largest eigenvalue of A' and $\lambda_{\max}(A')$ be the corresponding eigenvalue. It is clear that there is some $i \in [n]$ such that $|\langle v_{\max}, e_i\rangle| \geq \frac{1}{\sqrt{n}}$.

Let i^* be any such index. We will show that

$$w = \frac{A'^k \cdot e_{i^*}}{\|A'^k \cdot e_{i^*}\|_2} \quad \text{and} \quad \lambda = \|A \cdot w\|_2 \tag{1}$$

are such that $\tilde{w}$ and $\tilde{\lambda}$ satisfy the conditions given in (a) and (b) of Theorem 7. Lemma 9 gives that $\sqrt{\mu_{i^*}} \geq (1-\delta)\cdot\lambda_{\max}(A')$, and hence $\|A'\cdot w\|_2^2 \geq (1-\delta)^2\lambda_{\max}(A')^2$.

Let $v_1, \ldots, v_n$ be the eigenvectors of A' (and hence of A) and $\lambda'_1, \ldots, \lambda'_n$ be the corresponding eigenvalues of A'. Let $w = \sum_{i=1}^n c_i \cdot v_i$ and let $S = \{i \in [n] : \lambda'_i \geq (1-\sqrt{\delta})\cdot\lambda_{\max}(A')\}$.

Proposition 10. $\sum_{i\notin S} c_i^2 \leq 2\sqrt{\delta}$.

Proof. We have $A'\cdot w = \sum_{i=1}^n c_i\cdot\lambda'_i\cdot v_i$ and hence $\|A'\cdot w\|_2^2 = \sum_{i=1}^n c_i^2\cdot\lambda_i'^2 = \sum_{i\in S} c_i^2\cdot\lambda_i'^2 + \sum_{i\notin S} c_i^2\cdot\lambda_i'^2$. As all eigenvalues of A' are non-negative, for $i\notin S$ we have $\lambda_i'^2 \leq (1-\sqrt{\delta})^2\cdot\lambda_{\max}^2(A')$. If $\sum_{i\notin S} c_i^2 = \kappa$, then

$$(1-\delta)^2\lambda_{\max}^2(A') \leq \sum_{i\in S} c_i^2\cdot\lambda_i'^2 + \sum_{i\notin S} c_i^2\cdot\lambda_i'^2 \leq (1-\kappa)\lambda_{\max}^2(A') + \kappa(1-\sqrt{\delta})^2\lambda_{\max}^2(A').$$

The last inequality uses that $\|A'\cdot w\|_2^2 \geq (1-\delta)^2\lambda_{\max}(A)^2$. Thus, $-\kappa\sqrt{\delta}(2-\sqrt{\delta}) \geq -\delta(2-\delta)$ and hence $\kappa \leq 2\sqrt{\delta}$. □

Proposition 11. *If $\lambda_{\max}(A) \geq \epsilon\cdot\|A\|_F$, then for w as defined above, $\|A\cdot w\|_2^2 \geq (1-6\cdot\delta^{1/4})\cdot\lambda_{\max}^2(A)$.*

Proof. Recall that an eigenvector v_i with eigenvalue λ_i of A maps to an eigenvalue $\lambda_i + t$ of A'. Thus, if i is such that $\lambda_i + t \geq (1-\sqrt{\delta})(\lambda_{\max}(A)+t)$, then $\lambda_i \geq (1-\sqrt{\delta})\lambda_{\max}(A) - \sqrt{\delta}\cdot t$. Since $\lambda_{\max}(A) \geq \epsilon t/2$, if we choose $\delta \leq \epsilon^4$, then $\lambda_i \geq (1-2\cdot\delta^{1/4})\cdot\lambda_{\max}(A)$. Thus, we have

$$S \subseteq \{i \in [n] : \lambda_i \geq (1-2\cdot\delta^{1/4})\cdot\lambda_{\max}(A)\}.$$

Now, observe that $A\cdot w = \sum_{i\in S} c_i\cdot\lambda_i\cdot v_i + \sum_{i\notin S} c_i\cdot\lambda_i\cdot v_i$. Hence,

$$\|A\cdot w\|_2^2 \geq \sum_{i\in S} c_i^2\cdot\lambda_i^2 \geq (1-2\cdot\delta^{1/4})^2\cdot\lambda_{\max}^2(A)\cdot(1-2\sqrt{\delta}) \geq (1-6\cdot\delta^{1/4})\cdot\lambda_{\max}^2(A),$$

where the second inequality uses Proposition 10. □

Proposition 12. *For w as defined in Equation (1) and $\lambda \geq 0$, if $\lambda_{\max}(A) \geq (\epsilon/2)\cdot\|A\|_F$ and $\|A\cdot w\|_2^2 = \lambda^2 \geq (1-10\cdot\delta^{1/4})\cdot\lambda_{\max}^2(A)$, then for $B = A - \lambda w\cdot w^T$, $\|B\cdot w\|_2 \leq \eta\cdot\|A\|_F/2$.*

Proof. We begin by noting that for $i\in S$, $\lambda'_i \geq (1-\sqrt{\delta})\lambda_{\max}(A')$. This implies that $\lambda_i + t \geq (1-\sqrt{\delta})(\lambda_{\max}(A)+t)$. Using the bounds $\lambda_{\max}(A) \geq (\epsilon/2)\|A\|_F$ and $\delta \leq \epsilon^4$, we get that $\lambda_i \geq (1-2\cdot\delta^{1/4})\cdot\lambda_{\max}(A)$.

By assumption we have that $(1-10\cdot\delta^{1/4})\lambda_{\max}(A) \leq \lambda$, and since $\|A\cdot w\|_2^2 = \lambda^2$ we also have that $\lambda \leq \lambda_{\max}(A)$. As a consequence, we have that for every $i\in S$, $|\lambda - \lambda_i| \leq 10\cdot\delta^{1/4}|\lambda_{\max}(A)|$. Note that

$$\begin{aligned}
\|B\cdot w\|_2^2 = \|A\cdot w - \lambda\cdot w\|_2^2 &= \sum_{i\in S} c_i^2(\lambda_i-\lambda)^2 + \sum_{i\notin S} c_i^2(\lambda_i-\lambda)^2 \\
&\leq 100\cdot\sqrt{\delta}\cdot\lambda_{\max}^2(A)\cdot\left(\sum_{i\in S} c_i^2\right) + \sum_{i\notin S} 2\cdot c_i^2\cdot(\lambda_i^2+\lambda^2) \\
&\leq 100\cdot\sqrt{\delta}\cdot\lambda_{\max}^2(A) + 8\sqrt{\delta}\|A\|_F^2 \leq 108\sqrt{\delta}\|A\|_F^2 \leq \eta^2\cdot\|A\|_F^2/4,
\end{aligned}$$

where we used Proposition 10 and Fact 64 in the last line. The last inequality holds because $\delta \leq \eta^4/10^8$. □

Claim 13. *If $\lambda_{\max}(A) \geq \epsilon \cdot \|A\|_F$, then the output satisfies the guarantees of part (a) of Theorem 7.*

Proof. Recall that $\delta = \min\{\epsilon^4/100, \eta^4/10^8\}$. We can then combine Proposition 10, Proposition 11 and Proposition 12 to get that $(1-\eta/2)\lambda_{\max}(A) \leq \lambda \leq \lambda_{\max}(A)$ and $\|Bw\|_2 \leq (\eta/2) \cdot \|A\|_F$. Finally, we use Lemma 15 (proved below) to get that $\|B\|_F \leq (1-\epsilon^2/20)\|A\|_F$.

Now, recall that $\|w - \tilde{w}\|_2 \leq \mathrm{poly}(\delta, 1/b, 1/n)$ and $|\lambda - \tilde{\lambda}| \leq \mathrm{poly}(\delta, 1/b, 1/n)$. This implies guarantees (i), (ii) and (iii). □

Claim 14. *If $\epsilon \geq \lambda_{\max}(A) \geq \epsilon/2$, then the output satisfies the conditions in part (b) of Theorem 7.*

Proof. If $\tilde{\lambda}^2 \leq (1 - 9 \cdot \delta^{1/4}) \cdot \epsilon^2 \cdot \|A\|_F^2$, then the algorithm outputs "small max eigenvalue" and the output is correct. On the other hand, if $\tilde{\lambda}^2 \geq (1 - 9 \cdot \delta^{1/4}) \cdot \epsilon^2 \cdot \|A\|_F^2$, then it implies that $\lambda^2 \geq (1 - 10 \cdot \delta^{1/4}) \cdot \epsilon^2 \cdot \|A\|_F^2$. As $\lambda_{\max}^2(A) \leq \epsilon^2 \|A\|_F^2$, hence $\lambda^2 \geq (1 - 10 \cdot \delta^{1/4}) \cdot \lambda_{\max}^2$. Since $\delta \leq \eta^4/10^8$, we get that $(1-\eta/2)\lambda_{\max}(A) \leq \lambda \leq \lambda_{\max}(A)$. As above, Proposition 10, Proposition 12 and Proposition 12 give that $\|Bw\|_2 \leq (\eta/2) \cdot \|A\|_F$, and Lemma 15 gives that $\|B\|_F \leq (1-\epsilon^2/20)\|A\|_F$. As before, using that $\|w - \tilde{w}\|_2 \leq \mathrm{poly}(\delta, 1/b, 1/n)$ and $|\lambda - \tilde{\lambda}| \leq \mathrm{poly}(\delta, 1/b, 1/n)$, we get guarantees (i), (ii) and (iii) from the output. □

Claims 8, 13 and 14 together establish Theorem 7 modulo the proof of Lemma 15, which we provide below. □

Lemma 15. *Let $A \in \mathbb{R}^{n \times n}$ be symmetric with $\|A\|_F = 1$. For $0 < \delta \leq \epsilon < 1$, let λ with $|\lambda| \geq 3\epsilon$ and $v \in \mathbb{R}^n$ with $\|v\|_2 = 1$ be such that the matrix $B = A - \lambda(vv^T)$ satisfies $\|Bv\|_2 \leq \delta$. Then, $\|B\|_F^2 \leq (1 - 3\epsilon^2)$.*

Proof. Recall that for any symmetric matrix $C \in \mathbb{R}^{n \times n}$ we have $\|C\|_F^2 = \mathrm{tr}(C^2)$. Hence we may prove the lemma by bounding from above the quantity $\mathrm{tr}(B^2)$. We can write

$$B^2 = (A - \lambda vv^T)^2 = A^2 + \lambda^2 (vv^T)^2 - \lambda A(vv^T) - \lambda (vv^T)A$$

and therefore

$$\mathrm{tr}(B^2) = \mathrm{tr}(A^2) + \lambda^2 \mathrm{tr}\left((vv^T)^2\right) - \lambda \mathrm{tr}\left(A(vv^T)\right) - \lambda \mathrm{tr}\left((vv^T)A\right).$$

Since $\|v\|_2 = 1$, we have that $\mathrm{tr}\left((vv^T)^2\right) = 1$. Moreover, $\mathrm{tr}\left(A(vv^T)\right) = \mathrm{tr}\left((vv^T)A\right)$. Therefore,

$$\mathrm{tr}(B^2) = 1 + \lambda^2 - 2\lambda \mathrm{tr}\left(A(vv^T)\right).$$

We will need the following claim:

Claim 16. *We have that $|\mathrm{tr}(Bvv^T)| \leq \delta$.*

Proof. It follows easily from the definition that

$$\mathrm{tr}(Bvv^T) = \sum_{i=1}^{n} (b^{(i)} \cdot v) v_i,$$

where $b^{(i)} \in \mathbb{R}^n$ is the i-th row of B. Since $Bv = [(b^{(1)} \cdot v) \ldots (b^{(n)} \cdot v)]^T$ the Cauchy-Schwarz inequality implies that

$$|\mathrm{tr}(Bvv^T)| \leq \|Bv\|_2 \|v\|_2.$$

The claim follows from the fact that $\|Bv\|_2 \leq \delta$ and $\|v\|_2 = 1$. □

Since

$$Avv^T = Bvv^T + \lambda(vv^T)^2$$

we have

$$\mathrm{tr}(Avv^T) = \mathrm{tr}(Bvv^T) + \lambda \mathrm{tr}\left((vv^T)^2\right) = \mathrm{tr}(Bvv^T) + \lambda.$$

Claim 16 now implies that

$$\lambda - \delta \leq \mathrm{tr}(Avv^T) \leq \lambda + \delta.$$

Since $|\lambda| \geq \epsilon > \delta$, the numbers $\lambda - \delta, \lambda, \lambda + \delta$ have the same sign, hence

$$\lambda \mathrm{tr}(Avv^T) = |\lambda| \cdot |\mathrm{tr}(Avv^T)| \geq |\lambda| \cdot (|\lambda| - \delta) = \lambda^2 - \delta|\lambda|$$

which gives that

$$\mathrm{tr}(B^2) \leq 1 + \lambda^2 - 2\lambda^2 + 2\delta|\lambda| = 1 - \lambda^2 + 2\delta|\lambda| = 1 - |\lambda|(|\lambda| - 2\delta) \leq 1 - 3\epsilon^2$$

where the last inequality follows from our assumptions on λ and δ. □

2.2.2 Technical claims about degree-2 polynomials. Before presenting and analyzing the APPROXIMATE-DECOMPOSE algorithm we need some technical setup. We start with a useful definition:

Definition 17 (Rotational invariance of polynomials)**.** *Given two polynomials $p(x) = \sum_{1 \leq i \leq j \leq n} a_{ij} x_i x_j + \sum_{1 \leq i \leq n} b_i x_i + C$ and $q(x) = \sum_{1 \leq i \leq j \leq n} a'_{ij} x_i x_j + \sum_{1 \leq i \leq n} b'_i x_i + C$ with the same constant term, we say that they are* rotationally equivalent *if there is an orthogonal matrix Q such that $Q^T \cdot A \cdot Q = A'$ and $Q^T \cdot b = b'$. If the matrix A' is diagonal then the polynomial q is said to be the* decoupled equivalent of p*. In this case, the eigenvalues of A (or equivalently A') are said to be the eigenvalues of the quadratic form p.*

Claim 18. *For any degree-2 polynomials $p(x)$ and $q(x)$ which are rotationally equivalent, the distributions of $p(x)$ and $q(x)$ are identical when $(x_1, \ldots, x_n) \sim \mathcal{N}(0,1)^n$.*

Proof. Observe that $q(x) = p(y)$ where $y = Q \cdot x$. Now, since $(x_1, \ldots, x_n)$ is distributed according to $\mathcal{N}(0,1)^n$ and Q is an orthogonal matrix, $(y_1, \ldots, y_n)$ has the same distribution. This proves the claim. □

We will also use the following simple fact which relates the variance of a degree-2 polynomial p with the Frobenius norm of the quadratic part of p.

Fact 19. *Let $p : \mathbb{R}^n \to \mathbb{R}$ be a degree-2 polynomial and let A be the matrix corresponding to its quadratic part. Then* $\mathrm{Var}(p) \geq 2\|A\|_F^2$.

Proof. Let $p(x) = \sum_{1 \leq i \leq j \leq n} a_{ij} x_i x_j + \sum_{1 \leq i \leq n} b_i x_i + C$. Equivalently, $p(x) = x^T \cdot A \cdot x + b^T \cdot x + C$ where A is the matrix corresponding to the quadratic part of p. Using the fact that A is symmetric, we get that there is an orthogonal matrix Q such that $Q^T A Q = \Lambda$ where Λ is diagonal. Using the fact that if $x \sim \mathcal{N}(0,1)^n$, then so is Qx, we get that the distribution of $p(x)$ and $q(x) = x^T Q^T A Q x + b^T Q x + C$ are identical when $x \sim \mathcal{N}(0,1)^n$. However, $q(x) = x^T \Lambda x + \mu^T x + C$ where $\mu = Q^T \cdot b$. Note that $q(x) = \sum_{i=1}^n (\lambda_i x_i^2 + \mu_i x_i) + C$. Hence,

$$\mathrm{Var}(q) = \sum_{i=1}^n \mathrm{Var}(\lambda_i x_i^2 + \mu_i x_i) = \sum_{i=1}^n 2\lambda_i^2 + \mu_i^2 \geq 2 \sum_{i=1}^n \lambda_i^2 = 2\|A\|_F^2$$

(recall that for a univariate Gaussian random variable $x \sim \mathcal{N}(0,1)$ we have $\mathbf{E}[x^4] = 3$ and hence $\mathrm{Var}(x^2) = 2$.). Since the distributions of $p(x)$ and $q(x)$ are identical, we have that $\mathrm{Var}(p) \geq 2\|A\|_F^2$.

□

We will also require another definition for degree-2 polynomials.

Definition 20. *Given $p : \mathbb{R}^n \to \mathbb{R}$ defined by $p(x) = \sum_{1\le i\le j\le n} a_{ij}x_ix_j + \sum_{1\le i\le n} b_ix_i + C$, define $SS(p)$ as $SS(p) = \sum_{1\le i\le j\le n} a_{ij}^2 + \sum_{1\le i\le n} b_i^2$.*

We now have the following simple claim.

Claim 21. *Given $p : \mathbb{R}^n \to \mathbb{R}$, we have that $2\,SS(p) \ge \mathrm{Var}(p) \ge SS(p)$.*

Proof. Since neither $SS(p)$ nor $\mathrm{Var}(p)$ is changed by adding a constant term to p it suffices to prove the claim for $p(x) = \sum_{1\le i\le j\le n} a_{ij}x_ix_j + \sum_{1\le i\le n} b_ix_i$. We have

$$\begin{aligned}\mathbf{E}_{x_1,\ldots,x_n\sim\mathcal{N}(0,1)}[p(x_1,\ldots,x_n)^2] &= \sum_{i=1}^n b_i^2\mathbf{E}[x_i^2] + \sum_{1\le i<j\le n} a_{ij}^2\mathbf{E}[x_i^2x_j^2] + \sum_{i=1}^n a_{ii}^2\mathbf{E}[x_i^4] + \sum_{1\le i<j\le n} 2a_{ii}a_{jj}\mathbf{E}[x_i^2x_j^2]\\ &= \sum_{i=1}^n (b_i^2 + 3a_{ii}^2) + \sum_{1\le i<j\le n}(a_{ij}^2 + 2a_{ii}a_{jj}).\end{aligned}$$

The first equality holds because every other cross-term has an odd power of some x_i (for $i \in [n]$), and for $x \sim \mathcal{N}(0,1)$ we have that $\mathbf{E}[x^t] = 0$ if t is odd. On the other hand, by linearity of expectation, $\mathbf{E}_{x_1,\ldots,x_n}[p] = \sum_{i=1}^n a_{ii}$ and hence

$$(\mathbf{E}_{x_1,\ldots,x_n}[p])^2 = \sum_{i=1}^n a_{ii}^2 + \sum_{1\le i<j\le n} 2a_{ii}a_{jj}$$

Hence, $\mathrm{Var}(p) = \sum_{i=1}^n (b_i^2 + 2a_{ii}^2) + \sum_{1\le i<j\le n} a_{ij}^2$. This implies the claimed bounds. □

Claim 22. *For the polynomial $q(y_1, x)$ constructed in Definition 5, the distributions of $q(y_1, x)$ and $p(x)$ are identical when $(y_1, x_1, \ldots, x_n) \sim \mathcal{N}(0,1)^n$.*

Proof. Note that

$$p(x) = p(\alpha_1 L_1(x) + R_1(x), \ldots, \alpha_n L_1(x) + R_n(x)).$$

Let D be the joint distribution of $(R_1(x), \ldots, R_n(x))$ when $(x_1, \ldots, x_n) \sim \mathcal{N}(0,1)^n$. As $R_i(x)$ is orthogonal to $L_1(x)$ for all $i \in [n]$, hence D is independent of the distribution of $L_1(x) = \sum_{i=1}^n w_i x_i$ (when $(x_1, \ldots, x_n) \sim \mathcal{N}(0,1)^n$). Also, $L_1(x)$ is distributed like a standard normal. Using these two facts, we get the claimed statement. □

Claim 23. *Given a degree-2 polynomial $p : \mathbb{R}^n \to \mathbb{R}$, let $L_1(x)$ be a normalized linear form and A be the matrix corresponding to the quadratic part of p. Let w_1 be the vector corresponding to $L_1(x)$ and let Q be any orthonormal matrix whose first column is w_1. Let $\tilde{A} = Q^T \cdot A \cdot Q$. Then $\mathrm{Var}(Res(p, L_1(x))) = 4 \cdot \sum_{1\le j\le n} \tilde{A}_{1j}^2$.*

Proof. Let $x_i = \alpha_{i1}L_1(x) + R_i(x)$. Let $Q = [w_1, \ldots, w_n]$ be an orthonormal matrix. Let $L_i(x)$ be the linear form corresponding to w_i. Note that $x_i = \sum_{j=1}^n Q_{ij}L_j(x)$ and hence $R_i(x) = \sum_{j>1}^n Q_{ij}L_j(x)$. Since $L_1(x), \ldots, L_n(x)$ are orthonormal, hence their joint distribution is same as $(y_1, \ldots, y_n) \sim \mathcal{N}(0,1)^n$.

Also, observe that this implies that the joint distribution of $R_1(x), \ldots, R_n(x)$ is independent of $L_1(x)$. As a consequence, we get that the distribution of $Res(p, L_1(x))$ is same as $Res(\tilde{p}, y_1)$ where

$$\tilde{p}(y_1, \ldots, y_n) = p(\sum_{j=1}^n Q_{1j}y_j, \ldots, \sum_{j=1}^n Q_{nj}y_j) = p((Q\cdot y)_1, \ldots, (Q\cdot y)_n)$$

Note that since A is the matrix corresponding to the quadratic part of p, we get that

$$Res(\tilde{p}, y_1) = Res(p((Q\cdot y)_1, \ldots, (Q\cdot y)_n), y_1) = Res(y^T \cdot Q^T \cdot A \cdot Q \cdot y, y_1) = Res(y^T \cdot \tilde{A} \cdot y, y_1)$$

Thus, $Res(y^T \cdot \tilde{A} \cdot y, y_1) = \sum_{j=2}^n 2\tilde{A}_{1j}y_1y_j$. Thus, $\mathrm{Var}(Res(y^T \cdot \tilde{A} \cdot y, y_1) = 4 \cdot \sum_{1\le j\le n} \tilde{A}_{1j}^2$. □

2.2.3 Proof of Theorem 6. Recall the statement of Theorem 6:

Theorem 6. *Let $p : \mathbb{R}^n \to \mathbb{R}$ be a degree-2 polynomial (with constant term 0) whose entries are b-bit integers and let $\epsilon, \eta > 0$. There exists a deterministic algorithm APPROXIMATE-DECOMPOSE which on input an explicit description of p, ϵ and η runs in time $\mathrm{poly}(n, b, 1/\epsilon, 1/\eta)$ and has the following guarantee :*

(a) If $\lambda_{\max}(p) \geq \epsilon\sqrt{\mathrm{Var}(p)}$, then the algorithm outputs rational numbers λ_1, μ_1 and a degree-2 polynomial $r : \mathbb{R}^{n+1} \to \mathbb{R}$ with the following property: for $(y, x_1, \ldots, x_n) \sim \mathcal{N}(0,1)^{n+1}$, the two distributions $p(x_1, \ldots, x_n)$ and $q(y_1, x_1, \ldots, x_n)$ are identical, where $q(y_1, x_1, \ldots, x_n)$ equals $\lambda_1 y_1^2 + \mu_1 y_1 + r(y_1, x_1, \ldots, x_n)$. Further, $\mathrm{Var}(Res(r, y_1)) \leq 4\eta^2 \mathrm{Var}(p)$ and $\mathrm{Var}(r) \leq (1 - \epsilon^4/40) \cdot \mathrm{Var}(p)$.

(b) If $\lambda_{\max}(p) < \epsilon\sqrt{\mathrm{Var}(p)}$, then the algorithm either outputs "small max eigenvalue" or has the same guarantee as (a).

Proof. The algorithm works as follows :

(i) Let A be the matrix corresponding to the quadratic part of p. If $\|A\|_F^2 < \epsilon^2 \cdot \mathrm{Var}(p)$, then output "small max eigenvalue".

(ii) Run the algorithm APPROXIMATE-LARGEST-EIGEN from Theorem 7 on the matrix A. If the output is "small max eigenvalue", then return "small max eigenvalue".

(iii) If the output of the algorithm APPROXIMATE-LARGEST-EIGEN is the tuple (λ, w_1), then for each unit vector e_i, we express $e_i = \alpha_i w_1 + v_i$ where v_i is orthogonal to w_1. For the sake of brevity, we will henceforth refer to $\sum_{j=1}^n v_{ij} x_j$ as $R_i(x)$.

(iv) Define the polynomial $q(y_1, x_1, \ldots, x_n)$ as $p(\alpha_1 y_1 + R_1(x), \ldots, \alpha_n y_1 + R_n(x))$.

The bound on the running time of the algorithm is obvious. We now give the proof of correctness of the algorithm. First of all, if $\|A\|_F^2 < \epsilon^2 \cdot \mathrm{Var}(p)$, then $\lambda_{\max}^2(A) \leq \|A\|_F^2 < \epsilon^2 \cdot \mathrm{Var}(p)$ and hence the output is correct. So from now on, we assume that $\|A\|_F^2 \geq \epsilon^2 \cdot \mathrm{Var}(p)$.

Now, assuming that the output of Theorem 7 in Step (ii) is "small max eigenvalue", then by the guarantee of Theorem 7 it must be the case that $\lambda_{\max}^2(A) \leq \epsilon^2 \cdot \|A\|_F^2$. Using Fact 19, we get that $\lambda_{\max}^2(A) \leq \epsilon^2 \cdot \mathrm{Var}(p)/2$.

Thus, in both the cases that the output of the algorithm is "small max eigenvalue", it is the case that $\lambda_{\max}^2(A) \leq \epsilon^2 \cdot \mathrm{Var}(p)$.

Now, consider the case in which the algorithm reaches Step (iii). It must be the case that $\|A\|_F \geq \epsilon \cdot \sqrt{\mathrm{Var}(p)}$, and by Claim 8 it must hold that $\lambda_{\max}(A) \geq (\epsilon/2) \cdot \|A\|_F$. We start with the following claim.

Claim 24. *The distribution of $q(y_1, x_1, \ldots, x_n)$ and $p(x_1, \ldots, x_n)$ are identical when $(y_1, x_1, \ldots, x_n) \sim \mathcal{N}(0,1)^{n+1}$.*

Proof. The polynomial $q(y_1, x_1, \ldots, x_n)$ is the same as the one constructed in Definition 5 and hence we can use Claim 22 to get the stated claim. □

Next, we prove the bound on $\mathrm{Var}(Res(q, y_1))$.

Claim 25. $\mathrm{Var}(Res(q, y_1)) \leq 2\eta^2 \mathrm{Var}(p)$.

Proof. Consider the orthogonal matrix $Q = [w_1, \ldots, w_n]$. Then, by using Claim 23, for $\tilde{A} = Q^T \cdot A \cdot Q$, $\mathrm{Var}(Res(q, y_1)) = 4\sum_{j>1} \tilde{A}_{1j}^2$. Now, using that $w_1, \ldots, w_n$ form an orthonormal basis, we get

$$\sum_{j>1} \tilde{A}_{1j}^2 = \sum_{j>1} (w_j^T \cdot A \cdot w_1)^2$$

Now, note that for the value λ that APPROXIMATE-LARGEST-EIGEN outputs in Step (iii) of APPROXIMATE-DECOMPOSE, we have

$$\eta^2 \|A\|_F^2 \geq \|A \cdot w_1 - \lambda w_1\|_2^2 = \sum_{j=1}^{n} (w_j^T \cdot A \cdot w_1 - \lambda w_j^T \cdot w_1)^2 \geq \sum_{j=2}^{n} (w_j^T \cdot A \cdot w_1)^2$$

Here the first inequality follows from Theorem 7 part (a)(ii) while the second equality follows from the definition of ℓ_2 norm of a vector. Thus, we get that

$$\mathrm{Var}(Res(q, y_1)) = 4 \sum_{j>1} \tilde{A}_{1j}^2 \leq 4 \cdot \|A \cdot w_1 - \lambda_1 w_1\|_2^2 \leq 4\eta^2 \|A\|_F^2 \leq 2\eta^2 \,\mathrm{Var}(p)$$

The last inequality uses Fact 19. □

Thus, the only part that remains to be shown is that the variance of r goes down.

Claim 26. $\mathrm{Var}(r(y_1, x_1, x_2, \ldots, x_n)) \leq (1 - \epsilon^4/40) \cdot \mathrm{Var}(p)$.

Proof. Note that $q(x) = \lambda_1 y_1^2 + \mu_1 y_1 + r(y_1, x_1, \ldots, x_n)$. Let $\tilde{r}(y_1, x_1, \ldots, x_n) = Res(r, y_1)$. Since $\mathrm{Var}(\tilde{r}) \leq 2\eta^2 \,\mathrm{Var}(p)$, hence using Fact 65, we get that

$$\mathrm{Var}(\lambda_1 y_1^2 + \mu_1 y_1 + r(y_1, x_1, x_2, \ldots, x_n) - \tilde{r}(y_1, x_1, \ldots, x_n)) \leq (1 + 2\eta)^2 \,\mathrm{Var}(p).$$

However, note that $r(y_1, x_1, x_2, \ldots, x_n) - \tilde{r}(y_1, x_1, \ldots, x_n)$ is independent of y (call it $\tilde{r}_1(x_1, \ldots, x_n)$). As a result, we get

$$\mathrm{Var}(\tilde{r}_1(x_1, \ldots, x_n)) + \mathrm{Var}(\lambda_1 y_1^2 + \mu_1 y_1) \leq (1 + 2\eta)^2 \,\mathrm{Var}(p)$$

Since the algorithm reaches Step (iii) only if $\lambda_{\max}(A) \geq \epsilon \|A\|_F/2$ and $\|A\|_F \geq \epsilon\sqrt{\mathrm{Var}(p)}$, hence $\lambda_1 \geq (1 - \eta)(\epsilon^2/2)\sqrt{\mathrm{Var}(p)}$ and hence

$$\mathrm{Var}(\tilde{r}_1(x_1, \ldots, x_n)) \leq (1 + 2\eta)^2 \,\mathrm{Var}(p) - (1 - \eta)^2(\epsilon^4/4) \,\mathrm{Var}(p) \leq (1 - \epsilon^4/20) \cdot \mathrm{Var}(p).$$

This uses the fact $\eta \leq \epsilon^4/10^8$. Again, using Fact 65 and that $\eta \leq \epsilon^4/10^8$, we get that since $\mathrm{Var}(\tilde{r}) \leq 2\eta^2 \,\mathrm{Var}(p)$

$$\mathrm{Var}(r(y_1, x_1, x_2, \ldots, x_n)) \leq (1 - \epsilon^4/40) \cdot \mathrm{Var}(p).$$

□

This concludes the proof of Theorem 6. □

`Construct-Junta-PTF`

Input: Explicit description of an n-variable degree-2 polynomial p and $\epsilon > 0$.
Output: A degree-2 polynomial $q(y) = \sum_{i=1}^{K}(\lambda_i y_i^2 + \mu_i y_i) + C'$, with $K = \tilde{O}(1/\epsilon^4)$, such that $|\mathbf{Pr}_{x\in\mathcal{N}(0,1)^n}[p(x) \geq 0] - \mathbf{Pr}_{y\in\mathcal{N}(0,1)^K}[q(y) \geq 0]| = O(\epsilon)$.

Let $p(x) = p'(x) + C$, where p' has constant term 0. Assume by rescaling that $\mathrm{Var}(p') = \mathrm{Var}(p) = 1$.

Initialize: $i = 1$; $s_1(x) = p'(x)$; $h_0 \equiv 0$.

Repeat the following:

1. Fix $\alpha \stackrel{\text{def}}{=} \Theta(\epsilon^4/\log^2(1/\epsilon))$.
2. **If** $\mathrm{Var}(s_i) < \alpha$ **output** the polynomial $q_{i-1} : \mathbb{R}^{i-1} \to \mathbb{R}$ defined by $q_{i-1}(y_1, \ldots, y_{i-1}) = h_{i-1}(y_1, \ldots, y_{i-1}) + \mathbf{E}[s_i] + C$.
3. **If** $\mathrm{Var}(s_i) \geq \alpha$ **do the following**
   (a) Round down each coefficient of s_i to its closest integer multiple of $\gamma/(Kn)$, where $\gamma = \tilde{\Theta}((\epsilon/K)^2)\sqrt{\alpha}$ and $K = \tilde{\Theta}(1/\epsilon^4)$. Let $s_i'(x)$ be the rounded polynomial.
   (b) Run the routine APPROXIMATE-DECOMPOSE$(s_i', \epsilon, \eta := \tilde{\Theta}(\epsilon^4/K^4))$.
   (c) **If** the routine returns "small max eigenvalue" **output** the polynomial $q_i' : \mathbb{R}^i \to \mathbb{R}$ $q_i'(y_1, \ldots, y_i) = h_{i-1}(y_1, \ldots, y_{i-1}) + \beta_{s_i'} y_i + \mathbf{E}[s_i'] + C$. where $\beta_{s_i'}$ is obtained by rounding down $\sqrt{\mathrm{Var}(s_i')}$ to the nearest integer multiple of $\epsilon\alpha/2$.
   (d) **If** the routine outputs numbers λ_i, μ_i and a polynomial $r_i : \mathbb{R}^{n+1} \to \mathbb{R}$, we define $s_{i+1}(x) = r_i(y_i, x_1, \ldots, x_n) - Res(r_i, y_i)$ and $h_i(y_1, \ldots, y_i) = h_{i-1}(y_1, \ldots, y_{i-1}) + (\lambda_i y_i^2 + \mu_i y_i)$.
4. $i = i + 1$.

End Loop

2.3 The first stage: Constructing a junta polynomial. In this section, we describe an algorithm `Construct-Junta-PTF` which given as input an n variable quadratic polynomial p, runs in time $\mathrm{poly}(n/\epsilon)$ and outputs a quadratic polynomial q on $\tilde{O}(1/\epsilon^4)$ variables such that the distributions of $p(Y)$ and $q(Y)$ (when $Y \sim \mathcal{N}(0,1)^n$) are $O(\epsilon)$ close in Kolmogorov distance. More precisely, we prove the following theorem.

Theorem 27. *The algorithm* `Construct-Junta-PTF` *has the following performance guarantee: It takes as input an explicit description of an n-variable degree-2 polynomial p with b-bit integer coefficients, and a value $\epsilon > 0$. It runs (deterministically) in time $\mathrm{poly}(n, b, 1/\epsilon)$ and outputs a degree-2 polynomial $q = \sum_{i=1}^{K}(\lambda_i y_i^2 + \mu_i y_i) + C'$ such that $\left|\mathbf{Pr}_{x\in\mathcal{N}(0,1)^n}[p(x) \geq 0] - \mathbf{Pr}_{y\in\mathcal{N}(0,1)^K}[q(y) \geq 0]\right| \leq O(\epsilon)$, where each $\lambda_i, \mu_i \in \mathbb{Z}$ and $K = \tilde{O}(1/\epsilon^4)$.*

The full proof of the theorem is technical so first we give some intuition behind the algorithm and its proof of correctness.

As mentioned in the introduction, if we were given the exact SVD then we could construct a decoupled n-variable polynomial $\tilde{p}$ such that $p(X)$ and $\tilde{p}(X)$ have the same distribution when $X \sim \mathcal{N}(0,1)^n$. Since we cannot compute the exact SVD, we instead iteratively use the APPROXIMATE-DECOMPOSE algorithm.

Consider the first time the APPROXIMATE-DECOMPOSE algorithm is called (on the polynomial s'_1 – think of this as just being the input polynomial p). If it outputs "small max eigenvalue", then using Chatterjee's recent CLT for functions of Gaussians, we show that for $X \sim \mathcal{N}(0,1)^n$ the distribution of $s'_1(X)$ is close to $\mathcal{N}(\mathbf{E}[s'_1], \mathrm{Var}(s'_1))$ in total variation distance. In this case we can set the polynomial $q = \sqrt{\mathrm{Var}[s'_1]}y_1 + \mathbf{E}[s'_1]$ (note that we ignore technical details like the "rounding" that the algorithm performs in this intuitive discussion).

On the other hand, if the APPROXIMATE-DECOMPOSE algorithm does not output "small max eigenvalue", then let $\lambda_1 y_1^2 + \mu_1 y_1 + r_1(y_1, x_1, \ldots, x_n)$ be the output of APPROXIMATE-DECOMPOSE. Since $\mathrm{Var}(Res(r, y_1))$ is small, it is not difficult to show that the distribution of $\lambda_1 y_1^2 + \mu_1 y_1 + r_1(y_1, x_1, \ldots, x_n) - Res(r, y_1)$ is close to the distribution of s'_1 in Kolmogorov distance. However, note that by definition, $r_1(y_1, x_1, \ldots, x_n) - Res(r_1, y_1)$ does not involve the variable y_1. We now iteratively work with the polynomial $s_2(x_1, \ldots, x_n) = r_1(y_1, x_1, \ldots, x_n) - Res(r_1, y_1)$. In particular, we apply APPROXIMATE-DECOMPOSE to the polynomial s'_2 (this is a "rounded" version of s_2 – think of it as just being s_2). In this second stage, if APPROXIMATE-DECOMPOSE outputs "small max eigenvalue", we can set the polynomial q to be $q = \lambda_1 y_1^2 + \mu_1 y_1 + \sqrt{\mathrm{Var}(s'_2)}y_2 + \mathbf{E}[s'_2]$; otherwise we proceed as we did earlier to obtain λ_2, μ_2, $r_2(y_2, x_1, \ldots, x_n)$; and so on.

If this iterative procedure terminates within $\tilde{O}(1/\epsilon^4)$ steps because of APPROXIMATE-DECOMPOSE returning "small max eigenvalue" at some stage, then its output is easily seen to satisfy the conditions of the theorem. If the procedure continues through $K = \tilde{O}(1/\epsilon^4)$ steps without terminating, then using the critical-index style analysis, it can be shown that the variance of the remaining polynomial s_K is at most $O(\epsilon)$ (recall from Theorem 6 that each call to APPROXIMATE-DECOMPOSE reduces the variance of the polynomial by a multiplicative factor of $(1 - \epsilon^4/40)$). Since this variance is so small it can be shown that the remaining polynomial can be safely ignored and that the polynomial $\sum_{1 \le i \le K} \lambda_i y_i^2 + \mu_i y_i + \mathbf{E}[s_K]$ satisfies the conditions of the theorem.

The rest of this section is devoted to the proof of the above theorem. We start with a couple of preliminary lemmas:

Lemma 28. *Let $p, q : \mathbb{R}^n \to \mathbb{R}$ be degree-2 polynomials such that $\mathrm{Var}(q) \le \epsilon' \mathrm{Var}(p)$, where $\epsilon' = O(\epsilon^4 / \log^2(1/\epsilon))$. For $x \sim \mathcal{N}(0,1)^n$, let D denote the distribution of $p(x) + q(x)$ and $\tilde{D}$ the distribution of $p(x) + \mathbf{E}[q(x)]$. Then we have $d_K(D, \tilde{D}) \le \epsilon$.*

Proof. By the definition of the Kolmogorov distance it is no loss of generality to assume that $\mathbf{E}[q] = 0$. For a fixed but arbitrary $\theta \in \mathbb{R}$ we will show that

$$|\mathbf{Pr}[p(x) + q(x) \le \theta] - \mathbf{Pr}[p(x) \le \theta]| \le \epsilon.$$

We bound the LHS as follows: Fix $x \in \mathbb{R}^n$. The point x contributes to the LHS only if there exists $\delta > 0$ such that either $|p(x) - \theta| \le \delta$ or $|q(x)| \ge \delta$. We select δ appropriately and bound the probability of the first event using Theorem 69 and the probability of the second event using Theorem 68. Indeed, fix $\delta = \Theta(\epsilon^2)\sqrt{\mathrm{Var}(p)} \ge \Omega(\log(1/\epsilon))\sqrt{\mathrm{Var}(q)}$, where the inequality follows from the assumption $\mathrm{Var}(q) \le \epsilon' \mathrm{Var}(p)$. Theorem 69 yields

$$\mathbf{Pr}_{x \sim \mathcal{N}(0,1)^n}\left[|p(x) - \theta| \le \delta\right] = \mathbf{Pr}_{x \sim \mathcal{N}(0,1)^n}\left[|p(x) - \theta| \le \Theta(\epsilon^2)\sqrt{\mathrm{Var}(p)}\right] \le \epsilon/2 \tag{2}$$

and by Theorem 68

$$\mathbf{Pr}_{x \sim \mathcal{N}(0,1)^n}\left[|q(x)| \ge \delta\right] \le \mathbf{Pr}_{x \sim \mathcal{N}(0,1)^n}\left[|q(x)| \ge \Omega(\log(1/\epsilon))\sqrt{\mathrm{Var}(q)}\right] \le \epsilon/2. \tag{3}$$

The lemma now follows by a union bound. □

As a consequence we have the following:

Proposition 29. *Let $p : \mathbb{R}^n \to \mathbb{R}$ be a degree-2 polynomial with $\mathrm{Var}(p) \geq \alpha$. Consider the polynomial $p' : \mathbb{R}^n \to \mathbb{R}$ obtained by rounding down each coefficient of p to its closest integer multiple of γ/n, where $\gamma = O(\epsilon^2/\log(1/\epsilon)) \cdot \sqrt{\alpha}$. Then, we have*

$$d_{\mathrm{K}}(p, p') \leq \epsilon.$$

Proof. Note that $e(x) = p(x) - p'(x) = \sum_{i \leq j} \delta_{i,j} x_i x_j + \sum_i \gamma_i x_i$ where $|\delta_{i,j}| \leq \gamma/n$ and $|\gamma_i| \leq \gamma/n$. As a consequence, we have $SS(e) \leq \gamma^2$ and therefore

$$\mathrm{Var}(e) \leq 2SS(e) \leq \Theta(\epsilon^4/\log^2(1/\epsilon))\alpha \leq \Theta(\epsilon^4/\log^2(1/\epsilon))\,\mathrm{Var}(p)$$

where we used Claim 21 and the definition of γ. Fact 65 gives that $\mathrm{Var}(e) \leq \Theta(\epsilon^4/\log^2(1/\epsilon))\,\mathrm{Var}(p')$ and Lemma 28 now implies that

$$d_{\mathrm{K}}(p, p') \leq \epsilon.$$

□

For the sake of intuition, we start by analyzing the first iteration of the loop. In the beginning of the first iteration, we have $s_1(x) = p'(x)$, where p' is the polynomial p without its constant term C. Hence we have $\mathrm{Var}(s_1) = 1$, which means that Step 2 of the loop is not executed. In Step 3(a) we round s_1 to obtain the rounded polynomial s_1'. By Proposition 29, it follows that

$$d_{\mathrm{K}}(s_1, s_1') \leq \epsilon/K. \tag{4}$$

Also note that the coefficients of s_1' are integer multiples of $\gamma/(Kn)$ of magnitude $\mathrm{poly}(n/\epsilon)$, hence up to a scaling factor they are ℓ-bit integers for $\ell = O(\log(n/\epsilon))$. In Step 3(b) we run the routine APPROXIMATE-DECOMPOSE on the rounded polynomial s_1'. (Note that the routine runs in $\mathrm{poly}(n/\epsilon)$ time.)

If the routine returns "small max eigenvalue" (Step 3(c)) then Theorem 6 guarantees that the maximum magnitude eigenvalue of s_1' is indeed small, in particular $|\lambda_{\max}(s_1')| \leq \epsilon\sqrt{\mathrm{Var}(s_1')}$. In this case, the algorithm outputs the univariate polynomial $q_1'(y_1) = \beta_{s_1'} y_1 + \mathbf{E}[s_1'] + C$, where $|\beta_{s_1'} - \sqrt{\mathrm{Var}(s_1')}| \leq \epsilon\alpha/2$. We have the following:

Claim 30. *If $|\lambda_{\max}(s_1')| \leq \epsilon\sqrt{\mathrm{Var}(s_1')}$, we have that $d_{\mathrm{K}}(s_1'(x), q_1'(y_1)) = O(\epsilon)$.*

To prove this claim we will need the following lemma. Its proof uses a powerful version of the Central Limit Theorem for functions of independent Gaussian random variables (which can be obtained using Stein's method):

Lemma 31. *Let $p : \mathbb{R}^n \to \mathbb{R}$ be a degree-2 polynomial over independent standard Gaussians. If $|\lambda_{\max}(p)| \leq \epsilon\sqrt{\mathrm{Var}(p)}$, then p is $O(\epsilon)$-close to the Gaussian $\mathcal{N}(\mathbf{E}[p], \mathrm{Var}(p))$ in total variation distance (hence, also in Kolmogorov distance).*

The proof of Lemma 31 is deferred to Section 2.3.1.

Proof of Claim 30. Since $|\lambda_{\max}(s_1')| \leq \epsilon\sqrt{\mathrm{Var}(s_1')}$, by Lemma 31 it follows that

$$d_{\mathrm{K}}\left(s_1'(x), \sqrt{\mathrm{Var}(s_1')}y_1 + \mathbf{E}[s_1']\right) = O(\epsilon),$$

where $x \sim \mathcal{N}(0,1)^n$ and $y_1 \sim \mathcal{N}(0,1)$. Since $\beta_{s_1'} \leq \sqrt{\mathrm{Var}(s_1')}$, Fact 67 yields

$$d_{\mathrm{K}}\left(\beta_{s_1'} y_1 + \mathbf{E}[s_1'], \sqrt{\mathrm{Var}(s_1')} y_1 + \mathbf{E}[s_1']\right) \leq (1/2)\frac{|\beta_{s_1'}^2 - \mathrm{Var}(s_1')|}{\beta_{s_1'}^2} \leq \frac{\epsilon\alpha}{\beta_{s_1'}^2} = O(\epsilon)$$

where we used that $|\beta_{s_1'}^2 - \mathrm{Var}(s_1')| \leq 2|\beta_{s_1'} - \sqrt{\mathrm{Var}(s_1')}| \leq \epsilon\alpha$ and that $\beta_{s_1'}^2 \geq \alpha/2$ (which uses that $\mathrm{Var}(s_1) \geq \alpha$ and $|\mathrm{Var}(s_1) - \mathrm{Var}(s_1')| \leq 2\gamma^2/K^2 \leq \epsilon\alpha$).

The claim follows from the aforementioned and the triangle inequality. □

Now we analyze the execution of Step 3(d). Consider the numbers λ_1, μ_1 and degree-2 polynomial $r_1 : \mathbb{R}^{n+1} \to \mathbb{R}$ returned by the routine Approximate-Decompose. Consider the polynomial

$$g_1(y_1, x_1, \ldots, x_n) = \lambda_1 y_1^2 + \mu_1 y_1 + r_1(y_1, x_1, \ldots, x_n).$$

Theorem 6 guarantees that the random variables $s_1'(x_1, \ldots, x_n)$ and $g_1(y_1, x_1, \ldots, x_n)$, with $(y_1, x_1, \ldots, x_n) \sim \mathcal{N}(0,1)^{n+1}$ have identical distributions. (In particular, this implies that $\mathrm{Var}(g_1) = \mathrm{Var}(s_1')$.) The algorithm proceeds to define

$$s_2(x_1, \ldots, x_n) = r_1(y_1, x_1, \ldots, x_n) - Res(r_1, y_1)$$

and

$$h_1(y_1) = \lambda_1 y_1^2 + \mu_1 y_1.$$

Note that the two summands ($\lambda_1 y_1^2 + \mu_1 y_1$ and $r_1(y_1, x_1, \ldots, x_n)$) defining g_1 are correlated (because of the variable y_1). An important fact is that if we subtract $Res(r_1, y_1)$ from the polynomial r_1, the distribution of the resulting polynomial remains close in Kolmogorov distance:

Claim 32. *We have that* $d_{\mathrm{K}}(s_1', s_2 + h_1) \leq \epsilon/K$.

Proof. Note that

$$d_{\mathrm{K}}(s_1', h_1 + s_2) = d_{\mathrm{K}}(g_1, h_1 + s_2) = d_{\mathrm{K}}(g_1, g_1 - Res(r_1, y_1)).$$

By Theorem 6 it follows that

$$\mathrm{Var}(Res(r_1, y_1)) \leq \eta^2 \mathrm{Var}(s_1') = \eta^2 \mathrm{Var}(g_1).$$

Since $\mathbf{E}[Res(r_1, y_1)] = 0$, by Lemma 28 we get that $d_{\mathrm{K}}(g_1, s_2 + h_1) \leq \epsilon/K$ as desired. □

The advantage of doing this is that s_2 and h_1 are independent random variables, since s_2 does not depend on y_1. As a consequence, we also obtain the following:

Fact 33. $\mathrm{Var}(s_2 + h_1) = \mathrm{Var}(s_2) + \mathrm{Var}(h_1) \geq 1 - \epsilon/K$.

Proof. Note that

$$g_1 = h_1 + r_1 = h_1 + s_2 + Res(r_1, y_1).$$

We have that $\mathrm{Var}(g_1) = \mathrm{Var}(s_1') \geq (1 - \epsilon^2/K^2)\mathrm{Var}(s_1)$ and $\mathrm{Var}(Res(r_1, y_1)) \leq 4\eta^2 \mathrm{Var}(s_1')$. By Fact 65 it follows that

$$\mathrm{Var}(s_2 + h_1) \geq (1 - 2\eta)\mathrm{Var}(g_1) = (1 - 2\eta)\mathrm{Var}(s_1') \geq (1 - 2\eta)(1 - \epsilon^2/K^2)\mathrm{Var}(s_1)$$

which completes the proof since $\mathrm{Var}(s_1) = 1$. □

We also have that the variance of the polynomial s_2 is smaller than $\mathrm{Var}(s_1)$ by a multiplicative factor:

Claim 34. *We have* $\mathrm{Var}(s_2) \leq (1 - \epsilon^4/40)$.

Proof. By Theorem 6 we know that

$$\mathrm{Var}(r_1) \leq (1 - \epsilon^4/40)\,\mathrm{Var}(s_1') \leq (1 - \epsilon^4/40)$$

where the second inequality used the fact that $\mathrm{Var}(s_1') \leq \mathrm{Var}(s_1) \leq 1$. Now note that s_2 is obtained from r_1 by removing a subset of its terms. Therefore, $\mathrm{Var}(s_2) \leq \mathrm{Var}(r_1)$ and the claim follows. □

This concludes our analysis of the first iteration of the loop.

We are now ready to analyze a generic iteration of the loop. Many aspects of this analysis will be similar to our earlier analysis of the first iteration. Consider the j-th iteration of the loop, where $j \geq 2$. We can assume by induction that for all $i < j$ the following hold:

(a) $d_{\mathrm{K}}(s_i, s_i') \leq \epsilon/K$ (for $j = 2$ this holds by (4));

(b) $d_{\mathrm{K}}(h_{i-1} + s_i, h_i + s_{i+1}) \leq 2\epsilon/K$, (for $j = 2$ this holds by (a) and Claim 32);

(c) $\mathrm{Var}(s_{i+1}) + \mathrm{Var}(h_i) \geq (1 - \epsilon/K)^i$, and (for $j = 2$ this holds by Fact 33); and

(d) $\mathrm{Var}(s_{i+1}) \leq (1 - \epsilon^4/40)^i$ (for $j = 2$ this holds by Claim 34).

We start by observing that $j - 1 \leq K$, i.e., the total number of iterations is at most $K + 1$. Indeed, by (d) above, for $i = K$ we will have $\mathrm{Var}(s_{i+1}) \leq (1 - \epsilon^4/40)^K < \alpha$ and the algorithm terminates in Step 2.

In the beginning of the j-th iteration, we have the polynomial $s_j(x_1, \ldots, x_n)$, satisfying $\mathrm{Var}(s_j) \leq (1 - \epsilon^4/40)^{j-1}$ and the polynomial $h_{j-1}(y_1, \ldots, y_{j-1})$. If the variance has become very small, we can "truncate" s_j taking into account its expectation (Step 2). Consider the polynomial

$$q_{j-1}(y_1, \ldots, y_{j-1}) = h_{j-1}(y_1, \ldots, y_{j-1}) + \mathbf{E}[s_j] + C.$$

We have the following claim:

Claim 35. *Suppose that* $\mathrm{Var}(s_j) < \alpha$, *where* $\alpha \stackrel{def}{=} \Theta(\epsilon^4/\log^2(1/\epsilon))$. *Then, we have that*

$$d_{\mathrm{K}}(h_{j-1} + s_j + C, q_{j-1}) \leq \epsilon.$$

Proof. The claim is equivalent to showing that $d_{\mathrm{K}}(h_{j-1} + s_j, h_{j-1} + \mathbf{E}[s_j]) \leq \epsilon$. Note that by Part (c) of the inductive hypothesis we have that $\mathrm{Var}(h_{j-1}) + \mathrm{Var}(s_j) \geq (1 - \epsilon/K)^{j-1}$. Since $\mathrm{Var}(s_j) < \alpha$ we get that

$$\mathrm{Var}(h_{j-1}) > (1 - \epsilon/K)^{j-1} - \alpha \geq 1/2,$$

where the last inequality uses the fact that $j \leq K + 1$. The claim follows by an application of Lemma 28 for the polynomials h_{j-1} and s_j. □

Combining the above claim with Parts (a) and (b) of the inductive hypothesis and using the triangle inequality completes the correctness analysis of the algorithm in the case that it exits in Step 2.

We now consider the complementary case that $\mathrm{Var}(s_j) \geq \alpha$ (Step 3). In Step 3(a) we round s_j to obtain the rounded polynomial s_j'. By Proposition 29, it follows that

$$d_{\mathrm{K}}(s_j, s_j') \leq \epsilon/K$$

establishing Part (a) of the inductive hypothesis for $i = j$.

Also note that the coefficients of s'_j are integer multiples of $\gamma/(Kn)$ of magnitude $\mathrm{poly}(n/\epsilon)$. In Step 3(b) we run the routine APPROXIMATE-DECOMPOSE on the rounded polynomial s'_j. (Note that the routine runs in $\mathrm{poly}(n/\epsilon)$ time.)

If the routine returns "small max eigenvalue" (Step 3(c)) then Theorem 6 guarantees that the maximum magnitude eigenvalue of s'_j is indeed small, in particular $|\lambda_{\max}(s'_j)| \leq \epsilon\sqrt{\mathrm{Var}(s'_j)}$. In this case, the algorithm outputs the polynomial $q'_j(y_1, \ldots, j_j) = h_{j-1}(y_1, \ldots, y_{j-1}) + \beta_{s'_j} y_j + \mathbf{E}[s'_j] + C$, where $|\beta_{s'_j} - \sqrt{\mathrm{Var}(s'_j)}| \leq \epsilon\alpha/2$. We have the following, which is very similar to Claim 30:

Claim 36. *If $|\lambda_{\max}(s'_j)| \leq \epsilon\sqrt{\mathrm{Var}(s'_j)}$, we have that $d_{\mathrm{K}}(s'_j(x), \beta_{s'_j} y_j + \mathbf{E}[s'_j]) = O(\epsilon)$.*

Proof. Since $|\lambda_{\max}(s'_j)| \leq \epsilon\sqrt{\mathrm{Var}(s'_j)}$, by Lemma 31 it follows that

$$d_{\mathrm{K}}\left(s'_j(x), \sqrt{\mathrm{Var}(s'_j)} y_j + \mathbf{E}[s'_j]\right) = O(\epsilon),$$

where $x \sim \mathcal{N}(0,1)^n$ and $y_j \sim \mathcal{N}(0,1)$. Hence, by Fact 67, we get that

$$d_{\mathrm{K}}\left(\beta_{s'_j} y_j + \mathbf{E}[s'_j], \sqrt{\mathrm{Var}(s'_j)} y_j + \mathbf{E}[s'_j]\right) \leq (1/2)\frac{|\beta^2_{s'_j} - \mathrm{Var}(s'_j)|}{\beta^2_{s'_j}} \leq \frac{\epsilon\alpha}{\beta^2_{s'_j}} = O(\epsilon)$$

where the second inequality used the fact that $|\beta^2_{s'_j} - \mathrm{Var}(s'_j)| \leq 2\epsilon\alpha$ and the last uses the fact that $\beta^2_{s'_j} \geq \alpha/2$. The claim follows from the aforementioned and the triangle inequality. □

Our final claim for this case (the case that the algorithm exits in Step 3(c)) is the following:

Claim 37. *We have that $d_{\mathrm{K}}(q'_j, p) = O(\epsilon)$.*

Proof. First, recall that $d_{\mathrm{K}}(s_j, s'_j) \leq \epsilon/K$ and therefore by the above claim and triangle inequality we get $d_{\mathrm{K}}(s_j(x), \beta_{s'_j} y_j + \mathbf{E}[s'_j]) = O(\epsilon)$. A convolution argument (exploiting independence) now gives that $d_{\mathrm{K}}(h_{j-1} + s_j, h_{j-1} + \beta_{s'_j} y_j + \mathbf{E}[s'_j]) = O(\epsilon)$. Combining the above with Parts (a) and (b) of the inductive hypothesis yields the claim by an application of the triangle inequality. □

Now we analyze the execution of Step 3(d). To finish the proof it suffices to show that the inductive hypotheses (a)–(d) all hold for $i = j$. Consider the numbers λ_j, μ_j and degree-2 polynomial $r_j : \mathbb{R}^{n+1} \to \mathbb{R}$ returned by the routine APPROXIMATE-DECOMPOSE. Consider the polynomial

$$g_j(y_j, x_1, \ldots, x_n) = \lambda_j y_j^2 + \mu_j y_j + r_j(y_j, x_1, \ldots, x_n).$$

Theorem 6 guarantees that the random variables $s'_j(x_1, \ldots, x_n)$ and $g_j(y_j, x_1, \ldots, x_n)$, with $(y_j, x_1, \ldots, x_n) \sim \mathcal{N}(0,1)^{n+1}$ have identical distributions. (In particular, this implies that $\mathrm{Var}(g_j) = \mathrm{Var}(s'_j)$.) The algorithm proceeds to define

$$s_{j+1}(x_1, \ldots, x_n) = r_j(y_j, x_1, \ldots, x_n) - Res(r_j, y_j)$$

and

$$h_j(y_1, \ldots, y_j) = \sum_{i=1}^{j} (\lambda_i y_i^2 + \mu_i y_i).$$

Note that the two summands $\lambda_j y_j^2 + \mu_j y_j$ and $r_j(y_j, x_1, \ldots, x_n)$ in g_j are correlated (because of the variable y_j). Similarly to the first iteration, if we remove $Res(r_j, y_j)$, there is a very small change in the Kolmogorov distance :

Claim 38. *We have that* $d_{\mathrm{K}}(s'_j, \lambda_j y_j^2 + \mu_j y_j + s_{j+1}) \leq \epsilon/K$.

Proof. Note that

$$d_{\mathrm{K}}(s'_j, \lambda_j y_j^2 + \mu_j y_j + s_{j+1}) = d_{\mathrm{K}}(g_j, \lambda_j y_j^2 + \mu_j y_j + s_{j+1}) = d_{\mathrm{K}}(g_j, g_j - Res(r_j, y_j)).$$

By Theorem 6 it follows that

$$\mathrm{Var}(Res(r_j, y_j)) \leq \eta^2 \mathrm{Var}(s'_j) = \eta^2 \mathrm{Var}(g_j).$$

Since $\mathbf{E}[Res(r_j, y_j) = 0]$, by Lemma 28 we get that $d_{\mathrm{K}}(g_j, s_{j+1} + h_j) \leq \epsilon/K$ as desired. □

As a corollary, we establish Part (b) of the induction for $i = j$.

Corollary 39. *We have that* $d_{\mathrm{K}}(h_{j-1} + s_j, h_j + s_{j+1}) \leq 2\epsilon/K$.

Proof. By Claim 38 and the fact that $d_{\mathrm{K}}(s_j, s'_j) \leq \epsilon/K$, we get that $d_{\mathrm{K}}(s_j, \lambda_j y_j^2 + \mu_j y_j + s_{j+1}) \leq 2\epsilon/K$. By a convolution argument (exploiting independence), it follows that $d_{\mathrm{K}}(h_{j-1} + s_j, h_{j-1} + \lambda_j y_j^2 + \mu_j y_j + s_{j+1}) \leq 2\epsilon/K$ which completes the proof. □

As a consequence, we also obtain the following, establishing Part (c) of the induction:

Fact 40. $\mathrm{Var}(s_{j+1}) + \mathrm{Var}(h_j) \geq (1 - \epsilon/K)^j$.

Proof. By definition we can write

$$g_j + h_{j-1} = h_j + r_j = h_j + s_{j+1} + Res(r_j, y_j).$$

We first claim that

$$\mathrm{Var}(h_j + s_{j+1}) \geq (1 - 2\eta) \mathrm{Var}(h_{j-1} + g_j).$$

Indeed, by Theorem 6, we have that $\mathrm{Var}(Res(r_j, y_j)) \leq 4\eta^2 \mathrm{Var}(s'_j)$ and

$$\mathrm{Var}(g_j + h_{j-1}) = \mathrm{Var}(g_j) + \mathrm{Var}(h_{j-1}) \geq \mathrm{Var}(g_j) = \mathrm{Var}(s'_j)$$

where the first equality used independence. The claim now follows by Fact 65. Our second claim is that

$$\mathrm{Var}(g_j + h_{j-1}) \geq (1 - \epsilon^2/K^2) \mathrm{Var}(s_j + h_{j-1}).$$

Indeed, we can write

$$\mathrm{Var}(g_j + h_{j-1}) = \mathrm{Var}(s'_j) + \mathrm{Var}(h_{j-1}) \geq (1 - \epsilon^2/K^2) \mathrm{Var}(s_j) + \mathrm{Var}(h_{j-1}) \geq (1 - \epsilon^2/K^2) \mathrm{Var}(s_j + h_{j-1}).$$

The desired fact follows by combining the above two claims with Part (c) of the inductive hypothesis. □

Finally, we show that the variance of the polynomial s_{j+1} will decrease by a multiplicative factor, giving (d) and completing the induction.

Claim 41. *We have* $\mathrm{Var}(s_{j+1}) \leq (1 - \epsilon^4/40)^j$.

Proof. By Theorem 6 we know that

$$\mathrm{Var}(r_j) \leq (1 - \epsilon^4/40) \mathrm{Var}(s'_j) \leq (1 - \epsilon^4/40) \mathrm{Var}(s_j) \leq (1 - \epsilon^4/40)^j$$

where we used the fact that $\mathrm{Var}(s'_2) \leq \mathrm{Var}(s_2)$ and Part (d) of the induction hypothesis. Now note that s_{j+1} is obtained from r_j by removing a subset of its terms. Therefore, $\mathrm{Var}(s_{j+1}) \leq \mathrm{Var}(r_j)$ and the claim follows. □

This completes the proof of correctness. We claim that the algorithm runs in $\mathrm{poly}(n, b, 1/\epsilon)$ time. This follows from the fact that the number of iterations of the loop is at most $K + 1 = \mathrm{poly}(1/\epsilon)$ and each iteration runs in $\mathrm{poly}(n, b, 1/\epsilon)$ time. Indeed, it is easy to verify that the running time of a given iteration is dominated by the call to the APPROXIMATE-DECOMPOSE routine. Since the input to this routine is the polynomial s'_i whose coefficients (up to rescaling) are integers whose magnitude is $\mathrm{poly}(n/\epsilon)$, it follows that the routine runs in polynomial time.

2.3.1 Proof of Lemma 31. We note that even the Kolmogorov distance version of the lemma (which is sufficient for our purposes) does not follow immediately from the Berry-Esséen theorem. One can potentially deduce our desired statement from Berry-Esséen by using an appropriate case analysis on the structure of the coefficients. However, we show that it can be deduced in a more principled way from a CLT version obtained using Stein's method. In particular, we will need the following theorem of Chatterjee:

Theorem 42. *[Cha09] Let $X \sim \mathcal{N}(0,1)^n$ and $f : \mathbb{R}^n \to \mathbb{R}$. Let $W = f(X_1, \ldots, X_n)$. Suppose that $\mathbf{E}[W] = \mu$ and $\mathrm{Var}[W] = \sigma^2$. Let $Y \sim \mathcal{N}(0,1)^n$ be independent of X. Define the random variable $T(X)$ as*

$$T(X) = \int_{t=0}^{1} \frac{1}{2\sqrt{t}} \cdot \mathbf{E}_Y \left[\sum_{i=1}^{n} \frac{\partial f(X)}{\partial X_i} \cdot \frac{\partial f(\sqrt{t}X + \sqrt{1-t}Y)}{\partial X_i} \right] dt.$$

Then we have that

$$d_{\mathrm{TV}}\left(f(X), \mathcal{N}(\mu, \sigma^2)\right) \leq \frac{\sqrt{\mathrm{Var}[T]}}{\sigma^2}.$$

Note that by Claim 18 we can assume that p is of the form $p(x) = \sum_i (\lambda_i x_i^2 + \mu_i x_i)$. We want to apply this theorem to deduce that the random variable $p(X)$ with $X \sim \mathcal{N}(0,1)^n$ is $O(\epsilon)$ close to a Gaussian with the right mean and variance. Note that $\mathbf{E}[p(X)] = \sum_{i=1}^n \lambda_i$ and $\mathrm{Var}[p(X)] = \sum_{i=1}^n (2\lambda_i^2 + \mu_i^2)$. We will apply the above theorem for the function $f(x) = p(x) = \sum_{i=1}^n (\lambda_i x_i^2 + \mu_i x_i)$. We have that $\frac{\partial p(x)}{\partial x_i} = 2\lambda_i x_i + \mu_i$. For $t \in [0,1]$ we can write

$$p(\sqrt{t}x + \sqrt{1-t}y) = \sum_{i=1}^{n} \lambda_i (\sqrt{t}x_i + \sqrt{1-t}y_i)^2 + \sum_{i=1}^{n} \mu_i (\sqrt{t}x_i + \sqrt{1-t}y_i)$$

and therefore

$$\frac{\partial p(\sqrt{t}x + \sqrt{1-t}y)}{\partial x_i} = \lambda_i (2tx_i + 2\sqrt{t(1-t)}y_i) + \mu_i \sqrt{t}.$$

Therefore,

$$\mathbf{E}_Y \left[\sum_{i=1}^{n} \frac{\partial p(X)}{\partial X_i} \cdot \frac{\partial p(\sqrt{t}X + \sqrt{1-t}Y)}{\partial X_i} \right] = \sum_{i=1}^{n} (2\lambda_i X_i + \mu_i)(2\lambda_i t X_i + \mu_i \sqrt{t})$$

and the desired integral equals

$$\begin{aligned} T = \sum_{i=1}^{n} (2\lambda_i X_i + \mu_i)(\lambda_i X_i \int_0^1 \sqrt{t} dt + \mu_i/2) &= \sum_{i=1}^{n} (2\lambda_i X_i + \mu_i)(2\lambda_i X_i/3 + \mu_i/2) \\ &= \sum_{i=1}^{n} (4\lambda_i^2 X_i^2/3 + 5/3\lambda_i \mu_i X_i + \mu_i^2/2) \end{aligned}$$

from which it follows that

$$\begin{aligned} \mathrm{Var}[T] = \sum_{i=1}^{n} \left(2(4\lambda_i^2/3)^2 + (5/3\lambda_i\mu_i)^2\right) &\leq \sum_{i=1}^{n} \left(2(5\lambda_i^2/3)^2 + (5/3\lambda_i\mu_i)^2\right) \\ &= (25/9) \sum_{i=1}^{n} \left(2\lambda_i^4 + \lambda_i^2\mu_i^2\right) \\ &\leq (25/9) \max_i \lambda_i^2 \cdot \sum_{i=1}^{n} \left(2\lambda_i^2 + \mu_i^2\right) \\ &\leq (25/9)(\epsilon^2 \,\mathrm{Var}[p]) \cdot \mathrm{Var}[p] \\ &= (25/9)(\epsilon \,\mathrm{Var}[p])^2. \end{aligned}$$

An application of the Theorem now yields that $d_{\mathrm{TV}}\left(p(X), \mathcal{N}(\mu, \sigma^2)\right) \leq 5\epsilon/3$. Since $d_{\mathrm{K}}(X,Y) \leq d_{\mathrm{TV}}(X,Y)$ for any pair of random variables X, Y the lemma follows. □

2.4 The second stage: deterministic approximate counting for degree-2 junta PTFs over Gaussians.

In this section we use the $K = \tilde{O}(1/\epsilon^4)$-variable polynomial $q(y)$ provided by Theorem 27 to do efficient deterministic approximate counting.

One possible approach is to break the polynomial $q(y)$ into two components q_+ (corresponding to those variables y_i that have $\lambda_i > 0$) and q_- (containing those y_i that have $\lambda_i < 0$). Both $q_+(y)$ and $-q_-(y)$ follow non-centered generalized chi-squared distributions, so it is conceivable that by directly analyzing the pdfs of such distributions, one could (approximately) specify the region of $\mathbb{R}^K$ over which $q_+(y) - q_-(y) \geq 0$, and then perform approximate numerical integration over that region to directly estimate $\mathbf{Pr}_{y\sim\mathcal{N}(0,1)^K}[q(y) \geq 0]$. While expressions have been given for the pdf of a generalized chi-squared distribution without the linear part, we need expressions for the pdf when there is an additional linear part. Even in the case where there is no linear part, the expressions for the pdf are somewhat forbidding (see equations (6) and (7) of [BHO09]), so an approach along these lines is somewhat unappealing.

Instead of pursuing this technically involved direction, we adopt a technically and conceptually straightforward approach based on simple dynamic programming. The algorithm `Count-Junta` that we propose and analyze is given below. Intuitively, the rounding that is performed in the first step transforms the polynomial q to one with "small integer weights." This, together with the discretization in Step 2 (which lets us approximate each independent Gaussian input with a small discrete set of values), makes it possible to perform dynamic programming to exactly count the number of satisfying assignments of the corresponding PTF.

`Count-Junta`

Input: Explicit description of a $K = \tilde{O}(1/\epsilon^4)$-variable degree-2 polynomial $q(y) = \sum_{i=1}^{K}(\lambda_i y_i^2 + \mu_i y_i) + \tau$, where each $\lambda_i, \mu_i, \tau \in \mathbb{Z}$; parameter $\epsilon > 0$.

Output: A value $v \in [0,1]$ such that

$$\left|\mathbf{Pr}_{y\in\mathcal{N}(0,1)^K}[q(y) \geq 0] - v\right| \leq \epsilon.$$

1. **Rounding.** Set $\epsilon' = \tilde{\Theta}(\epsilon^6)$ to be a value of the form $1/2^{\text{integer}}$. Let $M = \max\{|\lambda_1|, \ldots, |\lambda_K|, |\mu_1|, \ldots, |\mu_K|\}$. Let $q'(y) = \sum_{i=1}^{K}(\lambda_i' y_i^2 + \mu_i' y_i) + \tau'$ be obtained from $q(y)$ by dividing all coefficients λ_i, μ_i, τ by $2^{\lceil \log_2 M \rceil} \cdot (\epsilon'/2)$ and rounding the result to the nearest integer (so each of λ_i', μ_i' is an integer with absolute value at most $2/\epsilon'$),
2. **Discretizing each coordinate.** Set $\epsilon^\star = \Theta(\epsilon/K)$ to be of the form $1/2^{\text{integer}}$. For $i = 1, \ldots, K$: run `Discretize`$(\lambda_i', \mu_i', \epsilon^\star)$ and let $S_i = \{s_{i,1}, \ldots, s_{i,R}\}$ be the multiset that it returns.
3. **Counting via dynamic programming.** Run `DP`$(S_1, \ldots, S_K, \tau')$ and output the value it returns.

The performance guarantee of `Count-Junta` is given in the following theorem:

Theorem 43. *Algorithm* `Count-Junta` *is given as input an explicit description of a polynomial* $q(y) = \sum_{i=1}^{K}(\lambda_i y_i^2 + \mu_i y_i) + \tau$ *and* $\epsilon > 0$ *where* $\lambda_i, \mu_i \in \mathbb{Z}$, $K = \tilde{O}(1/\epsilon^4)$, *and each coefficient* λ_i, μ_i, τ *is a* B*-bit integer. It runs (deterministically) in* $O(KB) \cdot \mathrm{polylog}(1/\epsilon) + \mathrm{poly}(1/\epsilon)$ *bit operations and outputs a value* $v \in [0,1]$ *such that*

$$\left|\mathbf{Pr}_{y\sim\mathcal{N}(0,1)^K}[q(y) \geq 0] - v\right| \leq \epsilon. \tag{5}$$

2.4.1 Proof of Theorem 43

Runtime analysis. It is straightforward to verify that Step 1 (rounding) can be carried out, and the integers $\lambda'_i, \mu'_i, \tau'$ obtained, in $O(KB) \cdot \log(1/\epsilon)$ bit operations (note that the coefficients λ'_i, μ'_i are obtained from the original values simply by discarding all but the $O(\log(1/\epsilon))$ most significant bits). Note that each of λ'_i, μ'_i is a $O(\log(1/\epsilon))$-bit integer.

The claimed running time of `Count-Junta` then follows easily from the running times established in Lemma 45 and the analysis of `DP` given below.

Correctness. We start with a simple lemma establishing that the "rounding" step, Step 1, does not change the acceptance probability of the PTF by more than a small amount:

Lemma 44. *We have*

$$\left|\mathbf{Pr}_{y\sim\mathcal{N}(0,1)^K}[q(y) \geq 0] - \mathbf{Pr}_{y\sim\mathcal{N}(0,1)^K}[q'(y) \geq 0]\right| \leq \epsilon/2. \tag{6}$$

Proof. This is a standard argument using concentration (tail bounds for degree-2 polynomials in Gaussian random variables) and anti-concentration (Carbery-Wright). Let $a(y) = (2^{\lceil \log_2 M \rceil} \cdot (\epsilon'/2))q'(y) - q(y)$. We have that $\mathrm{sign}(q(y)) \neq \mathrm{sign}(q'(y))$ only if at least one of the following events occurs: (i) $|a(y)| \geq c\epsilon^2\mathrm{Var}(q)$, or (ii) $|q(y)| \leq c\epsilon^2\mathrm{Var}(q)$ (where c is an absolute constant). For (i), we observe that $a(y)$ has at most $2K+1$ coefficients that are each at most $\epsilon' M$ in magnitude and hence $|\mathbf{E}_{y\sim\mathcal{N}(0,1)^K}[a(y)]| \leq (K+1)\epsilon' M$ while $\mathrm{Var}(q) \geq M$ (recall that at last one of $|\lambda_i|, |\mu_i|$ is at least M). So $\mathrm{Var}(a) \leq \sqrt{2K+1} \cdot \epsilon' \cdot M$, and by Theorem 68 (the "degree-2 Chernoff bound") $\mathbf{Pr}_{y\sim\mathcal{N}(0,1)^K}[|a(y)| \geq c\epsilon^2\mathrm{Var}(q)] \leq \epsilon/4$. On the other hand, Theorem 69 gives us that $\mathbf{Pr}[|q(y)| \leq c\epsilon^2\mathrm{Var}(q)] \leq O(\sqrt{c\epsilon^2}) \leq \epsilon/4$. The lemma follows by a union bound. □

We now turn to Step 2 of the algorithm, in which each distribution $\lambda'_i y_i^2 + \mu'_i y_i$, $y_i \sim \mathcal{N}(0,1)$, is converted to a nearby discrete distribution. The procedure `Discretize` is given below:

`Discretize`

Input: Integers ℓ, m; real value $\epsilon^\star = 1/2^{\text{integer}}$.

Output: A multiset $S = \{s_1, \ldots, s_R\}$ of $R = 2/\epsilon^\star$ values such that the distribution $\mathcal{D}_S$ satisfies $d_{\mathrm{K}}(\mathcal{D}_S, \ell Y^2 + mY) \leq \epsilon^\star$, where $Y \sim \mathcal{N}(0,1)$.

1. Let $t_1 < \cdots < t_R$ be the real values given by Fact 46 when its algorithm is run on input parameter $\epsilon^\star$.
2. Output the multiset $S = \{s_1, \ldots, s_R\}$ where $s_i = \ell t_i^2 + m t_i$ for all i.

Our key lemma here says that $\mathcal{D}_S$ is Kolmogorov-close to the univariate degree-2 Gaussian polynomial $\ell y_i^2 + m y_i$:

Lemma 45. *Given $\ell, m, \epsilon^\star$ as specified in* `Discretize`*, the procedure* `Discretize`$(\ell, m, \epsilon^\star)$ *outputs a multiset $S = \{s_1, \ldots, s_R\}$ of $R = 8/\epsilon^\star$ values such that the distribution $\mathcal{D}_S$ satisfies*

$$d_{\mathrm{K}}(\mathcal{D}_S, \ell Y^2 + mY) \leq \epsilon^\star \quad \textit{where } Y \sim \mathcal{N}(0,1). \tag{7}$$

Moreover, if $\epsilon^\star$ is of the form $1/2^{\text{integer}}$ and $|\ell|, |m| \leq L$, then the running time is $\tilde{O}(1/(\epsilon^\star)^4 + \log(L)/\epsilon^\star)$ and each element s_i is of the form $a_i/(C'/(\epsilon^\star)^2)$ where C' is a (positive integer) absolute constant and a_i is an integer satisfying $|a_i| = \tilde{O}(L/(\epsilon^\star)^2)$.

Proof. We will use the following basic fact which says that it is easy to construct a high-accuracy ϵ-cover for $\mathcal{N}(0,1)$ w.r.t. Kolmogorov distance:

Fact 46. *There is a deterministic procedure with the following property: given as input any value $\epsilon^\star = 1/2^j$ where $j \geq 0$ is an integer, the procedure runs in time $\tilde{O}(1/(\epsilon^\star)^4)$ bit operations and outputs a set $T = \{t_1, \ldots, t_{4/\epsilon^\star}\}$ of $4/\epsilon^\star$ real values such that $d_{\mathrm{K}}(\mathcal{D}_T, Z) \leq \epsilon^\star$ where $Z \sim \mathcal{N}(0,1)$. Moreover each t_i is a rational number of the form integer$/(C/\epsilon^\star)$, where $C > 0$ is some absolute constant and the numerator is at most $\tilde{O}(1/\epsilon^\star)$.*

Proof. (A range of different proofs could be given for this fact; we chose this one both for its simplicity of exposition and because the computational overhead of the dynamic programming routine outweighs the runtime of this procedure, so its exact asymptotic running time is not too important.)

The deterministic procedure is very simple: it explicitly computes the values $p_j := \binom{n}{j}/2^n$ for $j = 0, 1, \ldots, n$ where $n = \Theta(1/(\epsilon^\star)^2)$ is odd (we can and do take it to additionally be a perfect square). It is easy to see that all n of these values can be computed using a total of $\tilde{O}(n^2) = \tilde{O}(1/(\epsilon^\star)^4)$ bit operations (each of the $n = O(1/(\epsilon^\star)^2)$ binomial coefficients can be computed from the previous one by performing a constant number of multiplications and divisions of an $n = O(1/(\epsilon^\star)^2)$-bit number with an $O(\log(1/\epsilon^\star))$-bit number). For $r \in \mathbb{R}$, let $A_r = \{z \in \mathbb{Z} : r \geq z \geq 0\}$ and let $P_r = \sum_{z \in A_r} p_z$. The Berry-Esséen theorem implies that for all $r = 0, 1, \ldots, n$ we have

$$|P_r - \mathbf{Pr}_{Z \sim \mathcal{N}(0,1)}[Z \leq ((r - (n+1)/2)/\sqrt{n})]| \leq 1/\sqrt{n} \leq \epsilon^\star/10.$$

Let $\mathcal{D}_{\mathcal{B}}$ denote the binomial distribution $B(n, 1/2)$. Consider the distribution $\widetilde{\mathcal{D}_{\mathcal{B}}} = (\mathcal{D}_{\mathcal{B}} - (n+1)/2)/\sqrt{n}$. Then, note that $d_K(\mathcal{N}(0,1), \widetilde{\mathcal{D}_{\mathcal{B}}}) \leq \epsilon^*/10$. Note that every element in the support of $\widetilde{\mathcal{D}_{\mathcal{B}}}$ is a rational number whose numerator is an integer (bounded by C/ϵ^{*2}) and the denominator is C/ϵ^*. Further, as $1/\sqrt{n} \leq \epsilon^\star/10$ for any $z \in \mathbb{R}$, $\mathbf{Pr}[\widetilde{\mathcal{D}_{\mathcal{B}}} = z] \leq \epsilon^*/10$. This gives a straightforward method to obtain a distribution $\mathcal{D}'$ supported on a $2/\epsilon^*$ points such that $d_K(\mathcal{D}', \widetilde{\mathcal{D}_{\mathcal{B}}}) \leq \epsilon^*/2$. Rounding each value to a multiple of $\epsilon^*/4$ (and carefully moving the mass around), it is easy to obtain the set T of size $4/\epsilon^*$ points such that $d_{TV}(\mathcal{D}', \mathcal{D}_T) \leq \epsilon^*/4$. As a consequence,

$$d_K(\mathcal{D}_T, \mathcal{N}(0,1)) \leq d_{TV}(\mathcal{D}', \mathcal{D}_T) + d_K(\mathcal{D}', \widetilde{\mathcal{D}_{\mathcal{B}}}) + d_K(\mathcal{N}(0,1), \widetilde{\mathcal{D}_{\mathcal{B}}}) \leq \epsilon.$$

The claim about the representation of points in T follows from the fact that they are a subset of the points in the support of $\widetilde{\mathcal{D}_{\mathcal{B}}}$ for which we proved this property. This concludes the proof of Fact 46 □

We next make the following claim for Kolmogorov distance for functions of random variables which is an analogue of the "data processing inequality" for total variation distance. First, we make the following definition.

Definition 47. *Let $f : \mathbb{R} \to \mathbb{R}$ be a differentiable function. It is said to be k-modal if there are at most k points $z_1, \ldots, z_k$ such that $\frac{df}{dx}|_{x=z_i} = 0$.*

Claim 48. *Let f be a unimodal function and let X and Y be real valued random variables. Then $d_K(f(X), f(Y)) \leq 2d_K(X, Y)$.*

Proof. For any real number t, there are two possibilities :

(i) There are real numbers $t_1 \leq t_2$ such that $f(x) \geq t$ if and only if $x \in [t_1, t_2]$.

(ii) There are real numbers $t_1 \leq t_2$ such that $f(x) \geq t$ if and only if $x \notin (t_1, t_2)$.

In case (i),

$$\mathbf{Pr}[f(X) \in [t, \infty)] = \mathbf{Pr}[X \in [t_1, t_2]] \quad \text{and} \quad \mathbf{Pr}[f(Y) \in [t, \infty)] = \mathbf{Pr}[Y \in [t_1, t_2]].$$

As a consequence,

$$\begin{aligned} d_K(f(X), f(Y)) &= \sup_{t\in\mathbb{R}} |\mathbf{Pr}[f(X) \in [t,\infty)] - \mathbf{Pr}[f(Y) \in [t,\infty)]| \\ &\leq \sup_{t_1,t_2\in\mathbb{R}} |\mathbf{Pr}[X \in [t_1,t_2]] - \mathbf{Pr}[Y \in [t_1,t_2]]| \leq 2d_K(X,Y). \end{aligned}$$

In case (ii),

$$\mathbf{Pr}[f(X) \in [t,\infty)] = \mathbf{Pr}[X \notin [t_1,t_2]] \quad \text{and} \quad \mathbf{Pr}[f(Y) \in [t,\infty)] = \mathbf{Pr}[Y \notin [t_1,t_2]].$$

As a consequence,

$$\begin{aligned} d_K(f(X), f(Y)) &= \sup_{t\in\mathbb{R}} |\mathbf{Pr}[f(X) \in [t,\infty)] - \mathbf{Pr}[f(Y) \in [t,\infty)]| \\ &\leq \sup_{t_1,t_2\in\mathbb{R}} |\mathbf{Pr}[X \notin [t_1,t_2]] - \mathbf{Pr}[Y \notin [t_1,t_2]]| \leq 2d_K(X,Y). \end{aligned}$$

This proves the stated claim. □

We first apply Fact 46 to construct a distribution $\mathcal{D}_T$ such that $d_K(\mathcal{D}_T, \mathcal{N}(0,1)) \leq \epsilon^*/2$. We then observe that the function $f(t) = \ell t^2 + mt$ is unimodal and hence $\mathcal{D}_S = f(\mathcal{D}_T)$, we have $d_K(\mathcal{D}_S, \ell Y^2 + mY) \leq \epsilon^*$ where $Y \sim \mathcal{N}(0,1)$. This concludes the proof of Lemma 45. □

With Lemma 45 in hand it is simple to obtain the following:

Lemma 49. *Let $X_1, \ldots, X_K$ be independent random variables where $X_i \sim \mathcal{D}_{S_i}$ (see Step 2 of* `Count-Junta`*). Then*

$$d_{\mathrm{K}}\left(\sum_{i=1}^{K} X_i, \sum_{i=1}^{K} (\lambda_i' y_i^2 + \mu_i' y_i)\right) \leq K\epsilon^\star \leq \epsilon/2 \quad \textit{where } y = (y_1, \ldots, y_k) \sim \mathcal{N}(0,1)^K. \tag{8}$$

Proof. This follows immediately from (7) and the sub-additivity property of Kolmogorov distance: for $A_1, \ldots, A_n$ independent random variables and $B_1, \ldots, B_n$ independent random variables,

$$d_{\mathrm{K}}(\sum_{i=1}^{n} A_i, \sum_{i=1}^{n} B_i) \leq \sum_{i=1}^{n} d_{\mathrm{K}}(A_i, B_i).$$

(See e.g. Equation (4.2.3) of [BK01] for an explicit statement; this also follows easily from the triangle inequality and the basic bound that $d_{\mathrm{K}}(X_1+Y, X_2+Y) \leq d_{\mathrm{K}}(X_1, X_2)$ for X_1, X_2, Y independent random variables.) □

Finally we turn to Step 3, the dynamic programming. The algorithm `DP` uses dynamic programming to compute the exact value of $\mathbf{Pr}[X_1 + \cdots + X_K + \tau' \geq 0]$, where $X_1, \ldots, X_K$ are independent random variables with X_i distributed according to $\mathcal{D}_{S_i}$. Observe that by Lemma 45, for any $1 \leq i \leq K$ the partial sum $X_1 + \cdots + X_i$ must always be of the form $c/(C'/(\epsilon^\star)^2)$ where c is an integer satisfying $|c| \leq \tilde{O}((i/\epsilon)/(\epsilon/K)^2) = \tilde{O}((K/\epsilon)^3) = N$, where $N = \mathrm{poly}(1/\epsilon)$. Thus the dynamic program has a variable $v_{i,n}$ for each pair (i,n) where $1 \leq i \leq K$ and $|n| \leq N$; the value of variable $v_{i,n}$ is $\mathbf{Pr}[X_1 + \cdots + X_i = n/(C'/(\epsilon^\star)^2)$. Given the values of variables $v_{i-1,n}$ for all n and the multiset S_i it is straightforward to compute the values of variables $v_{i,n}$ for all n. Since each nonzero probability under any distribution $\mathcal{D}_{S_i}$ is a rational number with both numerator and denominator $O(\log(1/\epsilon))$ bits long, the bit complexity of every value $v_{i,n}$ is at most $\tilde{O}(K)$ bits, and since there are KN entries in the table, the overall running time of `DP` is $\tilde{O}(K^2 N) = \mathrm{poly}(1/\epsilon)$ bit operations. The procedure $\texttt{DP}(S_1, \ldots, S_K, \tau')$ returns the value $v = \sum_{n=0}^{N} v_{K,n}$, which by the above discussion is exactly equal to

$$\mathbf{Pr}_{(X_1,\ldots,X_K)\sim\mathcal{D}_{S_1}\times\cdots\times\mathcal{D}_{S_K}}[X_1 + \cdots + X_K + \tau' \geq 0]. \tag{9}$$

Now equations (6) and (8) together establish that the value (9) output by `Count-Junta` satisfies (5) as required for correctness. This concludes the proof of Theorem 43. □

2.4.2 Putting it all together. Combining algorithms `Construct-Junta-PTF` and `Count-Junta`, Theorems 27 and 43, give our main result for degree-2 PTFs over Gaussian variables:

Theorem 50. *[Deterministic approximate counting of degree-2 PTFs over Gaussians] There is an algorithm with the following properties: It takes as input an explicit description of an n-variable degree-2 polynomial p with b-bit integer coefficients and a value $\epsilon > 0$. It runs (deterministically) in time $\mathrm{poly}(n, b, 1/\epsilon)$ and outputs a value $v \in [0,1]$ such that*

$$\left|\mathbf{Pr}_{x\sim\mathcal{N}(0,1)^n}[p(x) \geq 0] - v\right| \leq \epsilon. \tag{10}$$

3 Deterministic approximate counting for degree-2 polynomials over $\{-1,1\}^n$

In this section we extend the results of the previous section to give a deterministic algorithm for approximately counting satisfying assignments of a degree-2 PTF over the Boolean hypercube. We prove the following:

Theorem 51. *[Deterministic approximate counting of degree-2 PTFs over $\{-1,1\}^n$] There is an algorithm with the following properties: It takes as input an explicit description of an n-variable degree-2 multilinear polynomial p with b-bit integer coefficients and a value $\epsilon > 0$. It outputs a value $v \in [0,1]$ such that $\left|\mathbf{Pr}_{x\sim\{-1,1\}^n}[p(x) \geq 0] - v\right| \leq \epsilon$ and runs (deterministically) in time $\mathrm{poly}(n, b, 2^{\tilde{O}(1/\epsilon^9)})$.*

The main ingredient in the proof of Theorem 51 is the "regularity lemma for PTFs" of [DSTW10]. As stated in [DSTW10], this lemma is an existential statement which says that every degree-d PTF over $\{-1,1\}^n$ can be expressed as a shallow decision tree with variables at the internal nodes and degree-d PTFs at the leaves, such that a random path in the decision tree is quite likely to reach a leaf that has a "close-to-regular" PTF.

The precise statement is:

Lemma 52. *[Theorem 1 of [DSTW10]] Let $f(x) = \mathrm{sign}(p(x))$ be any degree-d PTF. Fix any $\tau > 0$. Then f is equivalent to a decision tree $\mathcal{T}$, of depth*

$$\mathrm{depth}(d,\tau) := \frac{1}{\tau} \cdot \left(d \log \frac{1}{\tau}\right)^{O(d)}$$

with variables at the internal nodes and a degree-d PTF $f_\rho = \mathrm{sign}(p_\rho)$ at each leaf ρ, with the following property: with probability at least $1-\tau$, a random path from the root reaches a leaf ρ such that f_ρ is τ-close to some τ-regular degree-d PTF.

Intuitively, this lemma is helpful for us because for regular polynomials g we can simply use $\mathbf{Pr}_{x\sim\mathcal{N}(0,1)^n}[g(x) \geq 0]$ (which we can approximate efficiently using Theorem 50) as a proxy for $\mathbf{Pr}_{x\sim\{-1,1\}^n}[g(x) \geq 0]$ and incur only small error. However, to use the lemma in our context we need a deterministic algorithm which efficiently constructs the decision tree. While it is not clear from the lemma statement above, fortunately the [DSTW10] proof in fact provides such an algorithm, as we explain below.

3.1 Proof of Theorem 51 As we describe below, the argument of [DSTW10] actually gives the following lemma, which is an effective version of Lemma 52 above.

Theorem 53. *Let $p(x_1, \ldots, x_n)$ be an input multilinear degree-d PTF with b-bit integer coefficients. Fix any $\tau > 0$. There is an algorithm $A_{\mathrm{Construct-Tree}}$ which, on input p and a parameter $\tau > 0$, runs in $\mathrm{poly}(n, b, 2^{\mathrm{depth}(d,\tau)})$ time and outputs a decision tree $\mathcal{T}$ of depth*

$$\mathrm{depth}(d,\tau) := \frac{1}{\tau} \cdot \left(d \log \frac{1}{\tau}\right)^{O(d)},$$

where each internal node of the tree is labeled with a variable and each leaf ρ of the tree is labeled with a pair $(p_\rho, \mathrm{label}(\rho))$ where $\mathrm{label}(\rho) \in \{+1, -1,$ "fail", "regular"$\}$. The tree $\mathcal{T}$ has the following properties:

1. *Every input $x \in \{-1,1\}^n$ to the tree reaches a leaf ρ such that $p(x) = p_\rho(x)$;*
2. *If leaf ρ has $\mathrm{label}(\rho) \in \{+1,-1\}$ then $\mathbf{Pr}_{x\in\{-1,1\}^n}[\mathrm{sign}(p_\rho(x)) \neq \mathrm{label}(\rho)] \leq \tau$;*
3. *If leaf ρ has $\mathrm{label}(\rho) =$ "regular" then p_ρ is τ-regular; and*
4. *With probability at most τ, a random path from the root reaches a leaf ρ such that $\mathrm{label}(\rho) =$ "fail".*

We prove Theorem 53 in Section 3.2 below, but first we show how it gives Theorem 51.

Proof of Theorem 51, assuming Theorem 53: The algorithm for approximating $\mathbf{Pr}_{x\in\{-1,1\}^n}[p(x) \geq 0]$ to $\pm\epsilon$ works as follows. It first runs $A_{\mathrm{Construct-Tree}}$ with its "τ" parameter set to $\Theta(\epsilon^9)$ to construct the decision tree $\mathcal{T}$. It then iterates over all leaves ρ of the tree. For each leaf ρ at depth d_ρ that has $\mathrm{label}(\rho) = +1$ it adds 2^{-d_ρ} to v (which is initially zero), and for each leaf ρ at depth d_ρ that has $\mathrm{label}(\rho) =$ "regular" it runs the algorithm of Theorem 50 on p_ρ (with its "ϵ" parameter set to $\Theta(\epsilon^9)$) to obtain a value $v_\rho \in [0,1]$ and adds $v_\rho \cdot 2^{-d_\rho}$ to v. It outputs the value $v \in [0,1]$ thus obtained.

Theorems 53 and 50 imply that the running time is as claimed. To establish correctness of the algorithm we will use the "invariance principle" of [MOO10]:

Theorem 54 ([MOO10])**.** *Let $p(x) = \sum_{S\subseteq[n],|S|\leq d} p_S x_S$ be a degree-d multilinear polynomial with $\mathrm{Var}[p] = 1$. Then $\sup_{t\in\mathbb{R}} |\mathbf{Pr}_{x\in\{-1,1\}^n}[p(x) \leq t] - \mathbf{Pr}_{\mathcal{G}\sim\mathcal{N}(0,1)^n}[p(\mathcal{G}) \leq t]| \leq O(d\tau^{1/(4d+1)})$, where τ is such that each coordinate $i \in [n]$ has $\mathrm{Inf}_i(p) \leq \tau$.*

By Theorem 53, the leaves of $\mathcal{T}$ that are marked $+1$, -1 or "fail" collectively contribute at most $\Theta(\epsilon^9) \leq \epsilon/2$ to the error of the output value v. Theorem 54 implies that each leaf ρ at depth d_ρ that is marked "regular" contributes at most $2^{-d_\rho} \cdot \epsilon/2$ to the error, so the total contribution from all such leaves is at most $\epsilon/2$. This concludes the proof of Theorem 51. □

3.2 Proof of Theorem 53: The [DSTW10] construction. Theorem 1 of [DSTW10] establishes the existence of the claimed decision tree $\mathcal{T}$ by analyzing an iterative procedure that constructs $\mathcal{T}$. Inspection of this procedure reveals that it can be straightforwardly implemented by an efficient deterministic algorithm. We first provide some details of the procedure below and then analyze its running time.

The iterative procedure uses parameters $\beta = \tau$, and $\tilde{\tau}$ chosen such that $\tau = \tilde{\tau} \cdot (C' d \ln d \ln(1/\tilde{\tau}))^d$ where C' is a universal constant (see Lemma 12 and the proof of Theorem 1 of [DSTW10]). It works to construct $\mathcal{T}$ by processing each node of the tree that has not yet been declared a leaf of $\mathcal{T}$ in the manner that we now describe.

Processing a single node: Consider a given node that is currently a leaf of the partially-constructed decision tree but has not yet been declared a leaf of $\mathcal{T}$. Call such a node ρ; it corresponds to a restriction of some of the variables, and such a node is currently labeled with the restricted polynomial p_ρ. (At the beginning of the procedure the node ρ is the root of $\mathcal{T}$, corresponding to the empty restriction that fixes no variables, and the polynomial p_ρ is simply p.) Let us write $p_\rho(x) = \sum_{|S|\leq d, S\subset[n]} p_{\rho,S} x_S$ where $x_S = \prod_{i\in S} x_i$.

If the depth d_ρ of ρ is greater than $\mathrm{depth}(d,\tau)$ then the procedure declares ρ to be a leaf of $\mathcal{T}$ and labels it with the pair $(p_\rho,$ "fail"$)$. Otherwise, the procedure first computes $\mathrm{Inf}_i(p_\rho) = \sum_{S\ni i} (p_{\rho,S})^2$ for all $i = 1,\ldots,n$ and $\mathrm{Inf}(p_\rho) = \sum_{i=1}^n \mathrm{Inf}_i(p_\rho)$. It sorts the variables in decreasing order of influence (for notational convenience we shall suppose that $\mathrm{Inf}_1(p_\rho) \geq \mathrm{Inf}_2(p_\rho) \geq \cdots$), and operates as follows:

1. If $\mathrm{Inf}_1(p_\rho) \leq \tau \cdot \mathrm{Inf}(p_\rho)$ then the node ρ is declared a leaf of $\mathcal{T}$ and is labeled with the pair $(p_\rho,$ "regular"$)$.

Otherwise, let $\mathrm{ci}_\tau(p_\rho)$, the τ-critical index of p_ρ, be the least i such that

$$\mathrm{Inf}_{i+1}(p_\rho) \leq \tau \cdot \sum_{j=i+1}^{n} \mathrm{Inf}_j(p_\rho).$$

Let $\alpha = \Theta(d \log\log(1/\tau) + d \log d)$.

2. If $\mathrm{ci}_\tau(p_\rho) \geq \alpha/\tilde{\tau}$ then the procedure "expands" node ρ by replacing it with a complete decision tree of depth $\alpha/\tilde{\tau}$, where all internal nodes at the i-th level of this tree contain variable x_i. For each new restriction ρ' (an extension of ρ) resulting from this expansion the procedure computes $p_{\rho'}$ and labels node ρ' with that polynomial.

   Let us write $p_\rho(x) = p_\rho(x_H, x_T) = p'_\rho(x_H) + q_\rho(x_H, x_T)$ where $p'_\rho(x_H)$ is the truncation of p containing only the monomials all of whose variables lie in the set $H = \{1, \ldots, \mathrm{ci}_\tau(p_\rho)\}$.

   It is easy to see that the constant term of the polynomial $p_{\rho'}$ is precisely $p'_\rho(\rho')$. The procedure computes $|p'_\rho(\rho')|$ and $\|q_\rho(\rho, x_T)\|_2$. If $|p'_\rho(\rho')| \geq t^\star \stackrel{\text{def}}{=} 1/(2C^d)$ (here $C > 0$ is a universal constant; see Definition 2 and the discussion at the end of Section 1.2 of [DSTW10]) and $\|q_\rho(\rho, x_T)\|_2 \leq t^\star \cdot (\Theta(\log(1/\beta)))^{-d/2}$ then the procedure declares ρ' to be a leaf and labels it with the pair $(p_{\rho'}, \mathrm{sign}(p'_\rho(\rho')))$.

3. If $\mathrm{ci}_\tau(p_\rho) < \alpha/\tilde{\tau}$ then the procedure expands node ρ by replacing it with a complete decision tree of depth $\mathrm{ci}_\tau(p_\rho)$, where again all internal nodes at the i-th level of this tree contain variable x_i. As in the previous case, for each new restriction ρ' resulting from this expansion the procedure computes $p_{\rho'}$ and labels node ρ' with that polynomial.

It is clear that the above procedure constructs a tree $\mathcal{T}$ that satisfies properties (1), (2) and (3) of Theorem 53. The analysis of [DSTW10] establishes that the tree $\mathcal{T}$ satisfies property (4).

Finally, the running time bound is easily verified from the description of the algorithm and the fact that the input is a degree-d PTF with b-bit integer coefficients.

3.3 Fully polynomial deterministic approximate counting for regular degree-2 PTFs. As a special case of the above analysis we easily obtain the following result for regular PTFs:

Theorem 55. *[Deterministic approximate counting of regular degree-2 PTFs over $\{-1,1\}^n$] Let p be an n-variable degree-2 multilinear polynomial p with b-bit integer coefficients that is $O(\epsilon^9)$-regular. Then the above algorithm runs in deterministic time* $\mathrm{poly}(n, b, 1/\epsilon)$ *and outputs a value $v \in [0,1]$ such that*

$$\left|\mathbf{Pr}_{x \sim \{-1,1\}^n}[p(x) \geq 0] - v\right| \leq \epsilon.$$

This is because if p is already ϵ^9-regular then the tree-construction procedure will halt immediately at the root.

4 A deterministic FPT approximation algorithm for absolute moments

In this section we prove Theorem 3. Note that since we are working with polynomials over the domain $\{-1,1\}^n$, it is sufficient to consider multilinear polynomials.

We begin with the following easy observation:

Observation 56. *Let $q(x)$ be a degree-2 multilinear polynomial over $\{-1,1\}^n$ that has $\mathbf{E}_{x \in \{-1,1\}^n}[q(x)^2] = 1$. Then for all $k \geq 1$ we have that the k-th raw moment $\mathbf{E}_{x \in \{-1,1\}^n}[|q(x)|^k]$ is at least c where $c > 0$ is some universal constant.*

Proof. For $k \geq 2$ this is an immediate consequence of the monotonicity of norms, which gives us

$$1 = \mathbf{E}[|q(x)|^2]^{1/2} \leq \mathbf{E}[|q(x)|^k]^{1/k} \quad \text{for } k \geq 2.$$

For $k = 1$ the desired statement is an easy consequence of Theorem 4.1 of [AH11]. □

Given an input degree-2 multilinear polynomial $p(x_1, \ldots, x_n)$, we may divide by $\|p\|_2$ to obtain a scaled version $q = p/\|p\|_2$ which has $\|q\|_2 = 1$. Observation 56 implies that an additive $\pm\epsilon$-approximation to $\mathbf{E}[|q(x)|^k]$ is also a multiplicative $(1 \pm O(\epsilon))$-approximation to $\mathbf{E}[|q(x)|^k]$. Multiplying the approximation by $\|p\|_2^k$ we obtain a multiplicative $(1 \pm O(\epsilon))$-approximation to $\mathbf{E}[|p(x)|^k]$. Thus to prove Theorem 3 it suffices to give a deterministic algorithm which finds an additive $\pm\epsilon$-approximation to $\mathbf{E}[|q(x)|^k]$ for degree-2 polynomials with $\|q\|_2 = 1$. We do this by proving Theorem 57 below:

Theorem 57. *Let $p(x)$ be an input multilinear degree-2 PTF with b-bit integer coefficients. Let $q(x) = p(x)/\|p\|_2$ so $\|q\| = 1$.*

There is an algorithm A_{moment} that, on input $k \in \mathbb{Z}^+$, p, and $\epsilon > 0$, runs in time $\text{poly}\left(n, b, 2^{\tilde{O}((k \log k \log(1/\epsilon))^{9k}/\epsilon^9)}\right)$ *and outputs a value $\tilde{\mu}_k$ such that*

$$\left|\tilde{\mu}_k - \mathbf{E}_{x \in \{-1,1\}}[|q(x)|^k]\right| \leq \epsilon.$$

4.1 Proof of Theorem 57. The idea behind the theorem is very simple. Since we can estimate $\mathbf{Pr}_{x \sim \{-1,1\}^n}[q(x) \geq t]$ for any t of our choosing, we can get a detailed picture of where the probability mass of the random variable $q(x)$ lies (for x uniform over $\{-1,1\}^n$), and with this detailed picture it is straightforward to estimate the k-th moment.

We now enter into the details. For $j \in \mathbb{Z}$ let $q_{j,\Delta}$ denote $\mathbf{Pr}_{x \in \{-1,1\}^n}[q(x) \in [(j-1)\Delta, j\Delta]]$.

We start with the following claim which follows immediately from Theorem 51:

Claim 58. *Fix any $0 < \Delta < 1$ and any degree-2 multilinear polynomial p with b-bit integer coefficients. As above let $q(x) = p(x)/\|p\|_2$. There is a $\text{poly}(n, b, 2^{\tilde{O}(1/\epsilon^9)})$-time algorithm which, given as input p, $0 < \epsilon < 1/2$, $\Delta \in \mathbb{R}$ and $j \in \mathbb{Z}$, outputs a value $\tilde{q}_{j,\Delta}$ such that*

$$\tilde{q}_{j,\Delta} \in [q_{j,\Delta} - \epsilon, q_{j,\Delta} + \epsilon].$$

We recall the following tail bound for polynomials in $\{-1,1\}$ random variables which follows easily from Theorem 68:

Theorem 59. *Let q be a degree-2 polynomial with $\|q\|_2 = 1$. For any $z \geq 0$ we have*

$$\mathbf{Pr}_{x \in \{-1,1\}^n}[|q(x)| \geq z] \leq O(1) \cdot \exp(-\Omega(z)).$$

Fix $\Delta > 0$. Let $\gamma_q(t)$ denote the probability mass function of $q(x)$ when x is distributed uniformly over $\{-1,1\}^n$. We may write the k-th absolute moment as

$$\mathbf{E}_{x \in \{-1,1\}^n}[|q(x)|^k] = \int_{-\infty}^{\infty} |t|^k \gamma_q(t) dt. \tag{11}$$

For $R \geq 1$ we have

$$\int_{R-1}^{R} |t|^k \gamma_q(t) dt = \int_{(R-1}^{R} t^k \gamma_q(t) dt \leq R^k \, \mathbf{Pr}_{x \in \{-1,1\}^n}[q(x) \geq R-1] \leq O(R^k e^{-\Omega(R)}),$$

so for integer $M \geq 1$ we have

$$\int_{t=M}^{\infty} |t|^k \gamma_q(t) dt \leq \sum_{R=M}^{\infty} O(R^k e^{-\Omega(R)})$$

which is at most $\epsilon/8$ for $M = O(k \log k \log \frac{1}{\epsilon})$ (fix M to this value). Doing similar analysis for $R \leq -1$ gives that

$$\mathbf{E}_{x \in \{-1,1\}^n}[|q(x)|^k] = \int_{-M}^{M} |t|^k \gamma_q(t) dt \pm \epsilon' \quad \text{where } \epsilon' < \epsilon/4.$$

So to approximate $\mathbf{E}_{x \sim \{-1,1\}^n}[|q(x)|^k]$ to an additive $\pm\epsilon$, it suffices to approximate $\int_{-M}^{M} |t|^k \gamma_q(t) dt$ to an additive $\pm 3\epsilon/4$.

Fix $\Delta = (\epsilon/4)^{1/k}\tau/k$, and consider the interval $[(j-1)\Delta, j\Delta]$ where $j \geq k/\tau$ for some $0 < \tau < 1$. Recalling that $q_{j,\Delta} = \mathbf{Pr}_{x \in \{-1,1\}^n}[q(x) \in [(j-1)\Delta, j\Delta]]$, we have

$$\int_{(j-1)\Delta}^{j\Delta} |t|^k \gamma_q(t) dt \in [((j-1)\Delta)^k q_{j,\Delta}, (j\Delta)^k q_{j,\Delta}] = q_{j,\Delta} \cdot \Delta^k \cdot [(j-1)^k, j^k].$$

Since

$$j^k - (j-1)^k = j^k \left(1 - \left(1 - \frac{1}{j}\right)^k\right) \leq j^k \left(1 - \left(1 - \frac{\tau}{k}\right)^k\right) \leq \tau j^k,$$

we have

$$\int_{(j-1)\Delta}^{j\Delta} |t|^k \gamma_q(t) dt \in [1-\tau, 1] \cdot |j\Delta|^k q_{j,\Delta}. \tag{12}$$

A similar analysis gives that we likewise have

$$\int_{-j\Delta}^{(-j+1)\Delta} |t|^k \gamma_q(t) dt \in [1-\tau, 1] \cdot |j\Delta|^k q_{j,\Delta}. \tag{13}$$

Finally, we observe that

$$\int_{-(k/\tau-1)\Delta}^{(k/\tau-1)\Delta} |t|^k \gamma_q(t) dt < ((k/\tau)\Delta)^k = \epsilon/4, \tag{14}$$

where we used $\Delta = (\epsilon/4)^{1/k}\tau/k$ for the final step.

With the above ingredients in hand it is clear how we shall deterministically estimate the k-th moment $\mathbf{E}_{x \in \{-1,1\}^n}[|q(x)|^k]$. Given as input an integer $k \geq 1$, a real value $0 < \epsilon < 1$, and a degree-2 multilinear polynomial q with $\|q\| = 1$, the algorithm for estimating this moment works as follows:

1. Set $M = O(k \log k \log \frac{1}{\epsilon})$, set $\tau = \epsilon/(4M^k)$, and set $\Delta = 1/2^r$ where r is the largest value such that $1/2^r \leq (\epsilon/4)^{1/k}\tau/k$.

2. For $j = (k/\tau - 1)$ to M/Δ: compute a $\pm\tau/4$-accurate additive estimate $\tilde{q}_{j,\Delta}$ of $q_{j,\Delta}$ (using Claim 58) and sum the values $|j\Delta|^k \tilde{q}_{j,\Delta}$ to obtain E_+.

   Similarly, for $j = -(k/\tau - 1)$ to $-M/\Delta$: compute a $\pm\tau/4$-accurate additive estimate $\tilde{q}_{j,\Delta}$ of $q_{j,\Delta}$ (using Claim 58) and sum the values $|j\Delta|^k \tilde{q}_{j,\Delta}$ to obtain E_-.

3. Output $E_+ + E_-$.

It is easy to see that the above algorithm runs in time

$$(M/\Delta) \cdot \mathrm{poly}(n, b, 2^{\tilde{O}(1/\tau^9)}) = \mathrm{poly}\left(n, b, 2^{\tilde{O}((k \log k \log(1/\epsilon))^{9k}/\epsilon^9)}\right).$$

To prove correctness recall that we need to show that $E_+ + E_-$ is within $\pm 3\epsilon/4$ of $\int_{-M}^{M} |t|^k \gamma_q(t)dt$. Recalling (14), it suffices to show that E_+ and E_- are each within $\pm\epsilon/4$ of $\int_{-M}^{-(k/\tau-1)\Delta} |t|^k \gamma_q(t)dt$ and $\int_{(k/\tau-1)\Delta}^{M} |t|^k \gamma_q(t)dt$ respectively. Recalling our choice of τ, it follows easily from (12) that

$$\left| E_+ - \int_{-M}^{-(k/\tau-1)\Delta} |t|^k \gamma_q(t)dt \right| \leq \tau M^k = \epsilon/4.$$

An identical argument works for E_- and the other integral, and we are done with the proof. □

References

[AH11] Per Austrin and Johan Håstad. Randomly supported independence and resistance. *SIAM J. Comput.*, 40(1):1–27, 2011.

[APL07] H. Aziz, M. Paterson, and D. Leech. Efficient algorithm for designing weighted voting games. In *IEEE Intl. Multitopic Conf.*, pages 1–6, 2007.

[AW85] M. Ajtai and A. Wigderson. Deterministic simulation of probabilistic constant depth circuits. In *Proc. 26th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 11–19, 1985.

[Bec75] W. Beckner. Inequalities in fourier analysis. *Annals of Mathematics*, 102:159–182, 1975.

[BG08] A. Bandyopadhyay and D. Gamarnik. Counting without sampling: Asymptotics of the log-partition function for certain statistical physics models. *Random Struct. Algorithms*, 33(4):452–479, 2008.

[BGK+07] M. Bayati, D. Gamarnik, D. Katz, C. Nair, and P. Tetali. Simple deterministic approximation algorithms for counting matchings. In *Proc. 39th Annual ACM Symposium on Theory of Computing (STOC)*, pages 122–127, 2007.

[BHO09] E. Björnson, D. Hammarwall, and B. Ottersten. Exploiting quantized channel norm feedback through conditional statistics in arbitrarily correlated mimo systems. *IEEE Trans. on Signal Processing*, 57(10):4027–4041, 2009.

[BK01] M. V. Boutsikas and M. V. Koutras. Compound Poisson Approximation for Sums of Dependent Random Variables. In Ch.A. Charalambides, M. V. Koutras, and N. Balakrishnan, editors, *Probability and Statistical Models with Applications: A volume in honor of Prof. T. Cacoullos*, pages 63–86. Chapman and Hall/CRC Press, 2001.

[Bon70] A. Bonami. Etude des coefficients fourier des fonctiones de $l^p(g)$. *Ann. Inst. Fourier (Grenoble)*, 20(2):335–402, 1970.

[Cha09] S. Chatterjee. Fluctuations of eigenvalues and second-order Poincaré inequalities. *Probability Theory and Related Fields*, 143:1–40, 2009.

[CW01] A. Carbery and J. Wright. Distributional and L^q norm inequalities for polynomials over convex bodies in R^n. *Mathematical Research Letters*, 8(3):233–248, 2001.

[DDFS12] A. De, I. Diakonikolas, V. Feldman, and R. Servedio. Near-optimal solutions for the Chow Parameters Problem and low-weight approximation of halfspaces. In *Proc. 44th ACM Symposium on Theory of Computing (STOC)*, pages 729–746, 2012.

[DDS12] Anindya De, Ilias Diakonikolas, and Rocco A. Servedio. The inverse shapley value problem. In *ICALP (1)*, pages 266–277, 2012.

[DDS13] A. De, I. Diakonikolas, and R. Servedio. Deterministic approximate counting for juntas of degree-2 polynomial threshold functions. manuscript, 2013.

[DGJ+09] I. Diakoniokolas, P. Gopalan, R. Jaiswal, R. Servedio, and E. Viola. Bounded independence fools halfspaces. In *Proc. 50th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 171–180, 2009.

[DHK+10] Ilias Diakonikolas, Prahladh Harsha, Adam Klivans, Raghu Meka, Prasad Raghavendra, Rocco A. Servedio, and Li-Yang Tan. Bounding the average sensitivity and noise sensitivity of polynomial threshold functions. In *STOC*, pages 533–542, 2010.

[DOSW11] I. Diakonikolas, R. O'Donnell, R. Servedio, and Y. Wu. Hardness results for agnostically learning low-degree polynomial threshold functions. In *SODA*, pages 1590–1606, 2011.

[DRST09] I. Diakonikolas, P. Raghavendra, R. Servedio, and L.-Y. Tan. Average sensitivity and noise sensitivity of polynomial threshold functions, 2009. Available at http://arxiv.org/abs/0909.5011.

[DS13] A. De and R. Servedio. Efficient deterministic approximate counting for low-degree polynomial threshold functions. manuscript, 2013.

[DSTW10] I. Diakonikolas, R. Servedio, L.-Y. Tan, and A. Wan. A regularity lemma, and low-weight approximators, for low-degree polynomial threshold functions. In *CCC*, pages 211–222, 2010.

[FGRW09] Vitaly Feldman, Venkatesan Guruswami, Prasad Raghavendra, and Yi Wu. Agnostic learning of monomials by halfspaces is hard. In *FOCS*, pages 385–394, 2009.

[GHR92] M. Goldmann, J. Håstad, and A. Razborov. Majority gates vs. general weighted threshold gates. *Computational Complexity*, 2:277–300, 1992.

[GKM+11] Parikshit Gopalan, Adam Klivans, Raghu Meka, Daniel Stefankovic, Santosh Vempala, and Eric Vigoda. An fptas for #knapsack and related counting problems. In *FOCS*, pages 817–826, 2011.

[GMR13] P. Gopalan, R. Meka, and O. Reingold. DNF sparsification and a faster deterministic counting algorithm. *Computational Complexity*, 22(2):275–310, 2013.

[Hås94] J. Håstad. On the size of weights for threshold gates. *SIAM Journal on Discrete Mathematics*, 7(3):484–492, 1994.

[HKM09] P. Harsha, A. Klivans, and R. Meka. Bounding the sensitivity of polynomial threshold functions. Available at http://arxiv.org/abs/0909.5175, 2009.

[Jan97] S. Janson. *Gaussian Hilbert Spaces*. Cambridge University Press, Cambridge, UK, 1997.

[JSV01] Mark Jerrum, Alistair Sinclair, and Eric Vigoda. A polynomial-time approximation algorithm for the permanent of a matrix with non-negative entries. In *Proceedings of the Thirty-Third Annual Symposium on Theory of Computing*, pages 712–721, 2001.

[Kan10] D.M. Kane. The Gaussian surface area and noise sensitivity of degree-d polynomial threshold functions. In *CCC*, pages 205–210, 2010.

[Kan11a] Daniel M. Kane. k-independent gaussians fool polynomial threshold functions. In *IEEE Conference on Computational Complexity*, pages 252–261, 2011.

[Kan11b] Daniel M. Kane. A small prg for polynomial threshold functions of gaussians. In *FOCS*, pages 257–266, 2011.

[Kan12a] Daniel M. Kane. The correct exponent for the gotsman-linial conjecture. *CoRR*, abs/1210.1283, 2012.

[Kan12b] Daniel M. Kane. A pseudorandom generator for polynomial threshold functions of gaussian with subpolynomial seed length. *CoRR*, abs/1210.1280, 2012.

[Kan12c] Daniel M. Kane. A structure theorem for poorly anticoncentrated gaussian chaoses and applications to the study of polynomial threshold functions. In *FOCS*, pages 91–100, 2012.

[Kan13] D. Kane. Personal communication, 2013.

[KKMS08] A. Kalai, A. Klivans, Y. Mansour, and R. Servedio. Agnostically learning halfspaces. *SIAM Journal on Computing*, 37(6):1777–1805, 2008.

[KRS12] Zohar Shay Karnin, Yuval Rabani, and Amir Shpilka. Explicit dimension reduction and its applications. *SIAM J. Comput.*, 41(1):219–249, 2012.

[LLY13] L. Li, P. Lu, and Y. Yin. Correlation Decay up to Uniqueness in Spin Systems. In *SODA*, pages 67–84, 2013.

[LV96] M. Luby and B. Velickovic. On deterministic approximation of DNF. *Algorithmica*, 16(4/5):415–433, 1996.

[MK61] J. Myhill and W. Kautz. On the size of weights required for linear-input switching functions. *IRE Trans. on Electronic Computers*, EC10(2):288–290, 1961.

[MOO10] E. Mossel, R. O'Donnell, and K. K. Oleszkiewicz. Noise stability of functions with low influences: Invariance and optimality. *Annals of Mathematics*, 171:295–341, 2010.

[MP68] M. Minsky and S. Papert. *Perceptrons: an introduction to computational geometry*. MIT Press, Cambridge, MA, 1968.

[MTT61] S. Muroga, I. Toda, and S. Takasu. Theory of majority switching elements. *J. Franklin Institute*, 271:376–418, 1961.

[Mur71] S. Muroga. *Threshold logic and its applications*. Wiley-Interscience, New York, 1971.

[MZ10] Raghu Meka and David Zuckerman. Pseudorandom generators for polynomial threshold functions. In *STOC*, pages 427–436, 2010.

[Orp92] P. Orponen. Neural networks and complexity theory. In *Proceedings of the 17th International Symposium on Mathematical Foundations of Computer Science*, pages 50–61, 1992.

[OS11] R. O'Donnell and R. Servedio. The Chow Parameters Problem. *SIAM J. on Comput.*, 40(1):165–199, 2011.

[Pap94] Christos H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[Pod09] V. V. Podolskii. Perceptrons of large weight. *Problems of Information Transmission*, 45(1):46–53, 2009.

[RS09] Y. Rabani and A. Shpilka. Explicit construction of a small epsilon-net for linear threshold functions. In *Proc. 41st Annual ACM Symposium on Theory of Computing (STOC)*, pages 649–658, 2009.

[Ser07] R. Servedio. Every linear threshold function has a low-weight approximator. *Comput. Complexity*, 16(2):180–209, 2007.

[She08] Alexander A. Sherstov. Halfspace matrices. *Computational Complexity*, 17(2):149–178, 2008.

[She09] A. Sherstov. The intersection of two halfspaces has high threshold degree. In *Proc. 50th IEEE Symposium on Foundations of Computer Science (FOCS)*, 2009.

[SSSS11] Shai Shalev-Shwartz, Ohad Shamir, and Karthik Sridharan. Learning kernel-based halfspaces with the 0-1 loss. *SIAM J. Comput.*, 40(6):1623–1646, 2011.

[Vio09] E. Viola. The Sum of *D* Small-Bias Generators Fools Polynomials of Degree *D*. *Computational Complexity*, 18(2):209–217, 2009.

[Vis13] N. Vishnoi. $Lx = b$. 2013. Available at "http://research.microsoft.com/en-us/um/people/nvishno/Site/Lxb-Web.pdf".

[Wei06] D. Weitz. Counting independent sets up to the tree threshold. In *STOC*, pages 140–149, 2006.

A Definitions and Background

In this section we record the preliminaries we will need.

A.1 Basic Linear Algebra Facts. In this section we record some basic facts from linear algebra that will be crucial for our proofs.

Definition 60. *(orthogonal matrix) A matrix $Q \in \mathbb{R}^{n\times n}$ is said to be* orthogonal *if both its columns and its rows comprise a set of n orthonormal unit vectors. Equivalently, a matrix $Q \in \mathbb{R}^{n\times n}$ is orthogonal if its transpose is equal to its inverse, i.e., $Q^T = Q^{-1}$.*

Theorem 61. *(Spectral Theorem) If $A \in \mathbb{R}^{n\times n}$ is symmetric, there exists an orthogonal $Q \in \mathbb{R}^{n\times n}$ and a diagonal matrix $\Lambda \in \mathbb{R}^{n\times n}$ such that $A = Q\Lambda Q^T$. The diagonal entries of Λ are the eigenvalues of A and the columns of Q are the corresponding eigenvectors. That is, we can write $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$, $Q = [u^{(1)} \mid \ldots \mid u^{(n)}]$, with $u^{(i)} \cdot u^{(j)} = \delta_{ij}$, and $Au^{(i)} = \lambda_i u^{(i)}$ for all $i \in [n]$. The expression $A = Q\Lambda Q^T$ of a symmetric matrix in terms of its eigenvalues and eigenvectors is referred to as* the spectral decomposition of A.

Definition 62. *Given a degree-2 polynomial $p : \mathbb{R}^n \to \mathbb{R}$ defined as $p(x) = \sum_{1\le i\le j\le n} a_{ij}x_ix_j + \sum_{1\le i\le n} b_ix_i + C$, we define the (symmetric) matrix A corresponding to its quadratic part as : $A_{ij} = a_{ij}(1/2 + \delta_{ij}/2)$. With this definition, it is easy to see that $x^T \cdot A \cdot x = \sum_{1\le i\le j\le n} a_{ij}x_ix_j$ for the vector $x = (x_1, \ldots, x_n)$.*

Throughout the paper we adopt the convention that the eigenvalues $\lambda_1, \ldots, \lambda_n$ of a real symmetric matrix A satisfy $|\lambda_1| \geq \cdots \geq |\lambda_n|$. We sometimes write $\lambda_{\max}(A)$ to denote λ_1, and we sometimes write $\lambda_i(p)$ to refer to the i-th eigenvalue of the matrix A defined based on p as described above.

Definition 63. *For a real symmetric matrix A, with (real) eigenvalues $\lambda_1, \ldots, \lambda_n$ such that $|\lambda_1| \geq |\lambda_2| \geq \ldots \geq |\lambda_n|$ we define:*

- *The Frobenius norm of A is $\|A\|_F \stackrel{def}{=} \sqrt{\sum_{i,j} A_{i,j}^2}$.*
- *The trace of A is $\mathrm{tr}(A) \stackrel{def}{=} \sum_{i=1}^n A_{ii}$. We have that $\mathrm{tr}(A) = \sum_{i=1}^n \lambda_i$.*

We recall the following fact:

Fact 64. *Let $A \in \mathbb{R}^{n \times n}$ be symmetric with eigenvalues $\lambda_1, \ldots, \lambda_n$. The eigenvalues of the matrix A^k are $\lambda_1^k, \ldots, \lambda_n^k$. Since $\|A\|_F = \sqrt{\mathrm{tr}(A^2)}$, $\|A\|_F = \sqrt{\sum_{i=1}^n \lambda_i^2}$.*

A.2 Basic Probabilistic Facts. Given an r-element multiset $S = \{s_1, \ldots, s_r\}$ we write $\mathcal{D}_S$ to denote the distribution which is uniform over the elements of S (so if an element v occurs j times in S we have $\mathbf{Pr}_{x \sim \mathcal{D}_S}[x = v] = j/r$.).

Fact 65. *Let P and Q be real valued random variables such that $\mathrm{Var}(P) = \alpha$ and $\mathrm{Var}(Q) = \eta^2\alpha$. Then $(1 + 2\eta + \eta^2)\alpha \geq \mathrm{Var}(P - Q) \geq (1 - 2\eta + \eta^2)\alpha$.*

Proof.

$$\begin{aligned} \mathrm{Var}(P-Q) &= \mathbf{E}[(P-Q)^2] - (\mathbf{E}[P-Q])^2 = \mathrm{Var}(P) + \mathrm{Var}(Q) - 2\mathbf{E}[PQ] + 2\mathbf{E}[P]\mathbf{E}[Q] \\ &= \mathrm{Var}(P) + \mathrm{Var}(Q) - 2\,\mathrm{Cov}(P,Q) \\ &= (1+\eta^2)\alpha - 2\,\mathrm{Cov}(P,Q). \end{aligned}$$

Now the desired inequalities follow using the simple inequality $|\mathrm{Cov}(P,Q)| \leq \sqrt{\mathrm{Var}(P)}\sqrt{\mathrm{Var}(Q)}$ which is a consequence of Cauchy-Schwarz. □

We recall the Berry-Esseen theorem, which states that under suitable conditions a sum of independent random variables converges (in Kolmogorov distance) to a normal distribution:

Theorem 66. *(Berry-Esséen) Let $\{X_i\}_{i=1}^n$ be a set of independent random variables satisfying $\mathbf{E}[X_i] = 0$ for all $i \in [n]$, $\sqrt{\sum_i \mathbf{E}[X_i^2]} = \sigma$, and $\sum_i \mathbf{E}[|X_i|^3] = \rho_3$. Let $S = \sum_i X_i/\sigma$ and let F denote the cumulative distribution function (cdf) of S. Then $\sup_x |F(x) - \Phi(x)| \leq \rho_3/\sigma^3$ where Φ denotes the cdf of the standard gaussian random variable.*

Fact 67. *Let $\mu_1, \mu_2 \in \mathbb{R}$ and $0 < \sigma_1^2 \leq \sigma_2^2$. Then,*

$$d_{\mathrm{TV}}\left(\mathcal{N}(\mu_1, \sigma_1^2), \mathcal{N}(\mu_2, \sigma_2^2)\right) \leq \frac{1}{2}\left(\frac{|\mu_1 - \mu_2|}{\sigma_1} + \frac{\sigma_2^2 - \sigma_1^2}{\sigma_1^2}\right).$$

A.3 Useful Facts about Polynomials. We view $\mathbb{R}^n$ as a probability space endowed with the standard n-dimensional Gaussian measure. For a square-integrable function $f : \mathbb{R}^n \to \mathbb{R}$ and $r \geq 1$, we let $\|f\|_r$ denote $(\mathbf{E}_{x\sim\mathcal{N}^n}[|f(x)^r|])^{1/r}$. We will need a concentration bound for low-degree polynomials over independent Gaussians.

Theorem 68 ("degree-d Chernoff bound", [Jan97])**.** *Let $p : \mathbb{R}^n \to \mathbb{R}$ be a degree-d polynomial. For any $t > e^d$, we have*

$$\mathbf{Pr}_{x\sim\mathcal{N}(0,1)^n}[|p(x) - \mathbf{E}[p(x)]| > t \cdot \sqrt{\mathrm{Var}(p(x))}] \leq de^{-\Omega(t^{2/d})}.$$

The same bound holds for x drawn uniformly from $\{-1, 1\}^n$.

We will also use the following anti-concentration bound for degree-d polynomials over Gaussians:

Theorem 69 ([CW01])**.** *Let $p : \mathbb{R}^n \to \mathbb{R}$ be a degree-d polynomial that is not identically 0. Then for all $\epsilon > 0$ and all $\theta \in \mathbb{R}$, we have*

$$\mathbf{Pr}_{x\sim\mathcal{N}(0,1)^n}\left[|p(x) - \theta| < \epsilon\sqrt{\mathrm{Var}(p)}\right] \leq O(d\epsilon^{1/d}).$$

Definition 70. *Let $f : \{-1, 1\}^n \to \mathbb{R}$. The influence of the i^{th} coordinate on f under the uniform measure (denoted by $\mathrm{Inf}_i(f)$) is defined as*

$$\mathrm{Inf}_i(f) = \mathop{\mathbf{E}}_{x_i\in\{-1,1\}}[\mathrm{Var}_{x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n\in\{-1,1\}} f(x_1, \ldots, x_n)].$$

The total influence of a function f (denoted by $\mathrm{Inf}(f)$) is defined as $\sum_{i=1}^n \mathrm{Inf}_i(f)$.

We now define an extension of the notion of "critical index" previously used in several works on linear and polynomial threshold functions [Ser07, OS11, DRST09].

Definition 71. *Given a pair of sequences of non-negative numbers $\{c_i\}_{i=1}^n$ and $\{d_i\}_{i=1}^n$ where additionally the sequence $\{c_i\}_{i=1}^n$ is non-increasing, the τ-*critical index *of the pair is defined to be the smallest $0 \leq i \leq n - 1$ such that*

$$\frac{c_{i+1}}{\sum_{j>i}(c_j + d_j)} \leq \tau.$$

In case there is no such number, we define the critical index to be ∞. The sequence $\{c_i\}_{i=1}^n$ is called the "main sequence" and the sequence $\{d_i\}_{i=1}^n$ is called the "auxiliary sequence".

The following is a simple consequence of the definition of critical index.

Fact 72. *Given a pair of sequences of non-negative numbers, $\{c_i\}_{i=1}^n$ and $\{d_i\}_{i=1}^n$, if the τ-critical index of a sequence is j, then $\sum_{i=j+1}^n (c_i + d_i) < (1 - \tau)^j \cdot (\sum_{\ell=1}^n c_i + d_i)$.*

As noted earlier, special cases of this definition have appeared in previous work on polynomial threshold functions. Below, we recall the notion of the critical index of a polynomial that appeared previously in the work of Diakonikolas *et al.* [DRST09]:

Definition 73. *([DRST09]) Let $p : \mathbb{R}^n \to \mathbb{R}$ and $\tau > 0$. Assume the variables are ordered such that $\mathrm{Inf}_i(p) \geq \mathrm{Inf}_{i+1}(p)$ for all $i \in [n - 1]$. The τ-*critical index *of f is defined to be the τ-critical index of the pair of sequences $\{\mathrm{Inf}_i(p)\}_{i=1}^n$ and $\{0\}_{i=1}^n$ where $\{0\}_{i=1}^n$ is the auxiliary sequence.*

B Hardness of computing absolute moments

In this section, we show that for any fixed odd k, it is $\#P$-hard to exactly compute the k^{th} absolute moment of a degree two multilinear polynomial with $\{0,1\}$ coefficients over the uniform distribution on the hypercube.

Theorem 74. *Given a degree two multilinear polynomial $p : \mathbb{R}^n \to \mathbb{R}$ with $\{0,1\}$ coefficients, it is $\#P$-hard (under Turing reductions) to compute $\mathbf{E}_{x\in\{-1,1\}^n}[|p(x)|^k]$, the k^{th} absolute moment of p over the uniform distribution on $\{-1,1\}^n$, for $k = O(1)$.*

Proof. We begin by recalling that given an undirected graph $G = (V, E)$, it is NP-hard to find the size of the MAX-CUT in G. In fact, if the size of the MAX-CUT in G is ν, then it is $\#P$-hard to find the number of cuts in G whose size is ν (see Papadimitriou [Pap94]).

Let $|V| = n$ and $|E| = m$. We consider the polynomial $q_{G,\mathrm{CUT}} : \mathbb{R}^n \to \mathbb{R}$ defined as $q_{G,\mathrm{CUT}}(x) = (|E| - \sum_{\{i,j\}\in E} x_i x_j)/2$; recall from the introduction that on input $x \in \{-1,1\}^n$ the value $q_{G,\mathrm{CUT}}(x)$ equals the number of edges in the cut corresponding to x. Consequently $q_{G,\mathrm{CUT}}(x)) \in [0,\ldots,m]$ for every $x \in \{-1,1\}^n$. Let $\nu^*_G = \max_{x\in\{-1,1\}^n}[q_{G,\mathrm{CUT}}(x)]$ denote the size of the MAX-CUT in G. Thus, it is $\#P$-hard to compute the size of the following set

$$\{x \in \{-1,1\}^n : q_{g,\mathrm{CUT}}(x) = \nu^*_G\}.$$

We next observe that for any fixed $k = O(1)$, there is a poly(n)-time algorithm to compute the k-th raw moment $\mathbf{E}_{x\in\{-1,1\}^n}[p(x)^k]$ of a given degree-2 input polynomial $p(x)$. The algorithm works simply by expanding out $p(x)^k$ (in time $n^{O(k)}$), performing multilinear reduction, and outputting the constant term; its correctness follows from the fact that $\mathbf{E}_{x\in\{-1,1\}^n}[x_S] = 0$ for every $S \neq \emptyset$.

Suppose A is a poly(n)-time algorithm that, on input a degree-2 polynomial $p(x)$, outputs the value $\mathbf{E}_{x\in\{-1,1\}^n}[|p(x)|^k]$. Given such an algorithm we can efficiently compute $|\{x \in \{-1,1\}^n : q_{g,\mathrm{CUT}}(x) = \nu^*_G\}|$ as follows: For $\ell = m, m-1, \ldots$ successively compute $\mathbf{E}[|\ell - q_{G,\mathrm{CUT}}(x)|^k]$ (using algorithm A) and $\mathbf{E}[(\ell - q_{G,\mathrm{CUT}}(x))^k]$ (as described above). Let ℓ^* be the largest value in $\{m, m-1, \ldots, 0\}$ such that $\mathbf{E}[|\ell^* - q_{G,\mathrm{CUT}}(x)|^k] \neq \mathbf{E}[(\ell^* - q_{G,\mathrm{CUT}}(x))^k]$. Output the value

$$2^{n-1}\left(\mathbf{E}[|\ell^* - q_{G,\mathrm{CUT}}(x)|^k] - \mathbf{E}[(\ell^* - q_{G,\mathrm{CUT}}(x))^k]\right).$$

It is clear that the above-described algorithm runs in poly(n) time. To verify correctness, first consider a value of ℓ such that $\ell \geq \nu^*_G$. For such an ℓ we have that $|\ell - q_{G,\mathrm{CUT}}(x)|^k = (\ell - q_{G,\mathrm{CUT}}(x))^k$ for all $x \in \{-1,1\}^n$, and hence the raw moment $\mathbf{E}[(\ell - q_{G,\mathrm{CUT}}(x))^k]$ will equal the absolute moment $\mathbf{E}[(\ell - q_{G,\mathrm{CUT}}(x))^k]$. On the other hand, for $\ell = \nu^*_G - 1$, we have that all cuts of size $0,\ldots,\nu^*_G - 1$ contribute the same amount to $\mathbf{E}[|\ell - q_{G,\mathrm{CUT}}(x)|^k]$ and to $\mathbf{E}[(\ell - q_{G,\mathrm{CUT}}(x))^k]$, but each cut of size precisely ν^*_G contributes $1/2^n$ to the absolute moment and $-1/2^n$ to the raw moment. As a result, we get that $2^{n-1}\left(\mathbf{E}[|\ell - q_{G,\mathrm{CUT}}(x)|^k] - \mathbf{E}[(\ell - q_{G,\mathrm{CUT}}(x))^k]\right)$ is precisely the number of cuts of size ν^*_G, and the theorem is proved. □