

Efficient Deterministic Approximate Counting for Low-Degree Polynomial Threshold Functions

[Extended Abstract]

Anindya De^{*}
Institute for Advanced Study
Princeton, NJ
anindya@math.ias.edu

Rocco A. Servedio[†]
Columbia University
New York, NY
rocco@cs.columbia.edu

ABSTRACT

We give a *deterministic* algorithm for approximately counting satisfying assignments of a degree- d polynomial threshold function (PTF). Given a degree- d input polynomial $p(x)$ over \mathbb{R}^n and a parameter $\epsilon > 0$, our algorithm approximates $\Pr_{x \sim \{-1, 1\}^n} [p(x) \geq 0]$ to within an additive $\pm\epsilon$ in time $O_{d,\epsilon}(1) \cdot \text{poly}(n^d)$. (Since it is NP-hard to determine whether the above probability is nonzero, any sort of efficient multiplicative approximation is almost certainly impossible even for randomized algorithms.) Note that the running time of our algorithm (as a function of n^d , the number of coefficients of a degree- d PTF) is a *fixed* polynomial. The fastest previous algorithm for this problem [Kan12b], based on constructions of unconditional pseudorandom generators for degree- d PTFs, runs in time $n^{O_{d,c}(1) \cdot \epsilon^{-c}}$ for all $c > 0$.

The key novel technical contributions of this work are

- A new multivariate central limit theorem, proved using tools from Malliavin calculus and Stein's Method. This new CLT shows that any collection of Gaussian polynomials with small eigenvalues must have a joint distribution which is very close to a multidimensional Gaussian distribution.
- A new decomposition of low-degree multilinear polynomials over Gaussian inputs. Roughly speaking we show that (up to some small error) any such polynomial can be decomposed into a bounded number of multilinear polynomials all of which have extremely small eigenvalues.

^{*}This material is based upon work supported by the National Science Foundation under agreements Princeton University Prime Award No. CCF-0832797 and Sub-contract No. 00001583. Some of this work was done while the author was at the Simons Institute, UC Berkeley.

[†]Supported by NSF grants CCF-1115703 and CCF-1319788.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org. STOC '14, May 31 - June 03 2014, New York, NY, USA
Copyright 2014 ACM 978-1-4503-2710-7/14/05 \$15.00
<http://dx.doi.org/10.1145/2591796.2591800>

We use these new ingredients to give a deterministic algorithm for a Gaussian-space version of the approximate counting problem, and then employ standard techniques for working with low-degree PTFs (invariance principles and regularity lemmas) to reduce the original approximate counting problem over the Boolean hypercube to the Gaussian version.

As an application of our result, we give the first deterministic fixed-parameter tractable algorithm for the following moment approximation problem: given a degree- d polynomial $p(x_1, \dots, x_n)$ over $\{-1, 1\}^n$, a positive integer k and an error parameter ϵ , output a $(1 \pm \epsilon)$ -multiplicatively accurate estimate to $\mathbf{E}_{x \sim \{-1, 1\}^n} [|p(x)|^k]$. Our algorithm runs in time $O_{d,\epsilon,k}(1) \cdot \text{poly}(n^d)$.

Categories and Subject Descriptors

F.2.2 [Nonnumerical Algorithms and Problems]: Computations on Discrete Structures

General Terms

Theory

Keywords

Polynomial Threshold Function; Approximate Counting; Derandomization

1. INTRODUCTION

For decades a major research goal in computational complexity has been to understand the computational power of randomization – and perhaps to show that randomness does not actually augment the abilities of polynomial-time algorithms. Towards this end, an important research goal within unconditional derandomization has been the development of *deterministic approximate counting algorithms*. This line of research started with the work of Ajtai and Wigderson [AW85], who gave a sub-exponential time deterministic algorithm to approximately count the number of satisfying assignments of a constant-depth circuit. Since this early work many other classes of Boolean functions have been studied from this perspective, including DNF formulas, low-degree $GF[2]$ polynomials, linear threshold functions, and degree-2 polynomial threshold functions [LVW93, LV96, Tre04, GMR13, Vio09, GKM⁺11, DDS13a, DDS13b].

In this paper we study the problem of deterministic approximate counting for *degree- d polynomial threshold func-*

tions (PTFs). A degree- d PTF is a Boolean function $f : \{-1, 1\}^n \rightarrow \{-1, 1\}$ defined by $f(x) = \text{sign}(p(x))$ where $p : \{-1, 1\}^n \rightarrow \mathbb{R}$ is a degree- d polynomial. In the special case where $d = 1$, degree- d PTFs are often referred to as *linear threshold functions* (LTFs). While LTFs and low-degree PTFs have been researched for decades (see e.g. [MK61, MTT61, MP68, Mur71, GHR92, Orp92, Häs94, Pod09] and many other works), they have recently been the focus of renewed research attention in fields such as concrete complexity theory [She08, She09, DHK⁺10, Kan10, Kan12c, Kan12a, KRS12], computational learning theory [KKMS08, SSSS11, DOSW11, DDFS12], voting theory [APL07, DDS12] and others.

Our main result. The main contribution of this paper is to give a fixed polynomial time deterministic approximate counting algorithm for degree- d PTFs. We prove the following theorem:

THEOREM 1. *There is a deterministic algorithm A with the following properties: Let A be given as input a degree- d polynomial p over $\{-1, 1\}^n$ and an accuracy parameter $\epsilon > 0$. Algorithm A runs in time $O_{d,\epsilon}(1) \cdot \text{poly}(n^d)$ and outputs a value $\tilde{v} \in [0, 1]$ such that $|\tilde{v} - \Pr_{x \in \{-1, 1\}^n}[p(x) \geq 0]| \leq \epsilon$.*

Note that the above result guarantees an *additive* approximation to the desired probability. While additive approximation is not as strong as multiplicative approximation, one should recall that the problem of determining whether $\Pr_{x \in \{-1, 1\}^n}[p(x) \geq 0]$ is nonzero is well known to be NP-hard, even for degree-2 polynomials and even if all nonconstant monomials in p are restricted to have coefficients from $\{0, 1\}$ (this can be shown via a simple reduction from Max-Cut). Thus unless $\text{NP} \subseteq \text{RP}$ no efficient algorithm, even allowing randomness, can give any multiplicative approximation to $\Pr_{x \sim \{-1, 1\}^n}[p(x) \geq 0]$. Given this, additive approximation is a natural goal.

Related work. Several previous works have given $\text{poly}(n^d)$ -time deterministic approximate counting algorithms for DNF formulas of width d (see e.g. [Tre04, LV96, GMR13] as well as the approach of [AW85] augmented with the almost t -wise independent distributions of [NN93], as discussed in [Tre04]). Degree- d PTFs are of course a broad generalization of width- d DNF formulas, and the algorithms for width- d DNFs referenced above do not extend to degree- d PTFs.

The $d = 1$ case for degree- d PTFs (i.e. LTFs) is qualitatively different from $d > 1$. For $d = 1$ the satisfiability problem is trivial, so one may reasonably hope for a multiplicatively $(1 \pm \epsilon)$ -accurate deterministic approximate counting algorithm. Indeed such an algorithm, running in fully polynomial time $\text{poly}(n, 1/\epsilon)$, was given by Gopalan *et al.* and Stefankovic *et al.* in [GKM⁺11]. For $d \geq 2$, however, as noted above additive approximation is the best one can hope for, even for randomized algorithms. The only previous deterministic approximate counting results for degree- d PTFs for general d follow from known constructions of *unconditional pseudorandom generators* (PRGs) for degree- d PTFs. The first such construction was given by Meka and Zuckerman [MZ10], whose PRG yielded an $n^{O_d(1) \cdot \text{poly}(1/\epsilon^d)}$ -time deterministic approximate counting algorithm. Followup works by Kane [Kan11a, Kan11b, Kan12b] improved the parameters of these PRGs, with the strongest construction from [Kan12b] (for PTFs over Gaussian inputs) giving

a $n^{O_{d,\epsilon}(1) \cdot \epsilon^{-c}}$ -time algorithm. Thus these prior works do not give a fixed polynomial-time algorithm.

For the special case of $d = 2$, in separate work [DDS13a] the authors have given a deterministic algorithm for degree-2 PTFs that runs in time $\text{poly}(n, 2^{\text{poly}(1/\epsilon)})$. In [DDS13b] the authors extended the [DDS13a] result and gave an algorithm that does deterministic approximate counting for any $O(1)$ -junta of degree-2 PTFs. As we explain in detail in the rest of this introduction, much more sophisticated techniques and analyses are required to obtain the results of the current paper for general d . These include a new central limit theorem in probability theory based on Malliavin calculus and Stein’s method, and an intricate new decomposition procedure that goes well beyond the decomposition approaches employed in [DDS13a, DDS13b].

Our approach. The main step in proving Theorem 1 is to give a deterministic approximate counting algorithm for the *standard Gaussian distribution* $N(0, 1)^n$ over \mathbb{R}^n rather than the uniform distribution over $\{-1, 1\}^n$. The key result that gives us Theorem 1 is the following:

THEOREM 2. *There is a deterministic algorithm A with the following properties: Let A be given as input a degree- d polynomial p over \mathbb{R}^n and an accuracy parameter $\epsilon > 0$. Algorithm A runs in time $O_{d,\epsilon}(1) \cdot \text{poly}(n^d)$ and outputs a value $\tilde{v} \in [0, 1]$ such that $|\tilde{v} - \Pr_{x \sim N(0, 1)^n}[p(x) \geq 0]| \leq \epsilon$.*

Theorem 1 follows from Theorem 2 using the invariance principle of [MOO10] and the “regularity lemma” for polynomial threshold functions from [DSTW10]. The arguments that give Theorem 1 from Theorem 2 are essentially identical to the ones used in [DDS13a], so we omit them in this extended abstract (see the full version). In the rest of this introduction we describe the main ideas behind the proof of Theorem 2; as explained below, there are two main contributions.

First contribution: A new multivariate CLT. Our first contribution is a new multidimensional central limit theorem that we establish for r -tuples of degree- d Gaussian polynomials, i.e. r -tuples $(p_1(x), \dots, p_r(x))$ where each p_i is a degree- d polynomial and $x \sim N(0, 1)^n$. This CLT states that if each p_i has “small eigenvalues” (as defined at the start of Section 3), then the joint distribution converges to the multidimensional Normal distribution \mathcal{G} over \mathbb{R}^r whose mean and covariance match (p_1, \dots, p_r) . The closeness here is with respect to “test functions” that have globally bounded second derivatives; see Theorem 7 for a detailed statement of our CLT. In Section 5 we use tools from mollification to go from the aforementioned kind of “closeness” to the kind of closeness which is required to analyze polynomial threshold functions.

Comparing with previous work, the degree-2 case [DDS13a] required a CLT for a single degree-2 Gaussian polynomial. The main technical ingredient of the [DDS13a] proof was a result of Chatterjee [Cha09]. [DDS13b] established the $d = 2$ case of our multidimensional CLT via a relatively straightforward analysis (requiring just basic linear algebra) of the central limit theorem from [NPR10]. We note that in the $d = 2$ case it is clear what is the correct notion of the eigenvalues of a degree-2 polynomial, namely the eigenvalues of the quadratic form. In contrast, it is far from clear what is the correct notion of the eigenvalues of a degree- d

polynomial, especially since we require a notion that enables both a CLT and a decomposition as described later. (We note that the tensor eigenvalue definitions that are employed in [FW95, CS13, Lat06] do not appear to be suitable for our purposes.) Based on discussions with experts [Lat13, Nou13, Led13, Ole13], even the $d = 2$ version of our multidimensional CLT was not previously known, let alone the far more general version of the CLT which we establish in this work.

It is instructive to consider our CLT in the context of a result of Latala [Lat06], which shows that (a somewhat different notion of) tensor eigenvalues can be used to bound the growth of moments of degree- d Gaussian polynomials. However, the moment bounds that can be obtained from this approach are too weak to establish asymptotic normality [Lat13].

Like [DDS13b], in this paper we also use the central limit theorem from [NPR10] as a starting point. However, our subsequent analysis crucially relies on the fact that there is a geometry-preserving isomorphism between the space of symmetric tensors and multivariate Gaussian polynomials. This allows us to view Gaussian polynomials in terms of the associated tensors and greatly facilitates the use of language and tools from tensor algebra. To establish our condition for asymptotic normality, we make significant use of tensor identities from Malliavin calculus which were developed in the context of application to Stein’s method (see [NP09, Nou12, NPR10]).

Second contribution: Decomposition. The second main contribution of this paper is a novel decomposition that lets us transform a multilinear degree- d Gaussian polynomial p into a polynomial of the form $h(A_1, \dots, A_r)$, where (informally)

1. p and $h(A_1, \dots, A_r)$ are ϵ -close (meaning that $\mathbf{E}[p] = \mathbf{E}[h(A_1, \dots, A_r)]$ and $\mathbf{Var}[p - h(A_1, \dots, A_r)] \leq \epsilon$);
2. For each polynomial A_i , all of its eigenvalues are extremely small (at most η for some very small η); and
3. $r = r(\epsilon, d, \eta)$ is independent of n and depends only on the approximation parameter ϵ , the eigenvalue bound η , and the degree d of p .

This decomposition is useful for the following reasons: Property (1) ensures that the distributions of $h(A_1, \dots, A_r)$ and p are close in cdf-distance, and thus in order to do approximate counting of Gaussian satisfying assignments for p , it suffices to do approximate counting of Gaussian satisfying assignments for $h(A_1, \dots, A_r)$. Property (2) ensures that we may apply our new CLT to the r -tuple of polynomials A_1, \dots, A_r , and thus we may approximately count satisfying assignments to $h(A_1, \dots, A_r) \geq 0$ by approximating the fraction of assignments that satisfy $h(\mathcal{G}_1, \dots, \mathcal{G}_r)$ where $\mathcal{G} = (\mathcal{G}_1, \dots, \mathcal{G}_r)$ is the multidimensional Normal distribution given by our CLT. Finally, by Property (3), approximating $\mathbf{Pr}[h(\mathcal{G}_1, \dots, \mathcal{G}_r) \geq 0]$ is a “constant-dimensional problem” (independent of n) so it is straightforward for a deterministic algorithm to approximate this probability in time independent of n .

We note that there is a subtlety here which requires significant effort to overcome. As we discuss in Remark 8, in order for our CLT to give a nontrivial bound it must be the case that the eigenvalue bound η is much smaller than

$1/r$. Mimicking decomposition approaches previously used in literature [Ser07, MZ09, DSTW10] has the problem that they will necessarily make $r \geq 1/\eta$, thus rendering such decompositions useless for our purposes. (One exception is the decomposition procedure from [Kan11a] where a similar problem arises, but since the desired target conditions there are different from ours, that work uses a different approach to overcome the difficulty; we elaborate on this below.) In our context, achieving a decomposition such that $\eta \ll 1/r$ requires ideas that go beyond those used in previous decompositions, and is responsible for the large “constant-factor” overhead (captured by $O_{d,\epsilon}(1)$) in the overall running time bound.

At a very high level our decomposition is reminiscent of the regularity lemma for degree- d polynomials over $\{-1, 1\}^n$ that was given in [DSTW10], in that both procedures break a given degree- d input polynomial into a collection of “regular” polynomials, but as we now explain, this resemblance is a superficial one as there are many significant differences. First, in the [DSTW10] setting the given input polynomials are over $\{-1, 1\}^n$ while here the polynomials are over Gaussian space; this is a major distinction since the geometry of Gaussian space plays a fundamental role in our proofs and techniques. Second, the notion of “regularity” that is used is quite different between the two works; in [DSTW10] a polynomial is regular if all variable influences are small whereas here a polynomial is “regular” if all its “tensor eigenvalues” are small. (We subsequently refer to this new notion of regularity which is introduced and used in our work as *eigenregularity*.) Third, in [DSTW10] each “atomic step” of the decomposition is simply to restrict an individual input variable to $+1$ or -1 , whereas in this paper the atomic “decomposition step” now involves an eigenvalue computation (to identify two lower-degree polynomials whose product is non-trivially correlated with the polynomial being decomposed). Finally, the [DSTW10] decomposition produces a decision tree over input variables with restricted polynomials at the leaves, whereas in this paper we produce a *single degree- d polynomial* $h(A_1, \dots, A_r)$ as the output of our decomposition.

Our decomposition has some elements that are reminiscent of a decomposition procedure described in [Kan11a]. Kane’s procedure, like ours, breaks a degree- d polynomial into a sum of product of lower degree polynomials. However, there are significant differences between the procedures. Roughly speaking, Kane’s decomposition starts with a polynomial p and is aimed at upper bounding the higher moments of the resulting constituent polynomials, whereas our decomposition is aimed at upper bounding the eigenregularity (magnitude of the largest eigenvalues) of the constituent polynomials. To make sure that the number r of constituent polynomials compares favorably with the moment bounds, Kane divides these polynomials into several classes such that the number of polynomials in any class compares favorably with the moment bounds in that class (and some desired relation holds between the number of polynomials in the different classes). Instead, in our decomposition procedure, we want r to compare favorably with the eigenvalue bound η ; given this requirement, it does not seem possible to mimic Kane’s approach of splitting the constituent polynomials into several classes. Instead, through a rather elaborate decomposition procedure, we show that while it may not be possible to split the original polynomial

p in a way so that r compares favorably with η , it is always possible to (efficiently) find a polynomial \tilde{p} such that $p - \tilde{p}$ has small variance, and \tilde{p} can be decomposed so that the number of constituent polynomials compare favorably with the eigenregularity parameter.

We note that it is possible for the polynomial $p - \tilde{p}$ to have small variance but relatively huge moments. Thus our decomposition procedure is not effective for the approach in [Kan11a] which is based on bounding moments. However, because $p - \tilde{p}$ has small variance, the distributions of p and \tilde{p} are indeed close in cdf distance, which suffices for our purposes. Thus our decomposition procedure should be viewed as incomparable to that of [Kan11a].

We also remark that our decomposition is significantly more involved than the decompositions used in [DDS13a, DDS13b]. To see how this additional complexity arises, note that both these papers need to decompose either a single degree-2 Gaussian polynomial or a set of such polynomials; for simplicity assume we are dealing with a single degree-2 polynomial p . Then the [DDS13a] decomposition procedure splits p into a sum of products of linear functions plus a degree-2 polynomial which has small eigenvalues. Crucially, since a linear function of Gaussians is itself a Gaussian, this permits a change of basis in which these linear functions may be viewed as the new variables. By “restricting” these new variables, one is essentially left with a single degree-2 polynomial with a small eigenvalue. In contrast, if p has degree d greater than 2, then the [DDS13a] decomposition will split p into a sum of products of pairs of lower degree Gaussian polynomials plus a polynomial which has small eigenvalues. However, if $d > 2$ then some or all of the new constituent lower degree polynomials may have degree greater than 1. Since a polynomial of degree $d > 1$ cannot itself be viewed as a Gaussian, this precludes the possibility of “restricting” this polynomial as was done in [DDS13a]. Thus, one has to resort to an iterative decomposition, which introduces additional complications some of which were discussed above.

Organization. Because of space constraints proofs are omitted in this extended abstract (see the full version for all proofs). In Section 2 we show that it is sufficient to give an algorithm for deterministic approximate counting of degree- d polynomials in the special case where all the polynomials are multilinear. In Section 3 we state our new CLT for k -tuples of degree- d Gaussian polynomials with “small eigenvalues.” In Section 4 we describe our decomposition procedure that can be used to decompose a degree- d multilinear polynomial over Gaussian inputs into an essentially equivalent polynomial that has a highly structured “special form.” In Section 5 we show how the CLT from Section 3 can be combined with the highly structured polynomial from Section 4 to prove Theorem 2. We close in Section 6 by briefly describing how Theorem 1 can be applied to give the first deterministic fixed-parameter tractable algorithm for the problem of approximating the k -th absolute moment of a degree- d polynomial over $\{-1, 1\}^n$.

2. DEALING WITH NON-MULTILINEAR POLYNOMIALS

The decomposition procedure that we use relies heavily on the fact that the input polynomials p_i are multilinear. To handle general (non-multilinear) degree- d polynomials, the first step of our algorithm is to transform them to (essen-

tially) equivalent multilinear degree- d polynomials. This is accomplished by a simple procedure whose performance is described below. Note that given Theorem 3, in subsequent sections we can (and do) assume that the polynomial p given as input in Theorem 2 is multilinear.

THEOREM 3. *There is a deterministic procedure with the following properties: The algorithm takes as input a (not necessarily multilinear) variance-1 degree- d polynomial p over \mathbb{R}^n and an accuracy parameter $\delta > 0$. It runs in time $O_{d,\delta}(1) \cdot \text{poly}(n^d)$ and outputs a multilinear degree- d polynomial q over $\mathbb{R}^{n'}$, with $n' \leq O_{d,\delta}(1) \cdot n$, such that*

$$\left| \Pr_{x \sim N(0,1)^n} [p(x) \geq 0] - \Pr_{x \sim N(0,1)^{n'}} [q(x) \geq 0] \right| \leq O(\delta).$$

3. A MULTIDIMENSIONAL CLT FOR LOW-DEGREE GAUSSIAN POLYNOMIALS

Our goal in this section is to prove a CLT (Theorem 7 below) which says, roughly, the following: Let F_1, \dots, F_r be eigenregular low-degree Gaussian polynomials over \mathbb{R}^n (here the meaning of “eigenregular” is that the polynomial has “small eigenvalues”; more on this below). Then the distribution of (F_1, \dots, F_r) is close — as measured by test functions with absolutely bounded second derivatives — to the r -dimensional Normal distribution with matching mean and covariance.

To make this statement more precise, let us begin by explaining what exactly is meant by the eigenvalues of a polynomial — this is clear enough for a quadratic polynomial, but not so clear for degrees 3 and higher.

Eigenvalues of tensors and polynomials. Let \mathcal{H} denote the Hilbert space \mathbb{R}^n , and let $\mathcal{H}^{\odot p}$ denote the space of symmetric p -tensors over \mathcal{H} . (See the full version for detailed background on tensors.) We begin by defining the largest eigenvalue of a symmetric tensor.

DEFINITION 4. *For any $p \geq 2$ and $g \in \mathcal{H}^{\odot p}$, define $\lambda_{\max}(g)$, the largest-magnitude eigenvalue of g , as follows. Consider a partition of $[p]$ into S and $\bar{S} = [p] \setminus S$ where both S and \bar{S} are non-empty. We define $\lambda_{S, \bar{S}}(g) = \sup_{x \in \mathcal{H}^S, y \in \mathcal{H}^{\bar{S}}} \frac{(g, x \otimes y)}{\|x\|_F \cdot \|y\|_F}$ and $\lambda_{\max}(g) = \max_{S, \bar{S} \neq \emptyset} \lambda_{S, \bar{S}}(g)$. (Here $\|x\|_F$ denotes the Frobenius norm of x .) For $p \in \{0, 1\}$ and $g \in \mathcal{H}^{\odot p}$ we say that $\lambda_{\max}(g) = 0$.*

Let \mathcal{W}^q (referred to as the q -th Wiener chaos) denote the linear subspace of polynomials spanned by the Hermite polynomials of degree exactly q over \mathbb{R}^n . Fix a Gaussian polynomial F of degree d and recall that F admits a unique Wiener chaos decomposition $F = \sum_{q=0}^d I_q(f_q)$; here $f_q \in \mathcal{H}^{\odot q}$ and is the natural tensor associated with the projection of F onto \mathcal{W}^q . Thus $I_q(\cdot)$ maps the tensor $f_q \in \mathcal{H}^{\odot q}$ to a polynomial in \mathcal{W}^q . (While the precise definition of I_q is not required for the rest of this extended abstract, the curious reader is encouraged to consult Section 2 of the full version for additional details.)

The following definition plays a crucial role in the rest of the paper.

DEFINITION 5. *We define the largest-magnitude eigenvalue of F to be $\lambda_{\max}(F) = \max\{\lambda_{\max}(f_2), \dots, \lambda_{\max}(f_d)\}$. We say that F is ϵ -eigenregular if $\frac{\lambda_{\max}(F)}{\sqrt{\text{Var}[F]}} \leq \epsilon$, and we sometimes refer to $\frac{\lambda_{\max}(F)}{\sqrt{\text{Var}[F]}}$ as the eigenregularity of F .*

REMARK 6. If F is a polynomial of degree at most 1 then we say that the polynomial F is 0-eigenregular (and hence ϵ -eigenregular for every $\epsilon > 0$).

Now we can give a precise statement of our new CLT:

THEOREM 7. Fix $d \geq 2$ and let $F = (F_1, \dots, F_r)$ be Gaussian polynomials over \mathbb{R}^n , each of degree at most d , such that for each i we have $\mathbf{E}[F_i] = 0$, $\mathbf{Var}[F_i] \leq 1$ and F_i is ϵ -eigenregular. Let C denote the covariance matrix of F , so $C(i, j) = \text{Cov}(F_i, F_j) = \mathbf{E}_{x \sim N(0,1)^n}[F_i(x)F_j(x)]$. Let \mathcal{G} be a mean-zero r -dimensional Gaussian random variable with covariance matrix C . Then for any $\alpha : \mathbb{R}^r \rightarrow \mathbb{R}$, $\alpha \in \mathcal{C}^2$ such that all second derivatives of α are at most $\|\alpha''\|_\infty < \infty$, we have $|\mathbf{E}[\alpha(F_1, \dots, F_r)] - \mathbf{E}[\alpha(\mathcal{G})]| < 2^{O(d \log d)} \cdot r^2 \cdot \sqrt{\epsilon} \cdot \|\alpha''\|_\infty$.

The proof of Theorem 7 is somewhat involved, using Malliavin calculus in the context of Stein’s method; it builds on recent work by Nourdin, Peccati and Réveillac [NP09, Nou12, NPR10]. We give the proof in the full version.

REMARK 8. It is clear from the statement of Theorem 7 that in order for the theorem to yield a meaningful bound, it must be the case that the number of polynomials r is small compared to $1/\sqrt{\epsilon}$. Looking ahead, in our eventual application of Theorem 7, the r polynomials F_1, \dots, F_r will be obtained by applying the decomposition procedure described in Section 4 to the original degree- d input polynomial. Thus it will be crucially important for our decomposition procedure to decompose the original polynomial into r polynomials all of which are extremely eigenregular, in particular ϵ -eigenregular for a value $\epsilon \ll 1/r^2$. Significant work will be required in Section 4 to surmount this challenge.

4. DECOMPOSING K -TUPLES OF MULTILINEAR DEGREE- D GAUSSIAN POLYNOMIALS

In this section we prove our main decomposition result for k -tuples of multilinear Gaussian polynomials, Theorem 9. We begin by giving a precise statement of the result, followed by a discussion of how the result fits into our broader context.

THEOREM 9. Fix $d \geq 2$ and fix any non-increasing computable function $\beta : [1, \infty) \rightarrow (0, 1)$ that satisfies $\beta(x) \leq 1/x$. There is a procedure **Regularize-Poly** $_\beta$ with the following properties. The procedure takes as input a degree- d multilinear Gaussian polynomial p with $\mathbf{Var}[p] = 1$ and a parameter $\tau > 0$. It runs in $\text{poly}(n^d) \cdot O_{d,\tau}(1)$ time and outputs two collections of polynomials

$$\{h_q\}_{q=0,\dots,d} \text{ and } \{A_{q,\ell}\}_{q=0,\dots,d,\ell=1,\dots,m_q}.$$

Write $p(x)$ as $\sum_{q=0}^d c_q p_q(x)$ where $p_q \in \mathcal{W}^q$ for all q and $\mathbf{Var}[p_q] = 1$ for $1 \leq q \leq d$. For $0 \leq q \leq d$ and $x \in \mathbb{R}^n$, let $\tilde{p}_q(x) = c_q h_q(A_{q,1}(x), \dots, A_{q,m_q}(x))$ and let $\tilde{p}(x) = \sum_{q=0}^d \tilde{p}_q(x)$. The following conditions hold:

1. For each $q \in \{0, \dots, d\}$ the polynomial \tilde{p}_q belongs to \mathcal{W}^q . Moreover, for $q \in \{1, \dots, d\}$, each polynomial $A_{q,\ell}$ belongs to \mathcal{W}^j for some $1 \leq j \leq d$ and has $\mathbf{Var}[A_{q,\ell}] = 1$.

2. We have

$$|\Pr_{x \sim N(0,1)^n}[p(x) \geq 0] - \Pr_{x \sim N(0,1)^n}[\tilde{p}(x) \geq 0]| \leq O(\tau),$$

and moreover $\mathbf{Var}_{x \sim N(0,1)^n}[p(x) - \tilde{p}(x)] \leq (\tau/d)^{3d}$.

3. Each polynomial h_q is a multilinear polynomial in its m_q arguments. Moreover, there exist functions $N_\beta(d, \tau)$ and $M_\beta(d, \tau)$ such that if $\text{Coeff}(h_q)$ denotes the sum of the absolute values of the coefficients of h_q , then $\sum_{q=1}^d \text{Coeff}(h_q) \leq M_\beta(d, \tau)$ and the number of arguments to all h_q ’s, $\sum_{q=1}^d m_q$, is at most $N_\beta(d, \tau)$. Also, the degree of the polynomial h_q (for all $1 \leq q \leq d$) is upper bounded by d .
4. For all q, ℓ , $A_{q,\ell}(x)$ is $\beta(\text{Num} + \text{Coeff})$ -eigenregular, where $\text{Num} = \sum_{q=1}^d m_q$ and $\text{Coeff} = \sum_{q=1}^d \text{Coeff}(h_q)$.

Discussion. Intuitively, Condition (2) means that it will be sufficient to do deterministic approximate counting for the polynomial \tilde{p} rather than the original polynomial p . Condition (4) ensures that the eigenregularity of each polynomial $A_{q,\ell}$ compares favorably both with the number of polynomials produced and with the size of the coefficients in the “outer” polynomials h_q . As discussed earlier, having the eigenregularity be small relative to the number of polynomials is crucial since it is required in order for our CLT, Theorem 7, to yield a good bound. We need to deal with the size of the coefficients for technical reasons – as we will see in Section 5, we will apply our CLT where its “test function” α is a smooth approximator to the 0/1-valued function which, on input $\{A_{j,\ell}\}_{j,\ell}$, outputs 1 if and only if $\text{sign}\left(\sum_{q=0}^d h_q(A_{q,1}, \dots, A_{q,m_q})\right) = 1$. Our CLT’s quantitative bound depends on the second derivative of α , and to bound this we need coefficient size bounds on the h_q polynomials.

We build up to Theorem 9 in a sequence of incremental stages. In Section 4.1 we begin by describing how to decompose a single element of a given Wiener chaos. Because of our requirement that the number of polynomials produced in the decomposition must be very small relative to the eigenregularity that is achieved — see Remark 8 — even this is a non-trivial task, requiring two “layers” of decomposition and an approach that goes well beyond the decomposition techniques in previous work [DDS13a, DDS13b]. In Section 4.2 we extend this and describe how to simultaneously decompose a k -tuple of elements of the same Wiener chaos. (See the beginning of Section 4.2 for an explanation of why we need to be able to simultaneously decompose many polynomials at once.) In Section 4.3 we describe how to handle a $k(d+1)$ -tuple of elements where there are k elements from each of the $d+1$ Wiener chaoses $\mathcal{W}^0, \dots, \mathcal{W}^d$. In the full version, we show how the decomposition of Section 4.3 yields Theorem 9.

4.1 Decomposing a single multilinear element of the q -th Wiener chaos

Our first algorithmic primitive is the procedure **Split-One-Wiener**. (Here and subsequently the suffix “-One-Wiener” indicates that the procedure applies only to one element $I_q(f)$ belonging to one level of the Wiener chaos.) This procedure either certifies that its input polynomial (an

element $I_q(f)$ of the q -th Wiener chaos) is eigenregular, or else it “splits off” a product $P \cdot Q$ from its input polynomial and expresses $I_q(f)$ as $c \cdot P \cdot Q + R$ for some $c \in \mathbb{R}$.

LEMMA 10. *Fix any $q \geq 2$. There is a deterministic procedure **Split-One-Wiener** which takes as input a polynomial $I_q(f) \in \mathcal{H}_q$ that has $\mathbf{Var}[I_q(f)] = 1$ and a parameter $\eta > 0$. **Split-One-Wiener** runs in deterministic $\text{poly}(n^q, 1/\eta)$ time and has the following guarantee:*

- If $\lambda_{\max}(f) < \eta$, then **Split-One-Wiener** stops and returns “eigenregular”.
- Otherwise, if $\lambda_{\max}(f) \geq \eta$, then **Split-One-Wiener** outputs a quadruple (P, Q, R, c) with the following properties:
 1. $P = I_{q_1}(g_1) \in \mathcal{W}^{q_1}$ and $Q = I_{q_2}(g_2) \in \mathcal{W}^{q_2}$ where $q_1 + q_2 = q$, $q_1, q_2 > 0$, and $\mathbf{Var}[I_{q_1}(g_1)] = \mathbf{Var}[I_{q_2}(g_2)] = 1$.
 2. The tensors $g_1 \in \mathcal{H}^{\odot q_1}$ and $g_2 \in \mathcal{H}^{\odot q_2}$ are supported on disjoint sets $S, T \subset [n]$.
 3. $P \cdot Q \in \mathcal{W}^q$ and $\mathbf{Var}[P \cdot Q] = 1$, and all of P, Q, R are multilinear.
 4. The value $c \stackrel{\text{def}}{=} \mathbf{E}[I_q(f) \cdot P \cdot Q]$ satisfies $c \geq \eta/2^q$.
 5. $R \in \mathcal{W}^q$ and $I_q(f) = cP \cdot Q + R$ and $\mathbf{E}[P \cdot Q \cdot R] = 0$.
 6. $\mathbf{Var}(R) = 1 - c^2$.

Building on the algorithmic primitive **Split-One-Wiener**, we now describe a procedure **Decompose-One-Wiener** which works by iteratively executing **Split-One-Wiener** on the “remainder” portion R that was “left over” from the previous call to **Split-One-Wiener**. Intuitively, the overall effect of this procedure is to break its input polynomial into a sum of products of pairs of polynomials, plus a remainder term which is either eigenregular or else has variance which is negligibly small.

LEMMA 11. *Fix any $q \geq 2$. There is a deterministic procedure **Decompose-One-Wiener** which takes as input a polynomial $I_q(f) \in \mathcal{W}_q$ that has $\mathbf{Var}[I_q(f)] = 1$ and parameters η and ϵ . The running time of **Decompose-One-Wiener** is $\text{poly}(n^q, 1/\eta, \log(1/\epsilon))$ and has the following guarantee:*

1. It outputs a set L of triples $\{(c_i, P_i, Q_i)\}_{i=1}^m$ and a polynomial R such that $I_q(f) = \sum_{i=1}^m c_i P_i Q_i + R$.
2. For each $i = 1, \dots, m$ we have $P_i \in \mathcal{W}^{q_{i,1}}$ and $Q_i \in \mathcal{W}^{q_{i,2}}$ with $q_{i,1}, q_{i,2} > 0$ and $q_{i,1} + q_{i,2} = q$. Moreover $\mathbf{Var}[P_i] = \mathbf{Var}[Q_i] = \mathbf{Var}[P_i \cdot Q_i] = 1$ for all $i \in [m]$, $R \in \mathcal{W}^q$, all P_i, Q_i and R are multilinear, and P_i and Q_i are defined over disjoint sets of variables.
3. $m \leq O((4^q/\eta^2) \log(1/\epsilon))$ and $\sum_{j=1}^m c_j^2 \leq (2^q/\eta)^{4(m-1)}$.
4. Either R is η -eigenregular, in which case **Decompose-One-Wiener** returns “eigenregular remainder”, or else $\mathbf{Var}[R] \leq \epsilon$, in which case **Decompose-One-Wiener** returns “small remainder”.
5. $\mathbf{E}[(\sum_{j=1}^m c_j P_j \cdot Q_j) \cdot R] = 0$. As a consequence, we have $\mathbf{Var}[\sum_{j=1}^m c_j P_j \cdot Q_j] + \mathbf{Var}[R] = 1$.

We note that the guarantees of the **Decompose-One-Wiener** procedure bear some resemblance to the decomposition that is used in [DDS13a] for degree-2 Gaussian polynomials. However, in our current context of working with degree- d polynomials, **Decompose-One-Wiener** is not good enough, for the following reason: Suppose that **Decompose-One-Wiener** returns “eigenregular remainder” and outputs a decomposition of $I_q(f)$ as $\sum_{i=1}^m c_i P_i Q_i + R$. While the polynomial R is η -eigenregular, it is entirely possible that the number of polynomials P_i, Q_i in the decomposition (i.e. $2m$) may be as large as $\Omega(\frac{1}{\eta^2} \log(1/\epsilon))$. We would like to apply our CLT to conclude that the joint distribution of R and the polynomials obtained from the subsequent decomposition of $P_1, Q_1, \dots, P_m, Q_m$ is close to a normal distribution, but since the number $2m$ of polynomials is already too large when compared to the inverse of the eigenregularity parameter, we cannot use our CLT (recall Remark 8).¹

We surmount this difficulty by using **Decompose-One-Wiener** as a tool within an improved “two-level” decomposition procedure which we present and analyze below. This improved decomposition procedure has a stronger guarantee than **Decompose-One-Wiener** in the following sense: it breaks its input polynomial into a sum of products of pairs of polynomials plus *two* remainder terms R_{reg} (for “eigenregular”) and R_{neg} (for “negligible”). The R_{neg} remainder term is guaranteed to have negligibly small variance, and the R_{reg} remainder term is guaranteed to either be zero or else to be *extremely* eigenregular – in particular, for an appropriate setting of the input parameters, its eigenregularity is much “stronger” than the number of pairs of polynomials that are produced in the decomposition. We term this improved decomposition procedure **Regularize-One-Wiener** because of this extremely strong eigenregularity guarantee.

Before giving the formal statement, we note that intuitively this procedure will be useful because it “guarantees that we make progress” for the following reason: We can always erase the small-variance R_{neg} term at the cost of a small and affordable error, and the degree- q R_{reg} remainder term is so eigenregular that it will not pose an obstacle to our ultimate goal of applying the CLT. Thus we have reduced the original polynomial to a sum of pairwise products of lower-degree polynomials, which can each be tackled inductively using similar methods (more precisely, using the generalization of procedure **Regularize-One-Wiener** to simultaneously decompose multiple polynomials which we describe in the next subsection).

THEOREM 12. *Fix any $q \geq 2$. There is a deterministic procedure **Regularize-One-Wiener** which takes as input a polynomial $I_q(f)$ such that $\mathbf{Var}[I_q(f)] = 1$ and input parameters $\eta_0 = 1 \geq \eta_1 \geq \dots \geq \eta_K$ and ϵ , where $K = O(1/\epsilon \cdot \log(1/\epsilon))$. **Regularize-One-Wiener** runs in $\text{poly}(n^q, 1/\eta_K, 1/\epsilon)$ time and has the following guarantee:*

¹Note that the reason this problem did not arise in the degree-2 polynomial decompositions of [DDS13a] is because each polynomial P_i, Q_i obtained from **Decompose-One-Wiener** in that setting must have degree 1 (the only way to break the number 2 into a sum of non-negative integers is as 1+1). Degree-1 polynomials may be viewed as having “perfect eigenregularity” (note that any degree-1 polynomial in Gaussian variables is itself distributed precisely as a Gaussian) and so having any number of such degree-1 polynomials did not pose a problem in [DDS13a].

1. Define $M(i) = \frac{O(4^q)}{\eta_i^2} \log(1/\epsilon)$. The **Regularize-One-Wiener** procedure outputs a value $1 \leq \ell \leq k$, a set

$$L = \{(a_{i,j}, P_{i,j}, Q_{i,j})\}_{i=1, \dots, \ell, j=1, \dots, M(i)}$$

of triples, and a pair of polynomials $R_{\text{reg}}, R_{\text{neg}}$ such that $I_q(f) = \sum_{i=1}^{\ell} \sum_{j=1}^{M(i)} a_{i,j} P_{i,j} \cdot Q_{i,j} + R_{\text{reg}} + R_{\text{neg}}$.

2. For each i, j we have $P_{i,j} \in \mathcal{W}^{q_{i,j,1}}$ and $Q_{i,j} \in \mathcal{W}^{q_{i,j,2}}$ with $q_{i,j,1}, q_{i,j,2} > 0$ and $q_{i,j,1} + q_{i,j,2} = q$ and $\mathbf{Var}[P_{i,j}] = \mathbf{Var}[Q_{i,j}] = \mathbf{Var}[P_{i,j} \cdot Q_{i,j}] = 1$; moreover, $P_{i,j}$ and $Q_{i,j}$ are over disjoint sets of variables. In addition, $R_{\text{reg}}, R_{\text{neg}} \in \mathcal{W}^q$ and all of $P_{i,j}, Q_{i,j}, R_{\text{reg}}, R_{\text{neg}}$ are multilinear.
3. The polynomial R_{neg} satisfies $\mathbf{Var}[R_{\text{neg}}] \leq \epsilon$ and the polynomial R_{reg} is $\eta_{\ell+1}$ -eigenregular, where we define $\eta_{K+1} = 0$.
4. For $1 \leq i \leq \ell$ we have $\sum_{j=1}^{M(i)} (a_{i,j})^2 \leq (2^q/\eta_i)^{4(M(i)-1)}$.

We stress that it is crucially important that condition 3 provides $\eta_{\ell+1}$ -eigenregularity rather than η_{ℓ} -eigenregularity.

4.2 Decomposing a k -tuple of multilinear elements of the q -th Wiener chaos

In this section we generalize the **Regularize-One-Wiener** procedure to simultaneously decompose multiple polynomials that all belong to \mathcal{W}^q . Even though our ultimate goal is to decompose a single degree- d Gaussian polynomial, we require a procedure that is capable of handling many polynomials because even decomposing a single degree- d polynomial using **Regularize-One-Wiener** will give rise to many lower-degree polynomials which all need to be decomposed in turn.

A natural approach to decompose r polynomials $I_q(f_1), \dots, I_q(f_r) \in \mathcal{W}^q$ is simply to run **Regularize-One-Wiener** r separate times. However, this simpleminded approach could well result in different values ℓ_1, \dots, ℓ_r being obtained from the r calls, and hence in different levels of eigenregularity for the r “remainder” polynomials $R_{1,\text{reg}}, \dots, R_{r,\text{reg}}$ that are constructed. This is a problem because some of the calls may yield a relatively large eigenregularity parameter, while other calls may generate very many polynomials (and a much smaller eigenregularity parameter). Since the CLT can only take advantage of the largest eigenregularity parameter, the key advantage of **Regularize-One-Wiener** — that the number of polynomials it produces compares favorably with the eigenregularity of these polynomials — is lost.

We get around this difficulty by means of a procedure called **MultiRegularize-One-Wiener**. It takes as input an r -tuple of polynomials $(I_q(f_1), \dots, I_q(f_r))$ (that all belong to one fixed Wiener chaos) and input parameters $\eta_0 = 1 \geq \eta_1 \geq \dots \geq \eta_K$ and ϵ . Crucially, it guarantees that the *overall* number of polynomials that are produced from all the r decompositions compares favorably with the overall eigenregularity parameter that is obtained. Intuitively, **MultiRegularize-One-Wiener** augments the procedure **Regularize-One-Wiener** procedure with ideas from the decomposition procedure for k -tuples of degree-2 polynomials that was given in [DDS13b] (and which in turn built on ideas from [GOWZ10]).

THEOREM 13. *There is a procedure **MultiRegularize-One-Wiener** (for any $q \geq 2$) which takes as input an r -tuple of polynomials $(I_q(f_1), \dots, I_q(f_r))$ such that for all i , $\mathbf{Var}[I_q(f_i)] = 1$ and input parameters $\eta_0 = 1 \geq \eta_1 \geq \dots \geq \eta_K$ and ϵ , where $K = O(r/\epsilon \cdot \log(1/\epsilon))$. **MultiRegularize-One-Wiener** runs in $\text{poly}(n^q, 1/\eta_K, r/\epsilon)$ time and has the following guarantee:*

1. Define $M(i) = O(\frac{4^q}{\eta_i^2} \log(1/\epsilon))$. **MultiRegularize-One-Wiener** outputs an index t with $0 \leq t \leq K$ and for each $s \in [r]$ a set L_s of triples, defined as, $L_s = \{(a_{s,i,j}, P_{s,i,j}, Q_{s,i,j})\}_{i=1, \dots, t, j=1, \dots, M(i)}$ and a pair of polynomials $R_{s,\text{reg}}, R_{s,\text{neg}}$, such that

$$I_q(f_s) = \sum_{i=1}^t \sum_{j=1}^{M(i)} a_{s,i,j} P_{s,i,j} \cdot Q_{s,i,j} + a_{s,\text{reg}} R_{s,\text{reg}} + R_{s,\text{neg}}$$

2. For each s, i, j we have $P_{s,i,j} \in \mathcal{W}^{q_{s,i,j,1}}$ and $Q_{s,i,j} \in \mathcal{W}^{q_{s,i,j,2}}$ with $q_{s,i,j,1}, q_{s,i,j,2} > 0$ and $q_{s,i,j,1} + q_{s,i,j,2} = q$ and $\mathbf{Var}[P_{s,i,j}] = \mathbf{Var}[Q_{s,i,j}] = \mathbf{Var}[P_{s,i,j} \cdot Q_{s,i,j}] = 1$. Similarly we have $R_{s,\text{reg}}, R_{s,\text{neg}} \in \mathcal{W}^q$, and $\mathbf{Var}[R_{s,\text{reg}}] = 1$. Moreover $P_{s,i,j}$ and $Q_{s,i,j}$ are over disjoint sets of variables, and all of $P_{s,i,j}, Q_{s,i,j}, R_{s,\text{reg}}$ and $R_{s,\text{neg}}$ are multilinear.
3. For each s we have that $\mathbf{Var}[R_{s,\text{neg}}] \leq \epsilon$ and that $a_{s,\text{reg}} R_{s,\text{reg}}$ is η_{t+1} -eigenregular, where we define $\eta_{K+1} = 0$.
4. For $1 \leq s \leq r$ and $1 \leq i \leq t$ we have $\sum_{j=1}^{M(i)} (a_{s,i,j})^2 \leq (2^q/\eta_i)^{4(M(i)-1)}$.

4.3 Beyond the homogeneous case: handling multiple levels of Wiener chaos

In this subsection we sketch our most involved decomposition procedure, **MultiRegularize-Many-Wieners**, for decomposing a $k(d+1)$ -tuple consisting of k elements from the j -th Wiener chaos for each $j = 0, \dots, d$. We begin with an informal description of how the decomposition procedure works. Let p_1, \dots, p_k be k degree- d multilinear Gaussian polynomials. Each p_i has a unique expansion in terms of symmetric q -tensors $f_{i,q} \in \mathcal{H}^{\odot q}$ as $p_i = \sum_{q=0}^d p_{i,q}$, where $p_{i,q} = I_q(f_{i,q})$. For $2 \leq q \leq d-1$ let OLD_q denote the set of polynomials $\{I_q(f_{i,q})\}_{i=1, \dots, k}$.

The high-level idea of the decomposition is to “work downward” from higher to lower levels of the Wiener chaos in successive stages, at each stage using **MultiRegularize-One-Wiener** to simultaneously decompose all of the polynomials at the current level. By carefully choosing the eigenregularity parameters at each stage we can ensure that at the end of the decomposition we are left with a collection of “not too many” polynomials (corresponding to the $A_{i,j,\ell}$ ’s of Theorem 9) all of which are highly eigenregular.

In a bit more detail, in the first stage we simultaneously decompose the k degree- d polynomials $I_d(f_{1,d}), \dots, I_d(f_{k,d})$ using the **MultiRegularize-One-Wiener** algorithm with parameters $1 = \eta_0 \gg \dots \gg \eta_K$ and ϵ . This generates k polynomials in \mathcal{W}^d that are each η_{t+1} -eigenregular, for some $1 \leq t \leq K$, where $K \leq O_{k,d,\epsilon}(1)$; intuitively, these should be thought of as “extremely eigenregular” polynomials. Let REG denote this set of polynomials (they will not be used again in the decomposition). It also generates, for each $1 \leq q \leq d-1$, “not too many” (at most $O_{k,d,\epsilon,\eta_t}(1)$)

new polynomials in \mathcal{W}^q ; let NEW_q denote this set of polynomials. The key qualitative point is that the size of each NEW_q depends on η_t while the eigenregularity of the polynomials in REG is η_{t+1} . Thus if η_{t+1} is much less than η_t , the number of newly introduced polynomials compares favorably with the eigenregularity bound η_{t+1} .

We have thus “dealt with the degree- d part of the input” since the only remaining degree- d polynomials are in REG and are extremely regular. Next, we recursively apply the above approach to handle the lower-degree part, including both the original lower-degree components from OLD_q and the new lower-degree polynomials from NEW_q that were introduced in dealing with the degree- d part. The crux of the analysis is to argue that there is a suitable choice of parameters at each stage that allows this procedure to be carried out “all the way down,” so that the total number of polynomials that are ever produced in the analysis is far smaller than $1/\eta$, where η is the largest eigenregularity of any of the final polynomials. It turns out to be difficult to argue this formally using the “top-down” view on the decomposition procedure that we have adopted above. Instead, in the detailed proof which we give in the full version, we take a “bottom-up” view of the decomposition procedure: we first show that it can be successfully carried out for low-degree polynomials, and use this fact to show that it can be successfully carried out for higher-degree polynomials.

THEOREM 14. *Fix $d \geq 2$ and fix any non-increasing computable function $\beta : [1, \infty) \rightarrow (0, 1)$ that satisfies $\beta(x) \leq 1/x$. The procedure **MultiRegularize-Many-Wieners** $_{d,\beta}$ defined below has the following properties. The procedure takes as input the following:*

- It is given k lists of $d + 1$ multilinear Gaussian polynomials; the s -th list is $p_{s,0}, \dots, p_{s,d}$ where $p_{s,q} \in \mathcal{W}^q$ and $\mathbf{Var}[p_{s,q}] = 1$ for $1 \leq q \leq d$.
- It also takes as input a parameter $\tau > 0$.

The procedure runs in $\text{poly}(n^d) \cdot O_{k,d,\tau}(1)$ time and outputs, for each input polynomial $p_{s,q}$, a polynomial $\text{Out}(p_{s,q})$ and a collection of polynomials **Inner** defined to be $\text{Inner} = \{\text{In}(p_{s,q})_\ell\}_{\ell=1, \dots, \text{num}(p_{s,q})}$; here $\text{num}(p_{s,q})$ is the number of arguments of the polynomial $\text{Out}(p_{s,q})$. (“Out” stands for “outer” and “In” stands for “inner”.)

For $s = 1, \dots, k$, $0 \leq q \leq d$ and $x \in \mathbb{R}^n$, let

$$\tilde{p}_{s,q}(x) = \text{Out}(p_{s,q}) \left(\text{In}(p_{s,q})_1(x), \dots, \text{In}(p_{s,q})_{\text{num}(p_{s,q})}(x) \right) \quad (1)$$

(Intuitively, each $\tilde{p}_{s,q}$ is a polynomial that has been decomposed into constituent sub-polynomials from **Inner**; $\tilde{p}_{s,q}$ is meant to be a good approximator for $p_{s,q}$. The following conditions make this precise.)

The following conditions hold:

1. For each $s \in [k]$, $0 \leq q \leq d$ the polynomial $\tilde{p}_{s,q}(x)$ belongs to the q -th Wiener chaos \mathcal{W}^q . Additionally, each polynomial $\text{In}(p_{s,q})_\ell$ with $q \geq 1$ lies in \mathcal{W}^j for some $1 \leq j \leq d$ and has $\mathbf{Var}[\text{In}(p_{s,q})_\ell] = 1$.
2. For each $s \in [k]$, $0 \leq q \leq d$, we have $\mathbf{Var}[p_{s,q} - \tilde{p}_{s,q}] \leq \tau$.
3. Each polynomial $\text{Out}(p_{s,q})$ is a multilinear polynomial in its $\text{num}(p_{s,q})$ arguments. Moreover, there exists $N =$

$N_\beta(k, d, \tau)$ and $M = M_\beta(k, d, \tau)$ such that if $\text{Coeff}(p_{s,q})$ denotes the sum of the absolute values of the coefficients of $\text{Out}(p_{s,q})$, then $\sum_{s,q} \text{Coeff}(p_{s,q}) \leq M$ and $\sum_{s,q} \text{num}(p_{s,q}) \leq N$.

4. Further, let $\text{Num} = \sum_{s=1, \dots, k, q=0, \dots, d} \text{num}(p_{s,q})$ and $\text{Coeff} = \sum_{s=1, \dots, k, q=0, \dots, d} \text{Coeff}(p_{s,q})$. Then, each polynomial $\text{In}(p_{s,q})_\ell$ is $\beta(\text{Num} + \text{Coeff})$ -eigenregular.

5. PROOF OF THEOREM 2

In this section we combine the tools developed in the previous sections to prove Theorem 2. We do this in two main steps. First we use the CLT from Section 3 and the decomposition procedure from Section 4 to reduce the original problem (of ϵ -approximately counting satisfying assignments of a degree- d PTF under $N(0, 1)^n$) to the problem of ϵ -approximating an expectation $\mathbf{E}_{G \sim N(0^r, \Sigma)}[\tilde{g}_c(G)]$, where $N(0^r, \Sigma)$ is a mean-0 r -dimensional Gaussian with covariance matrix Σ , and $\tilde{g}_c : \mathbb{R}^r \rightarrow [0, 1]$ is a particular explicitly specified function. The key points here are that the value of r , the description length (bit complexity) of \tilde{g}_c , and the bit complexity of each entry of the covariance matrix Σ are all $O_{d,\epsilon}(1)$ (completely independent of n). Next, we describe how an $O_{d,\epsilon}(1)$ -time deterministic algorithm can ϵ -approximate the desired expectation $\mathbf{E}_{G \sim N(0^r, \Sigma)}[\tilde{g}_c(G)]$. Theorem 2 follows directly from Theorems 15 and 16 which we state below.

THEOREM 15. *There is an $O_{d,\epsilon}(1) \cdot \text{poly}(n^d)$ -time deterministic algorithm with the following performance guarantee: Given as input a degree- d real polynomial $p(x_1, \dots, x_n)$ and a parameter $\epsilon > 0$, it outputs an integer r , a matrix of covariances $\Sigma \in \mathbb{R}^{r \times r}$ (whose diagonal entries are all 1), and a description of a function $\tilde{g}_c : \mathbb{R}^r \rightarrow [0, 1]$, such that $|\mathbf{Pr}_{x \sim N(0, 1)^n}[p(x) \geq 0] - \mathbf{E}_{G \sim N(0^r, \Sigma)}[\tilde{g}_c(G)]| \leq O(\epsilon)$. Moreover r is $O_{d,\epsilon}(1)$, the description length of \tilde{g}_c is $O_{d,\epsilon}(1)$ bits, and each entry of Σ is a rational number whose numerator and denominator are both integers of magnitude $O_{d,\epsilon}(1)$.*

THEOREM 16. *There is a deterministic $O_{d,\epsilon}(1)$ -time algorithm which, given as input the output r, Σ, \tilde{g}_c of Theorem 15 and $\epsilon > 0$, outputs a value ν such that*

$$|\nu - \mathbf{E}_{(G_1, \dots, G_r) \sim N(0^r, \Sigma)}[\tilde{g}_c(G_1, \dots, G_r)]| \leq \epsilon.$$

6. APPLICATION: A DETERMINISTIC FPT FOR APPROXIMATING ABSOLUTE MOMENTS

Consider the following computational problem, which we call **ABSOLUTE-MOMENT**: Given a degree- d polynomial $p(x)$ and an integer parameter $k \geq 1$, compute the value

$$\mathbf{E}_{x \in \{-1, 1\}^n} [|p(x)|^k]$$

of the k -th absolute moment of p . It is clear that the raw moment $\mathbf{E}[p(x)^k]$ can be computed in roughly n^k time by expanding out the polynomial $p(x)^k$, performing multilinear reduction, and outputting the constant term. Since the k -th raw moment equals the k -th absolute moment for even k , this gives an n^k time algorithm for **ABSOLUTE-MOMENT** for even k . However, as shown in [DDS13a], even for $d = 2$ the **ABSOLUTE-MOMENT** problem is $\#P$ -hard for any odd $k \geq 1$, and thus it is natural to seek approximation algorithms.

Using the hypercontractive inequality [Bon70, Bec75] it is not difficult to show that the obvious randomized algorithm (draw uniform points from $\{-1, 1\}^n$ and use them to empirically estimate $\mathbf{E}_{x \in \{-1, 1\}^n} [|p(x)|^k]$) with high probability gives a $(1 \pm \epsilon)$ -accurate estimate of the k -th absolute moment of p in in $\text{poly}(n^d, 2^{dk \log k}, 1/\epsilon)$ time. In this section we observe that Theorem 1 yields a *deterministic* fixed-parameter-tractable $(1 \pm \epsilon)$ -multiplicative approximation algorithm for ABSOLUTE-MOMENT:

THEOREM 17. *There is a deterministic algorithm which, given any degree- d polynomial $p(x_1, \dots, x_n)$ over $\{-1, 1\}^n$, any integer $k \geq 1$, and any $\epsilon > 0$, runs in $O_{d,k,\epsilon}(1) \cdot \text{poly}(n^d)$ time and outputs a value v that multiplicatively $(1 \pm \epsilon)$ -approximates the k -th absolute moment: $v \in [(1 - \epsilon) \cdot \mathbf{E}_{x \in \{-1, 1\}^n} [|p(x)|^k], (1 + \epsilon) \cdot \mathbf{E}_{x \in \{-1, 1\}^n} [|p(x)|^k]]$.*

Acknowledgement. We thank Ilias Diakonikolas for his contributions in the early stages of this project. We also thank Rafal Latala, Michel Ledoux, Elchanan Mossel, Ivan Nourdin and Krzysztof Oleszkiewicz for answering questions about the CLT. Part of this work was done when A.D. was hosted by Oded Regev and the Simons Institute. A.D. would like to thank them for their kind hospitality and support.

7. REFERENCES

- [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.
- [Bon70] A. Bonami. Etude des coefficients Fourier des fonctions 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.
- [CS13] D. Cartwright and B. Sturmfels. The number of eigenvalues of a tensor. *Linear algebra and its applications*, 432(2):942–952, 2013.
- [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.
- [DDS13a] A. De, I. Diakonikolas, and R. Servedio. Deterministic approximate counting for degree-2 polynomial threshold functions. manuscript, 2013.
- [DDS13b] A. De, I. Diakonikolas, and R. Servedio. Deterministic approximate counting for juntas of degree-2 polynomial threshold functions. manuscript, 2013.
- [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.
- [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.
- [FW95] J. Friedman and A. Wigderson. On the Second Eigenvalue of Hypergraphs. *Combinatorica*, 15(1):43–65, 1995.
- [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 fpts 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.
- [GOWZ10] P. Gopalan, R. O’Donnell, Y. Wu, and D. Zuckerman. Fooling functions of halfspaces under product distributions. In *IEEE Conf. on Computational Complexity (CCC)*, pages 223–234, 2010.
- [Hås94] J. Håstad. On the size of weights for threshold gates. *SIAM Journal on Discrete Mathematics*, 7(3):484–492, 1994.
- [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.

- [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.
- [Lat06] R. Latala. Estimates of moments and tails of gaussian chaoses. *Annals of Probability*, 34(6):2315–2331, 2006.
- [Lat13] R. Latala. Personal communication, 2013.
- [Led13] M. Ledoux. Personal communication, 2013.
- [LV96] M. Luby and B. Velickovic. On deterministic approximation of DNF. *Algorithmica*, 16(4/5):415–433, 1996.
- [LVW93] Michael Luby, Boban Velickovic, and Avi Wigderson. Deterministic approximate counting of depth-2 circuits. In *Proceedings of the 2nd ISTCS*, pages 18–24, 1993.
- [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.
- [MZ09] R. Meka and D. Zuckerman. Pseudorandom Generators for Polynomial Threshold Functions. Available at <http://arxiv.org/abs/0910.4122>, 2009.
- [MZ10] Raghu Meka and David Zuckerman. Pseudorandom generators for polynomial threshold functions. In *STOC*, pages 427–436, 2010.
- [NN93] J. Naor and M. Naor. Small-bias probability spaces: efficient constructions and applications. *SIAM J. on Comput.*, 22(4):838–856, 1993. Earlier version in STOC’90.
- [Nou12] I. Nourdin. Lectures on gaussian approximations with malliavin calculus. Technical Report <http://arxiv.org/abs/1203.4147v3>, 28 June 2012.
- [Nou13] I. Nourdin. Personal communication, 2013.
- [NP09] I. Nourdin and G. Peccati. Stein’s method meets malliavin calculus: a short survey with new estimates. Technical Report <http://arxiv.org/abs/0906.4419v2>, 17 Sep 2009.
- [NPR10] I. Nourdin, G. Peccati, and A. Réveillac. Multivariate normal approximation using Stein’s method and Malliavin calculus. *Ann. Inst. H. Poincaré Probab. Statist.*, 46(1):45–58, 2010.
- [Ole13] K. Oleszkiewicz. Personal communication, 2013.
- [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.
- [Pod09] V. V. Podolskii. Perceptrons of large weight. *Problems of Information Transmission*, 45(1):46–53, 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.
- [Tre04] L. Trevisan. A note on approximate counting for k -DNF. In *Proceedings of the Eighth International Workshop on Randomization and Computation*, pages 417–426, 2004.
- [Vio09] E. Viola. The Sum of d Small-Bias Generators Fools Polynomials of Degree d . *Computational Complexity*, 18(2):209–217, 2009.