Asymptotically Optimal Hitting Sets Against Polynomials

Markus Bläser
Saarland University, Saarbrücken, Germany
mblaeser@cs.uni-sb.de

Moritz Hardt
Princeton University, Princeton, NJ
mhardt@cs.princeton.edu

David Steurer
Princeton University, Princeton, NJ
dsteurer@cs.princeton.edu

November 20, 2007

Abstract

Our main result is an efficient construction of a hitting set generator against the class of polynomials of degree d_i in the i-th variable. The seed length of this generator is $\log D + O^{\sim}(\log^{1/2} D)$. Here, $\log D = \sum_i \log(d_i + 1)$ is a lower bound bound on the seed length of any hitting set generator against this class. Our construction is the first to achieve asymptotically optimal seed length for every choice of the parameters d_i. In fact, we present a nearly linear time construction with this asymptotic guarantee. Furthermore, our results extend to classes of polynomials parameterized by upper bounds on the number of nonzero terms in each variable.

Underlying all of our constructions is a general and novel framework that exploits the product structure common to the classes of polynomials we consider. This framework allows us to obtain efficient and asymptotically optimal hitting set generators from primitives that need not be optimal or efficient by themselves.

Finally, our results imply blackbox polynomial identity tests that use fewer random bits than previous methods.

1 Introduction

Let F be a class of polynomials in n variables over some field K. A *hitting set* against F is a set of points $H \subseteq K^n$ such that no polynomial in F vanishes on all points in H. Hitting sets against polynomials are fundamental objects in both computer science and mathematics. The existence of hitting sets follows for instance from Hilbert's Nullstellensatz. The Nullstellensatz shows that the polynomials vanishing on the algebraic set of some polynomial ideal I are precisely the radicals of I. Hence, the set H of common zeros of I is a hitting set against the non-radicals of I (assuming that K is algebraically closed). A fundamental property of every class of polynomials is the *minimum size* of a hitting set against it. We are interested in classes of polynomials that are parameterized by upper bounds on the degree or the number of nonzero terms. To be more precise, consider the class F of nonzero polynomials of degree at most d_i in the i-th variable. If we fix arbitrary sets $S_i \subseteq K$ of size $d_i + 1$, then the set $H = S_1 \times \cdots \times S_n$ is a hitting set against F of size $D = \prod_i (d_i + 1)$. This fact [AT92, Alo99] was used among other applications in proving the Combinatorial Nullstellensatz. The size of H can be shown to be optimal, that is, no set of size less than D can be a hitting set against F.

From a computational point of view, this hitting set has two shortcomings. First, it guarantees only to contain a single non-root of each polynomial in F. But for robustness, it would be more desirable if the non-roots of any polynomial in F had *high density* in H. Second, the size of the hitting set is exponential in the description length of F. The parameters $d_1, \ldots, d_n$ have description length about $N = \log D$ whereas the smallest hitting set against F has size 2^N. Hence, we cannot afford to enumerate all points of H. Instead, we would like to have an efficient *implicit representation* of the hitting set of length polynomial in N. This leads us to the following definition and problem statement.

Construction of Hitting Set Generators. A *hitting set generator* of *density* $\alpha > 0$ against a class of polynomials F is a function $G\colon \{0,1\}^r \to K^n$ such that for all $f \in F$ we have $\Pr[f(G(z)) \neq 0] \geq \alpha$ where the *seed* $z \in \{0,1\}^r$ is drawn uniformly at random. The parameter r is called the *seed length* of the generator G. We think of G as an implicit representation of a hitting set of size at most 2^r. Further, suppose T is a family parameters such that for every parameter $t \in T$ we have defined a class of polynomials $F(t)$. Given parameters $t \in T$ and $\epsilon > 0$, the problem is to construct a hitting set generator G of density $1 - \epsilon$ against $F(t)$ such that the seed length of G is minimized. The construction algorithm should be deterministic and run in time $\mathrm{poly}(|t|)$, where $|t|$ is the length of the binary encoding of t. The required output of the algorithm is the description of a boolean circuit implementing G. The output of the circuit is understood as a natural binary encoding of a tuple of field elements.

Polynomial Identity Testing. Closely related is the question of polynomial identity testing. Here, we assume we are given access to a polynomial in some implicit representation. The problem is to distinguish the case where the given polynomial is identically zero from the case where the polynomial is member of some class $F \subseteq K[x_1, \ldots, x_n]$. Provided with a hitting set generator of high density against F, this can be done by picking a random seed and testing if the given polynomial is zero at the point produced by the generator. While the zero polynomial will always be zero on this point, any polynomial in F will evaluate to a nonzero value with high probability. This does not require more than "blackbox access" to the polynomial.

The study of polynomial identity testing was initiated by the work of DeMillo, Lipton, Schwartz and Zippel [DL78, Zip79, Sch80]. The observation is, if the size of the sets S_i in our above example

is $2nd_i$ instead of $d_i + 1$, then the hitting set generator G which chooses a random point from the set $H = S_1 \times \cdots \times S_n$ has density $1/2$ against the same class of polynomials. The seed length of G is roughly $\log D + n \log n$.

Since many problems reduce to checking polynomial identities, this early work entailed a variety of efficient randomized algorithms [CRS95, Lov79, MVV87, BCW80, BK95, AB03]. Similarly, several results in complexity theory [Sha92, LFKN92, AS98, ALM+98] involve hitting set generators against polynomials as a subroutine.

What remained wide open after this initial work is the question how much randomness is required in testing polynomial identities. There were two successful approaches towards making progress on this question. One is giving deterministic identity tests for restricted classes of polynomials and arithmetic circuits [Agr05, DS07, KS06, Shp07]. Testing general arithmetic circuits for identity in even subexponential deterministic time is linked to circuit lower bounds [KI04, Agr05]. The other approach has been to minimize the seed length of hitting set generators against more general classes of polynomials [CK00, LV98, KS01, Bog05].

In this work we continue the study of the latter question. For many classes of polynomials parameterized by degree or sparsity we are able to settle the question by giving constructions of hitting set generators whose seed lengths match the following lower bound asymptotically.

Lower Bound. Consider a class of polynomials $F \subseteq K[x_1, \ldots, x_n]$ for which there is a linear space $W \subseteq F \cup \{0\}$ of dimension at least d. As we fix any set of strictly less than d points in K^n, the space V of polynomials vanishing on these points has co-dimension strictly less than d. Hence, the intersection space $V \cap W$ has positive dimension. In particular, it contains a nonzero polynomial. We conclude that any hitting set generator of density $1 - \epsilon$ against F needs a support of size at least d/ϵ. In other words, the seed length is at least $r \geq \log(d/\epsilon)$.

1.1 Our Result

We introduce a general framework for obtaining efficient and asymptotically optimal constructions from primitives that need not be optimal or even efficient by themselves. Our framework requires the target class of polynomials to exhibit a typical product structure that we formalize. We exploit this structure by working with product operations on hitting set generators. A crucial primitive in our framework are hitting set generators which besides their seed have an additional source of randomness, called *random advice*. Random advice captures excess in randomness that can be shared when computing the product of two generators. Our constructions will generally be the product of several generators each working on one subset of the variables. A simple approximation algorithm determines a partition of the variables so as to minimize seed length, runtime or the required field size of our construction. In fact, all of our results are variations of the following two steps. We design from scratch some novel generators with optimal seed length but random advice. Next, our framework allows us to turn these generators into asymptotically optimal hitting set generators without any advice.

We say a polynomial f has degree $\mathbf{d} = (d_1, \ldots, d_n)$, if d_i is an upper bound on the degree of the i-th variable in f. We let $F(\mathbf{d}) \subseteq K[x_1, \ldots, x_n]$ denote the class of nonzero degree-$\mathbf{d}$ polynomials in n variables. We use the abbreviation $D = \prod_{i=1}^{n}(d_i + 1)$ throughout our work.

Theorem 1 *Given any degree $\mathbf{d} = (d_1, \ldots, d_n)$, we can efficiently construct a hitting set generator of density $1/2$ against $F(\mathbf{d})$ with seed length $\log D + \tilde{O}(\sqrt{\log D})$.*

Since the quantity D is the dimension of the space $F(\mathbf{d}) \cup \{0\}$, the lower bound implies that the seed length is asymptotically optimal for the entire family of parameters $d_1, \ldots, d_n$ where $n, d_i \in \mathbb{N}$. The multiplicative excess in seed length decreases inverse-polynomially in $\log D$. We will prove the statement generalized to arbitrary density. Our result holds over large enough finite fields and over any field of characteristic zero. These assumptions are roughly the same as those of the Schwartz-Zippel Lemma. It is worth noting, over fields of characteristic zero, our construction does not depend on the size of the coefficients of the polynomials. The dependence on each degree d_i is only logarithmic which makes our construction efficient even for high degrees.

In addition, we show how to obtain a nearly linear time construction at the cost of slightly more but still asymptotically optimal seed length. More generally we get the precise trade-off between runtime $O(\log^{1+\delta} D)$ and seed length $\log D + O\tilde{}(\log^{1-\delta} D)$ where $\delta \in (0, 1/2)$.

Furthermore, we extend our work to classes of polynomials where we are given an upper bound on the number of nonzero terms. Our notion of sparsity is analogous to the previous notion of degree. We say a polynomial f has *sparsity* $\mathbf{m} = (m_1, \ldots, m_n)$, if f has at most m_i nonzero terms when written as a univariate polynomial in the i-th variable. For any tuple $\mathbf{m} = (m_1, \ldots, m_n)$ and any integer $d \in \mathbb{N}$, we define $F(\mathbf{m}, d)$ as the class of nonzero sparsity-$\mathbf{m}$ polynomials of total degree at most d. Henceforth, let $M = \prod_{i=1}^{n} m_i$.

Theorem 2 *For any sparsity $\mathbf{m} = (m_1, \ldots, m_n)$ and any degree d where $d \leq M$, we can efficiently construct a hitting set generator with seed length $\log M + O\tilde{}(\sqrt{\log M \cdot \log d})$ and density $1/2$ against $F(\mathbf{m}, d)$ over any large enough finite field.*

The lower bound shows any hitting set generator of positive density against $F(\mathbf{m}, d)$ has seed length at least $\log M$, provided that d is sufficiently large, i.e., $d \geq \sum_{i=1}^{n} m_i$. Hence, the seed length of our generator is asymptotically optimal whenever $\log d = o(\log M/(\log \log M)^c)$ for some absolute constant c.

Theorem 3 *Given $\delta > 0$, $\mathbf{m}$, and d, we can construct in time $\mathrm{poly}(\log^{1/\delta} M, n \log d)$ a generator G such that G has density $1/2$ against $F(\mathbf{m}, d)$ over any field of characteristic zero and the seed length of G is $(1 + \delta) \log M + O(\log \log M + \log \log d)$.*

In the above theorem, for $\log \log d = o(\log M)$, the seed length can be made arbitrarily close in a multiplicative sense to the lower bound $\log M$ at the expense of a higher running time. This trade-off is comparable to the time-approximation trade-off in polynomial time approximation schemes (PTAS). The theorem is weaker than our other results in that it gives only quasi-polynomial time constructions of generators with asymptotically optimal seed length. However, in contrast to all previously known constructions against $F(\mathbf{m}, d)$, the dependence of the seed length on the total degree is not logarithmic but doubly-logarithmic. We obtain this exponential improvement by combining Descartes' Rule of Signs with an improved version of a reduction in [KS01].

Previous Work. The Schwartz-Zippel Lemma gives a generator against $F(\mathbf{d})$ of seed length $\log D + n \log n$ which is asymptotically optimal for large degree, i.e., $\log D = \omega(n \log n)$. Only recently, Bogdanov [Bog05] obtained improvements in the case where the total degree d of the polynomials is much smaller than the number of variables n, e.g., $d = O(\log n)$. Several results are concerned with the case where $\log D$ is comparable to n. Chen and Kao [CK00] achieve the seed length $\sum_{i=1}^{n} \lceil \log(d_i + 1) \rceil$. Their construction works only for polynomials with integer coefficients and has some dependence on the size of those coefficients. Strictly speaking, this is why our lower bound argument does not apply to their setting. Lewin and Vadhan [LV98] generalize the

techniques of Chen and Kao to fields of positive characteristic. While these upper bounds are as good as $\log D$ for some configurations of the parameters, they come arbitrarily close to $\log D + n$ in general. As we think of $\log D = \Theta(n)$, this is not asymptotically optimal. In fact, speaking in terms of the size of hitting sets, this is a multiplicative excess of order 2^n. Furthermore, both construction algorithms have a polynomial runtime dependence on each degree d_i. As soon as a single degree d_i is superpolynomial in n, their algorithms are not efficient. Notice, this range of d_i is natural even if $\log D = O(n)$. Small arithmetic circuits can compute polynomials of very high degree in a single variable. In the arithmetic circuit model, Agrawal and Biswas [AB03] give a polynomial identity test that uses $\log D$ random bits. However, in this case we have no lower bound. In particular, if P = coRP, then there is a *deterministic* polynomial time arithmetic circuit identity test [Sch80, IM83]. However, a particular tool introduced by [AB03] turns out to give us hitting set generators of the optimal seed length $\log(D)$ over finite fields of size at least D. This tool will be used and discussed later. We will see how to achieve asymptotically the same seed length over significantly smaller finite fields.

Klivans and Spielman [KS01] improve the Schwartz-Zippel Lemma when given information about the sparsity of the polynomials. Specifically, they construct a hitting set generator against the class of n-variate polynomials of total degree d and at most m nonzero terms. The seed length is $O(\log(mnd))$. This is better than previous work if $\log m = o(n \log d)$. In order to compare these results with the work of Klivans and Spielman, we can think of the quantity $M = \prod m_i$ as some approximation of the number of nonzero terms m. Notice that always $M \geq m$ and in general M can be strictly larger than m. The polynomial $1 + x_1 \cdots x_n$ has only two nonzero terms, but $m_i = 2$ for all $i \in [n]$ and thus $M = 2^n$. In general, our parameterization allows only $M \geq 2^n$, since all variables with $m_i = 1$ can be fixed to an arbitrary nonzero constant.

In Figure 1 we compare our results to the previous work in terms of the normalized size of the hitting set that we can efficiently represent and the time it takes to compute the implicit representation itself. Unless otherwise specified, the density α is fixed to be a constant, say, $1/2$. In the table, we neglect polylogarithmic factors in the runtime. The parameter δ may be chosen arbitrarily from the range $(0, 1/2)$.

Size/D	Runtime		Source
	char $= 0$	char > 0	
$1 \quad \alpha = 1/D$ $n^n \quad \alpha = 1/2$	$\log D$		[Sch80, Zip79, DL78, AT92]
2^n	poly(nd)	poly(qd)	[CK00, LV98]
1	poly(D)	poly$(q \log D)$	Kronecker substitution [AB03]
$D^{1/\log^\delta D}$	$\log^{1+\delta} D$	poly$(q \log D)$	This work
Size/M			
$d \cdot M^c$	poly$(\log M \cdot \log d)$	poly$(q \log M)$	[KS01]
$\log d \cdot M^\delta$	poly$(\log^{1/\delta} M \cdot \log d)$	—	This work
$d \cdot M^{\frac{\log^{1/2} d}{\log^{1/2} M}}$	—	poly$(q \cdot \log M)$	This work

Figure 1: Comparison of hitting set size and construction time

Notation We will fix some notation for the rest of this paper. If k and $m > 0$ are integers, then $\lfloor k \rfloor_m$ denotes the remainder of k modulo m. For a tuple $\mathbf{k} = (k_1, \ldots, k_n)$, we let $\lfloor \mathbf{k} \rfloor_m =$

$(\lfloor k_1 \rfloor_m, \ldots, \lfloor k_n \rfloor_m)$. The set of integers $\{1, \ldots, n\}$ is abbreviated by $[n]$. The notation $O\tilde{}(t)$ is used to suppress polylogarithmic factors of t, that is, $O\tilde{}(t) = O(t \cdot (\log t)^{O(1)})$.

2 Direct Products, Shared Advice, and Balanced Factors

In this section we give the technical exposition of our framework. It consists of three parts, product operations on hitting set generators and classes of polynomials, the notion of random advice, and an algorithmic approach working with these tools.

Definition 1 (Direct product) For two generators $G_1 \colon \{0,1\}^{r_1} \to K^{n_1}$ and $G_2 \colon \{0,1\}^{r_2} \to K^{n_2}$, we define the *direct product* $G_1 \otimes G_2 \colon \{0,1\}^{r_1+r_2} \to K^{n_1+n_2}$ to be the function defined by $G_1 \otimes G_2\ (z_1 z_2) = (G_1(z_1), G_2(z_2))$.

Clearly, if both G_1 and G_2 can be constructed efficiently, then so can the product $G_1 \otimes G_2$.

Now, suppose we have two hitting set generators with high density against two classes F_1 and F_2, respectively. We want to identify a large class of polynomials $F_1 F_2$ against which the direct product still has high density.

Definition 2 (Schwartz-Zippel product) Let $F_1 \subseteq K[\mathbf{x}_1]$ and $F_2 \subseteq K[\mathbf{x}_2]$ be two classes of polynomials on disjoint sets of variables $\mathbf{x}_1$ and $\mathbf{x}_2$, respectively. Let $n_i = |\mathbf{x}_i|$. We define the *Schwartz-Zippel product* $F_1 F_2$ to be the set of polynomials $f \in K[\mathbf{x}_1, \mathbf{x}_2]$ such that f as a polynomial in $\mathbf{x}_2$ has a coefficient $g \in K[\mathbf{x}_1]$ satisfying the following two properties:

- g is a member of F_1, and
- for every $\mathbf{a}_1 \in K^{n_1}$ with $g(\mathbf{a}_1) \neq 0 \in K$, the polynomial $f(\mathbf{a}_1, \mathbf{x}_2) \in K[\mathbf{x}_2]$ is a member of F_2.

Intuitively, this is the same product structure required in the well-known proof the Schwartz-Zippel Lemma. As desired we get the following lemma.

Lemma 1 *Let G_1 and G_2 be two generators, and let $F_1 \subseteq K[\mathbf{x}_1]$ and $F_2 \subseteq K[\mathbf{x}_2]$ be two classes of polynomials. Suppose that G_1 has density α_1 against F_1 and G_2 has density α_2 against F_2. Then, the direct product $G_1 \otimes G_2$ has density $\alpha_1 \alpha_2$ against the Schwartz-Zippel product $F_1 F_2$.*

Proof. Immediate from the definition of the Schwartz-Zippel product. □

We introduce hitting set generators with an additional source of randomness, called *random advice*.

Definition 3 (Advised generator) We call a function $G \colon \{0,1\}^a \times \{0,1\}^r \to K^n$ an *advised generator* with *seed length* $r(G) := r$ and *advice length* $a(G) := a$. We say an advised generator G has *quality* $1 - \epsilon$ against a class F of polynomials, if the generator $G(y, \cdot)$ has density $1 - \epsilon/2$ against F with probability $1 - \epsilon/2$ for a randomly chosen string $y \in \{0,1\}^a$. Formally,

$$\Pr_{y \in \{0,1\}^a} \left(\forall f \in F.\ \Pr_{z \in \{0,1\}^r} [f(G(y,z)) \neq 0] \geq 1 - \tfrac{\epsilon}{2} \right) \geq 1 - \tfrac{\epsilon}{2}.$$

We define the *advice-less generator* $\bar{G} \colon \{0,1\}^{a+r} \to K^n$ corresponding to G to be the function defined by $\bar{G}(yz) = G(y,z)$. Here yz denotes the string obtained from y and z by concatenation.

Fact 2 *If G has quality α against F, then $\bar{G}$ has density α against F.*

Definition 4 (Shared advice product) For two advised generators $G_1\colon \{0,1\}^{a_1} \times \{0,1\}^{r_1} \to K^{n_1}$ and $G_2\colon \{0,1\}^{a_2} \times \{0,1\}^{r_2} \to K^{n_2}$ with $a = \max\{a_1, a_2\}$, we define the *shared-advice product* $G_1 \otimes G_2\colon \{0,1\}^{a} \times \{0,1\}^{r_1+r_2} \to K^{n_1+n_2}$ to be the function defined by $G_1 \otimes G_2\ (y, z_1 z_2) = (G_1(y, z_1), G_2(y, z_2))$. Here we assume that G_i ignores all but the first a_i advice bits.

We can compute the shared-advice product at a moderate loss of quality.

Lemma 3 *Let $\{G_i\}_{i\in[k]}$ be a set of advised generators, and let $\{F_i\}_{i\in[k]}$ be a set of classes of polynomials. Suppose the generator G_i has quality $1-\epsilon$ against F_i. Then, the shared-advice product $G = \bigotimes_i G_i$ has quality $1 - k\epsilon$ against the Schwartz-Zippel product $\prod_{i\in[k]} F_i$.*

Proof. With probability $1 - k\epsilon/2$, each generator $G_i(y, \cdot)$ has density $1 - \epsilon/2$ against F_i. Condition on this event. By Lemma 1, the direct product $G(y, \cdot) = \bigotimes_i G_i(y, \cdot)$ has density $(1-\epsilon/2)^k > 1-k\epsilon/2$ against $\prod_i F_i$. □

Balanced Factors. The previous discussion gives rise to the following construction approach. Recall, our goal is a hitting set generator against some class of polynomials $F \subseteq K[x_1, \ldots, x_n]$. In a first step we identify classes $F_1, \ldots, F_k$ such that F is contained in the Schwartz-Zippel product $\prod_{i\in[k]} F_i$. We think of these classes F_i as factors of F. This step induces a partition of the variables into k parts. We will design advised generators G_i against each F_i, respectively. Each G_i works on one subset of the variables. Then we combine them into one generator G using the shared advice product. Our final candidate is the seedless generator $\bar{G}$. Since the quality of G suffers when the number of factors k is large, it is desirable to have a partition of the variables that consists of not too many parts. But once we determined k, what is a *good* partition of the variables for this choice of k? We want to have a partition that is balanced in the following sense. Suppose we can associate a weight with each variable such that the total weight of a subset of the variables corresponds to the length of advice needed by a generator G_i operating on this set of variables. Since we can share advice, the goal is to find a partition of the variables that distributes the weight equally among all parts. For technical reasons, we can allow that parts containing only a single variable have large weight.

Lemma 4 *Given a positive integer k and a polynomial ring $K[\mathbf{x}]$ with nonnegative weights $w\colon [n] \to \mathbb{R}_{\geq 0}$ on the variables, we can efficiently compute a partition $(S_1, \ldots, S_k)$ of the set $S = [n]$ of variables such that each part S_i either contains only a single variable or else the total weight of the variables in S_i is at most $w(S_i) \leq 4w(S)/k$.*

Proof. There are at most $\lfloor k/2 \rfloor$ variables with $w(i) > 2w(S)/k$. Each of these variables is put in a singleton set. The remaining variables are distributed among the at least $\lceil k/2 \rceil$ remaining sets using a greedy algorithm that aims to minimize the maximum weight of a set. □

Ultimately, we can always choose k so as to minimize seed length of our construction. But varying over k also gives rise to interesting trade-offs.

To illustrate the initial step of our approach, consider the class $F(\mathbf{d})$ of polynomials in some n variables. Let $(S_1, \ldots, S_k)$ be a partition of the set of coordinates $[n]$ and further let $\mathbf{d}_i$ denote the restriction of $\mathbf{d}$ to the coordinates in S_i. By inspection of the definition, we see that the Schwartz-Zippel product $\prod_i F(\mathbf{d}_i)$ is a superset of $F(\mathbf{d})$.

3 Polynomials of a Given Degree

We begin with the basic building blocks in our construction. For univariate polynomials we will need a simple generator that picks a random field element from a large enough range. We define the *trivial generator with seed length* r to be the generator $G\colon \{0,1\}^r \to K$ that outputs a field element that corresponds in fixed way to its seed. For example, if $\mathrm{char}(K) = 0$ or $\mathrm{char}(K) \geq 2^r$, G would output the field element corresponding to the binary number encoded by its seed, that is, $G(z_0 \cdots z_{r-1}) = \sum_{i=0}^{r-1} z_i(1+1)^i \in K$.

Proposition 5 *The trivial generator G with seed length $\log(d/\epsilon) + O(1)$ has density $1-\epsilon$ against the class of univariate polynomials over a field K of degree at most d, provided that K has size at least d/ϵ.*

We further use the Kronecker substitution as introduced by [AB03] for our choice of parameters.

Definition 5 Let $\mathbf{d} = (d_1, \ldots, d_n) \in \mathbb{N}^n$. Let $D_i = \prod_{0<j<i}(d_j + 1)$ for all $i \in \{1, \ldots, n\}$. The *Kronecker substitution* $\mathbf{kr} \in (K[X])^n$ with respect to $\mathbf{d}$ is defined as $\mathbf{kr}(X) = (X^{D_1}, \ldots, X^{D_n})$.

Lemma 6 *Let $f \in F(\mathbf{d})$. Then, $f(\mathbf{kr}(X)) \in K[X]$ is a a univariate polynomial of degree at most $D-1$. Furthermore, for any two distinct monomials w and w' in f, we have $w(\mathbf{kr}(X)) \neq w'(\mathbf{kr}(X))$. In particular, $f(\mathbf{kr}(X))$ is not the zero polynomial in $K[X]$.*

Proof. The degree bound is a direct consequence of the fact that $\sum_{i=1}^{n} d_i D_i = D-1$ as easily shown by induction. Now suppose we have two distinct monomials $w = \prod_i x_i^{u_i}$ and $w' = \prod_i x_i^{v_i}$ in f. We argue that under Kronecker substitution these monomials are mapped to distinct exponents of X. To see this, pick the largest index j for which $u_j \neq v_j$. Say $u_j > v_j$. We have $\sum_i (u_i - v_i)D_i > D_j - \sum_{i<j}(u_i - v_i)D_i$. But $u_i - v_i < d_i$ and further $\sum_{i<j} d_i D_i < D_j$. □

Remark 1 Over finite fields of cardinality at least D/ϵ, this lemma immediately gives us a generator G of density $1-\epsilon$ and *optimal* seed length. We simply combine Lemma 6 with Proposition 5. More precisely, we generate points of the form $\mathbf{kr}(s)$ where the element s is drawn uniformly at random from a subset of the field of size D/ϵ.

Unfortunately, over fields of characteristic zero the bit size of $\mathbf{kr}(s)$ is at least D. But that is exponential in the desired runtime of our construction algorithm. We show how to solve this problem at the cost of random advice.

Proposition 7 *Let K be of characteristic zero. For any degree $\mathbf{d}$ and any $\epsilon > 0$ we can construct a hitting set generator G of quality $1-\epsilon$ against $F(\mathbf{d})$ in time polynomial in $\log(D/\epsilon)$. Furthermore, $r(G) = \log(D/\epsilon) + O(1)$ and $a(G) = O(\log(D/\epsilon))$.*

Proof. First, the generator G uses its advice string y in order to obtain a number $p = p(y) > 2D/\epsilon$ such that $\Pr_y[p(y) \text{ is prime}] > 1 - \epsilon/2$. This can be done efficiently with an advice string of length $O(\log(D/\epsilon))$. An efficient algorithm for generating an N-bit prime number with high probability does not need more than $O(N + \log(1/\epsilon))$ random bits (see Appendix C). Second, G uses its seed to choose a random field element s from the range $R = \{1, \ldots, \lceil 2D/\epsilon \rceil\}$. Finally, G outputs the point $\lfloor \mathbf{kr}(s) \rfloor_p \in K^n$ where $\mathbf{kr}$ denotes the Kronecker substitution with respect to $\mathbf{d}$.

We claim whenever $p(y)$ is a prime number, then $G(y, \cdot)$ has density $1-\epsilon$ against $F(\mathbf{d})$. This will conclude our proof. So, suppose $p(y)$ is prime and let $f \in F(\mathbf{d})$. It suffices to show that there

are at most $D-1$ points $s \in R$ such that $f(\lfloor \mathbf{kr}(s) \rfloor_p) = 0$. To obtain a contradiction, suppose there were D such points $S \subseteq R$.

This determines a homogeneous system of D linear equations in D variables representing the coefficients of f. The associated matrix A of this system has entries $A_{st} = \prod_{i=1}^{n} \lfloor s^{D_i} \rfloor_p^{t_i}$ where $s \in S$ and the $t_i \in \{0, \ldots, d_i\}$ are such that $t = \sum t_i D_i$. Notice that this representation of t is unique (cf. Lemma 6). Now observe, $A_{st} = s^t \mod p$. This means in $\mathrm{GF}(p)$, the matrix A is simply a $D \times D$ Vandermonde matrix with respect to the points in S. Since $p > D$, this Vandermonde matrix is nonsingular over $\mathrm{GF}(p)$. Hence, A is nonsingular over K and therefore the only solution to the system is zero. But $F(\mathbf{d})$ does not contain the zero polynomial. □

Clearly, the advice length in Proposition 7 could be reduced, if we were able to generate the prime number in the proof of the proposition with significantly less random bits. This connection is discussed in Appendix C.

3.1 Proof of the Main Theorem

We prove a more general statement from which we deduce the main theorem.

Theorem 4 *Let $\mathbf{d} = (d_1, \ldots, d_n)$ and $\epsilon > 0$. Then, for any $k \in \{1, \ldots, n\}$, we can efficiently construct a hitting set generator of density $1-\epsilon$ against $F(\mathbf{d})$ and seed length $\log(D/\epsilon) + O(k \log(k/\epsilon)) + O(\log(D/\epsilon)/k)$.*

Proof. Define the weight of the variable x_i as $w(i) = \log(d_i+1)$. Apply Balanced Factors (Lemma 4) with the given choice of k so as to obtain a partition of the coordinates $[n]$ into sets $S_1, \ldots, S_k$. Let $\mathbf{d}_i$ denote the restriction of $\mathbf{d}$ to the coordinates in S_i. For each $i \in [k]$ we will construct an advised generator G_i against $F(\mathbf{d}_i)$ of quality $1 - \epsilon/2k$. If $|S_i| = 1$, then we obtain G_i from Proposition 5. In this case $a(G_i) = 0$. Whenever $|S_i| > 1$, we obtain G_i from Proposition 7.

Consider the advised generator $G = \bigotimes_{i \in [k]} G_i$. This is a generator against the Schwartz-Zippel product $\prod_{i \in [k]} F(\mathbf{d}_i)$ which is a superset of $F(\mathbf{d})$. The quality requirement follows from Lemma 3. Notice, $r(G) = \sum_{i=1}^{k} r(G_i) = \sum_{i=1}^{k} \log(D_i) + O(k \log(k/\epsilon))$ where $D_i = \prod_{j \in S_i} (d_j + 1)$. But, $\sum_i \log(D_i) = \log D$. Hence, $r(G) = \log D + O(k \log(k/\epsilon))$. On the other hand, $a(G) = \max_i O(\log(D_i) + \log(1/\epsilon))$. But the Balanced Factors Lemma guarantees $\log(D_i) = w(S_i) = O(w(S)/k) = O((\log D)/k)$. Therefore, we obtain the desired generator by combining seed and advice of G (see Fact 2). □

Corollary 8 (implies Theorem 1) *We can efficiently construct a hitting set generator against $F(\mathbf{d})$ of density $1-\epsilon$ and seed length $\log(D/\epsilon) + \tilde{O}(\sqrt{\log D \cdot \log(1/\epsilon)})$.*

Proof. Choose $k = \lceil \sqrt{\log D / \log(1/\epsilon)} \rceil$ in the previous theorem. □

Nearly Linear Time. The larger we choose k the more efficient is our construction. Notice the trivial generators from Proposition 5 can be constructed in time linear in their seed length. But to construct a generator from Proposition 7 we need more time. Let us say time $\bar{N}^c$ for some constant $c > 1$ where $\bar{N}$ is the length of the input parameters. In the context of the above theorem, let $N = \log D$. For simplicity fix the density to be some constant. The Balanced Factors Lemma guarantees that the seed and advice length of any advised generator used in our construction is bounded by $O(N/k)$. Hence, the time it takes to construct all advice generators will be no more

than $O(k \cdot (N/k)^c) = O(N^c/k^{c-1})$. As we set $k = N/(\log N)^{c+1}$, the over all construction time becomes $O\tilde{}(N)$. The seed length remains within $(1 + o(1))\mathrm{OPT}$. More generally, setting $k = N^{1-\delta}$ for any $\delta \in (0, 1/2)$ gives us the trade-off between time $N^{1+(c-1)\delta}$ and seed length $N + O\tilde{}(N^{1-\delta})$.

We point out that the exponent $1 + (c-1)\delta$ can be improved to $1 + \delta$ in our case. We compute the prime number required in the proof of Proposition 7 only once centrally and pass it on to all generators. This requires cubic time in the bit size of the prime number (see Appendix C). Provided with a prime number, the generators in Proposition 7 can be constructed in quadratic time, that is, the above c need not be larger than 2.

Smaller Finite Fields. We obtain a similar trade-off for the required size of finite fields. Notice, in this case we use the generator described in Remark 1 instead of Proposition 7 in the proof Theorem 4. So, suppose we want to construct a generator of density $1 - \epsilon$ against $F(\mathbf{d})$. For $k = 1$, Theorem 4 requires a field of size $q > D/\epsilon$. But for $k = n$, we observe that $q > \frac{n}{\epsilon} \cdot \max_i d_i$ is sufficient, which is also the minimum field size required by the Schwartz-Zippel Lemma. For general k, a field of size at least $\frac{k}{\epsilon} \cdot D^{c/k}$ is sufficient for absolute constant c.

4 Polynomials with a Given Number of Nonzero Terms

Let K be a sufficiently large finite field. In this section, we give an efficient construction of hitting set generators against $F(\mathbf{m}, d)$ with asymptotically optimal seed length, provided $\log d \ll \log M$. In the previous section, our basic building blocks were generators against the target class $F(\mathbf{d})$ that have optimal seed length, but require some amount of advice. For the target class $F(\mathbf{m}, d)$, however, we do not have advised generators with optimal seed length, even if we allow an arbitrary amount of advice. Instead we will start from generators that have a close to optimal seed length against certain subclasses $F(w, W)$ of $F(\mathbf{m}, d)$. Specifically, for a set of monomials W and a monomial $w \in W$, we let $F(w, W)$ be the set of polynomials over K that are in the linear span of W but not in the span of $W \setminus \{w\}$. In other words, $F(w, W)$ consists of all polynomials $f \in K[\mathbf{x}]$ such that w has a nonzero coefficient in f and all other monomials of f are in W. Note that all polynomials in $F(w, W)$ are nonzero.

Proposition 9 *Given $\mathbf{m}$, d, and $\epsilon > 0$, we can efficiently construct an advised generator G with $r(G) = \log M + O(\log nd/\epsilon)$ and $a(G) = O(\log(dM/\epsilon))$ such that G has quality $1 - \epsilon$ against every class $F(w, W) \subseteq F(\mathbf{m}, d)$.*

Intuitively, the proposition asserts that for every choice of w and W, most of the advice strings y give a generator $G(y, \cdot)$ that is dense against $F(w, W)$. On the other hand, possibly no single advice string yields a generator that is dense against $F(\mathbf{m}, d)$. We defer the proof to the end of the section. Its main ingredient is a randomized reduction from multivariate to univariate polynomials [KS01] (for a similar reduction see Lemma 13).

Using Proposition 9 as our basic building block, our construction against $F(\mathbf{m}, d)$ essentially works as follows. First, we compute a *balanced partition* $(S_1, \ldots, S_k)$ of the coordinates $[n]$ (Lemma 4). Here we use $w(j) = \log m_j$ as the weight function. Then, from the above proposition, we obtain generators G_i that have high quality against any class $F(w_i, W_i)$ contained in $F(\mathbf{m}_i, d)$, where $\mathbf{m}_i$ is the restriction of $\mathbf{m}$ to the coordinates in S_i. Since the partition $(S_i)_{i \in [k]}$ was balanced, the *shared-advice product* $G = \bigotimes_i G_i$ has only advice length about $\frac{1}{k} \log M$. On the other hand, the seed length of G is close to the lower bound $\log M$.

We claim that the advice-less generator $\bar{G}$ corresponding to G has high density against $F(\mathbf{m}, d)$. By Lemma 3, G has high quality against any product $\prod_i F(w_i, W_i)$ with $F(w_i, W_i) \subseteq F(\mathbf{m}_i, d)$. Hence, $\bar{G}$ has high density against the union of all such products. Finally, $\bar{G}$ has high density against $F(\mathbf{m}, d)$, because every polynomial in $F(\mathbf{m}, d)$ is contained in one of the products $\prod_i F(w_i, W_i)$.

The proof of Theorem 3 is deferred to Appendix A.1.

Proof (of Proposition 9). Let $t = 4mnd/\epsilon$. The generator G uses the advice string $y = y_1y_2$ in order to compute a number $p = p(y_1)$ of magnitude between t and $2t$ such that p is prime with high probability over a random choice of y_1. More precisely, $\Pr_{y_1}[\, p \text{ is prime}\,] \geq 1 - \epsilon/4$. Further, G computes a number $k = k(y_2)$ from the range $[t]$ such that k is uniformly distributed in $[t]$ when y_2 is chosen at random. Then, G uses the seed z in order to choose a random number $b = b(z)$ from some range in K of size $\lceil 4td/\epsilon \rceil$. Finally, G outputs the point $(b^{\lfloor k^{i-1} \rfloor_p})_{i \in [n]}$. By construction, G has advice length $a(G) = |y_1| + |y_2| = O(\log(t/\epsilon) + \log t) = O(\log mnd/\epsilon)$ and seed length $r(G) = \log(4td/\epsilon) = \log m + O(\log nd/\epsilon)$.

Let $F(w, W) \subseteq F(m, d)$. It remains to show that G has quality $1-\epsilon$ against $F(w, W)$. Consider the substitution σ from $K[\mathbf{x}]$ to $K[X]$ with $\sigma(x_i) = X^{\lfloor k^{i-1} \rfloor_p}$ for $i \in [k]$. Note that σ depends only on the advice string for G. We say that an advice string is (w, W)-good if $\sigma(w) \neq \sigma(w')$ for every $w' \in W$ with $w \neq w'$. By Lemma 2 in [KS01], a random advice string is (w, W)-good with probability at least $1 - mn/t \geq 1 - \epsilon/2$. On the other hand, if an advice y is (w, W)-good, then the generator $G(y, \cdot)$ has density at least $1 - \epsilon/2$, because then the substitution σ maps every member of $F(w, W)$ to a nonzero polynomial in $K[X]$ of degree at most $p \cdot d \leq 2td$ and the range of b has cardinality at least $4td/\epsilon$. □

We elaborate on why we need to work with the subclasses $F(w, W)$ instead of other more natural candidates in Remark 2 (see Appendix A).

4.1 Over Fields of Characteristic Zero

Let K be a field of characteristic 0. Lipton and Vishnoi [LV03] point out the following fact.

Proposition 10 *For every $\epsilon > 0$, the trivial generator with seed length $\log(m/\epsilon) + O(1)$ has density $1 - \epsilon$ against the class of univariate polynomials with at most m nonzero terms.*

Proof. A univariate polynomial with at most m nonzero terms has at most m rational roots over any field of characteristic zero. This follows in particular from Descartes' Rule of Signs. □

Let $F(W)$ denote the set of nonzero polynomials in the linear span of W.

Proposition 11 *Given $\epsilon > 0$, $\mathbf{m}$, and d, we can construct an advised generator G with $r(G) = \log m/\epsilon + O(1)$ and $a(G) = O(\log(mn/\epsilon \cdot \log d))$ in time $2^{a(G)} = \mathrm{poly}(mn/\epsilon \cdot \log d)$ such that G has quality $1 - \epsilon$ against every class $F(W) \subseteq F(\mathbf{m}, d)$.*

Similar to the generator in Proposition 9, the above generator first reduces the multivariate polynomial to a univariate one, and then it applies a generator against the resulting univariate polynomial. Here this univariate generator is obtained from Proposition 10. In contrast to the trivial generator, which was used in Proposition 9, this generator has no dependence on the degree of the polynomial. Another difference to Proposition 9 is that the construction time depends exponentially on the prime number which is used to reduce the degrees in the substitution. For Proposition 11,

we pick this prime number at random, which allows us to choose it from an exponentially smaller range with respect to d. The formal proofs of Proposition 11 and Theorem 3 are deferred to Appendix A.1. The main technical ingredient for the proof of Proposition 11 is an improved version of a reduction in [KS01]. The proof of Theorem 3 follows our general framework and uses the previous two propositions as building blocks.

Acknowledgments

We would like to thank Luca Trevisan and Avi Wigderson for helpful remarks on this work.

References

[AB03] Manindra Agrawal and Somenath Biswas. Primality and identity testing via chinese remaindering. *J. ACM*, 50(4):429–443, 2003.

[Agr05] Manindra Agrawal. Proving lower bounds via pseudo-random generators. In R. Ramanujam and Sandeep Sen, editors, *FSTTCS*, volume 3821 of *Lecture Notes in Computer Science*, pages 92–105. Springer, 2005.

[AKS04] Manindra Agrawal, Neeraj Kayal, and Nitin Saxena. PRIMES is in P. *Ann. of Math. (2)*, 160(2):781–793, 2004.

[ALM+98] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *J. ACM*, 45(3):501–555, 1998.

[Alo99] Noga Alon. Combinatorial Nullstellensatz. *Comb. Probab. Comput.*, 8(1-2):7–29, 1999.

[AS98] Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: a new characterization of NP. *J. ACM*, 45(1):70–122, 1998.

[AT92] Noga Alon and Michael Tarsi. Colorings and orientations of graphs. *Combinatorica*, 12(2):125–134, 1992.

[BCW80] Manuel Blum, Ashok K. Chandra, and Mark N. Wegman. Equivalence of free boolean graphs can be decided probabilistically in polynomial time. *Information Processing Letters*, 10:80–82, 1980.

[BK95] Manuel Blum and Sampath Kannan. Designing programs that check their work. *J. ACM*, 42(1):269–291, 1995.

[Bog05] Andrej Bogdanov. Pseudorandom generators for low degree polynomials. In Harold N. Gabow and Ronald Fagin, editors, *STOC*, pages 21–30. ACM, 2005.

[CK00] Zhi-Zhong Chen and Ming-Yang Kao. Reducing randomness via irrational numbers. *SIAM J. Comput.*, 29(4):1247–1256, 2000.

[CRS95] Suresh Chari, Pankaj Rohatgi, and Aravind Srinivasan. Randomness-optimal unique element isolation with applications to perfect matching and related problems. *SIAM J. Comput.*, 24(5):1036–1050, 1995.

[DL78] Richard A. DeMillo and Richard J. Lipton. A probabilistic remark on algebraic program testing. *Information Processing Letters*, 7, 1978.

[DS07] Zeev Dvir and Amir Shpilka. Locally decodable codes with two queries and polynomial identity testing for depth 3 circuits. *SIAM Journal on Computing*, 36(5):1404–1434, 2007.

[IM83] Oscar H. Ibarra and Shlomo Moran. Probabilistic algorithms for deciding equivalence of straight-line programs. *J. ACM*, 30(1):217–228, 1983.

[KI04] Valentine Kabanets and Russell Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. *Comput. Complex.*, 13(1/2):1–46, 2004.

[KS01] Adam Klivans and Daniel A. Spielman. Randomness efficient identity testing of multivariate polynomials. In *ACM Symposium on Theory of Computing*, pages 216–223, 2001.

[KS06] Neeraj Kayal and Nitin Saxena. Polynomial identity testing for depth 3 circuits. In *IEEE Conference on Computational Complexity*, pages 9–17, 2006.

[LFKN92] Carsten Lund, Lance Fortnow, Howard Karloff, and Noam Nisan. Algebraic methods for interactive proof systems. *J. ACM*, 39(4):859–868, 1992.

[Lov79] László Lovász. On determinants, matchings, and random algorithms. In *FCT*, pages 565–574, 1979.

[LV98] Daniel Lewin and Salil Vadhan. Checking polynomial identities over any field: Towards a derandomization? In *ACM Symposium on Theory of Computing*, pages 438–437, 1998.

[LV03] Richard Lipton and Nisheeth Vishnoi. Deterministic identity testing for multivariate polynomials. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 756–760, 2003.

[MVV87] Ketan Mulmuley, Umesh V. Vazirani, and Vijay V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7(1):105–113, 1987.

[Sch80] Jacob Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *Journal of the ACM*, 27:701–717, 1980.

[Sha92] Adi Shamir. IP = PSPACE. *J. ACM*, 39(4):869–877, 1992.

[Shp07] Amir Shpilka. Interpolation of depth-3 arithmetic circuits with two multiplication gates. In *STOC '07: Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 284–293, New York, NY, USA, 2007. ACM Press.

[Zip79] Richard Zippel. Probabilistic algorithms for sparse polynomials. In *International Symposium on Symbolic and Algebraic Computation*, volume 72 of *LNCS*, pages 216–226, Berlin, 1979. Springer-Verlag.

A Addendum to Section 4

Remark 2 In order to prove Proposition 9, we introduced the class of polynomials $F(w, W)$ containing all polynomials in the span of W in which w has a nonzero coefficient. Instead of $F(w, W)$ a more natural class of polynomials to consider in that proposition is $F(W)$, the class of all nonzero polynomials in the span of W. However, the techniques from [KS01] only allow to construct a hitting set generator against $F(W)$ with seed length $2\log m + O(\log nd/\epsilon)$. The reason is that the substitution used for $F(W)$ has to map *every* two distinct monomials of W to a pair of distinct univariate monomials, and so the resulting univariate polynomial has at least *quadratic* degree in m, which requires the seed length to be at least $2\log m$. It follows that every construction based on these hitting set generators for $F(W)$ has seed length at least twice the lower bound $\log m$. In contrast to this, the substitution for $F(w, W)$ used above gives a *linear* degree in m and hence a seed length of only about $\log m + O(\log nd/\epsilon)$.

A.1 Proofs Deferred from Section 4.1

We note the following fact.

Fact 12 *If an advised generator G has quality α against every class in a set $\mathcal{F}$ of classes of polynomials, then the advice-less generator $\bar{G}$ has density α against the union $\bigcup_{F\in\mathcal{F}} F$.*

The fact implies that, in order to construct a generator against a class of polynomials F^*, it is sufficient to construct an advised generator against a *cover* $\mathcal{F}$ of F^*, i.e., $\mathcal{F}$ satisfies $\bigcup_{F\in\mathcal{F}} F \supseteq F^*$.

Proof (of Theorem 2). Let $k = \lceil \sqrt{\log M/\log d} \rceil$ and $\epsilon = 1/2k$. Define the weight of variable x_j as $w(j) = \log m_j$. Let $S_1, \ldots, S_k$ be a partition of the coordinates $[n]$ as in Lemma 4 (Balanced Factors). For each $i \in [k]$, we construct an advised generator G_i of quality $1-\epsilon$ against every class $F(w_i, W_i) \subseteq F(M_i, d)$, where $M_i = \prod_{j\in S_i} m_j$. If $|S_i| = 1$, we can obtain G_i from Proposition 5. In this case, $r(G_i) = O(\log d/\epsilon)$ and $a(G_i) = 0$. If $|S_i| > 1$, we obtain G_i from Proposition 9. In that case, the seed length is $r(G_i) = \log M_i + O(\log nd/\epsilon)$ and the advice length is $a(G_i) = O(r(G_i))$. Let $G = \bigotimes_{i\in[k]} G_i$ be the shared advice product. The seed length of G is

$$r(G) = \textstyle\sum_i r(G_i) \le \sum_i \log(M_i) + O(k \log nd/\epsilon) = \log M + O\tilde{}(\sqrt{\log M \cdot \log d}).$$

Since, by Lemma 4, $\log M_i \le \frac{4}{k}\log M$ for all $i \in [k]$ with $|S_i| > 1$, the advice length of G is

$$a(G) = \max_i a(G_i) \le O\big(\tfrac{1}{k}\log M + \log(knd)\big) = O(\sqrt{\log M \cdot \log d}).$$

By Fact 12, it is sufficient to show that G has quality $1/2$ against a cover of $F(\mathbf{m}, d)$. Then, we can conclude that the advice-less generator $\bar{G}$ obtained from G has density $1/2$ against $F(\mathbf{m}, d)$. Also, $r(\bar{G}) = r(G) + a(G) = \log M + O\tilde{}(\sqrt{\log M \cdot \log d})$. By Lemma 3, G has quality $1 - k\epsilon = 1/2$ against every class of the form $\prod_{i\in[k]} F(w_i, W_i)$ with $F(w_i, W_i) \subseteq F(M_i, d)$. We claim that these products cover the class $F(\mathbf{m}, d)$. Consider any polynomial f in $F(\mathbf{m}, d)$. Let $\mathbf{x}_i$ denote the restriction of $\mathbf{x}$ to the coordinates in S_i. Let w be a monomial with nonzero coefficient in f. Write $w = \prod_{i\in[k]} w_i$ with $w \in K[\mathbf{x}_i]$. Let $W_i \subseteq K[\mathbf{x}_i]$ be the set of monomials with non-zero coefficient in f when written as polynomial in the variables $\mathbf{x}_i$. Note that $w_i \in W_i$ and $F(w_i, W_i) \subseteq F(M_i, d)$. Now we have $f \in \prod_{i\in[k]} F(w_i, W_i)$. In this way, we can demonstrate for every $f \in F(\mathbf{m}, d)$ that it is contained in one of the products $\prod_{i\in[k]} F(w_i, W_i)$. So these products indeed cover the class $F(\mathbf{m}, d)$. By Fact 12, it follows that $\bar{G}$ has density $1/2$ against $F(\mathbf{m}, d)$. □

Proof (of Proposition 11). Let $t = 4 \cdot 2m^2 n/\epsilon$. The generator G uses the advice string $y = y_1 y_2$ in order to compute a number $p = p(y_1)$ of magnitude at most $t \cdot 10 \log^2 dt$ such that p is a random prime with high probability over a uniform choice of y. More precisely, $\Pr_y[p \text{ is prime}] \geq 1 - \epsilon/4$ and we additionally require that p is uniformly distributed over the primes in this range. Further, G computes a number $k = k(y_2)$ from the range $[t]$ so that for all $\tau \in [t]$, we have $\Pr_y[k = \tau] = 1/t$.

Then G uses the seed z in order to choose a random number $b = b(z)$ from the range $[\lceil 2m/\epsilon \rceil]$. Finally, G outputs the point $(b^{\lfloor k^{i-1} \rfloor_p})_{i \in [n]}$. Note that G can be constructed in time polynomial in the magnitude of p, which is exponential in the length of the advice of G.

Let $F(W) \subseteq F(m, d)$. It remains to show that G has quality $1 - \epsilon$ against $F(W)$. Consider the substitution σ from $K[\mathbf{x}]$ to $K[X]$ with $\sigma(x_i) = X^{\lfloor k^{i-1} \rfloor_p}$ for $i \in [k]$. Notice, σ depends only on the advice string for G. We say that an advice string is W-good if $\sigma(w) \neq \sigma(w')$ for every pair of distinct monomials $w, w' \in W$. An improvement over Lemma 2 in [KS01] which is given in Lemma 13 shows that a random advice string is W-good with probability at least $1 - \epsilon/2$.

Now given this event occurs, we argue the generator $G(y, \cdot)$ has density at least $1 - \epsilon/2$. This is because in this case σ maps every member of $F(W)$ to a nonzero polynomial in $K[X]$ with at most m monomials and further the range of b has cardinality at least $2m/\epsilon$ (cf. Proposition 10).□

Proof (of Theorem 3). Given $\delta > 0$, $\mathbf{m} = (m_1, \ldots, m_n)$, and $d \in \mathbb{N}$, we show how to construct a generator G of density $1/2$ against $F(\mathbf{m}, d)$. Let $k = \lceil \delta \log M / \log \log M \rceil$ and $\epsilon = 1/2k$.

Define the weight of variable x_i to be $w(i) = \log m_j$. Let $S_1, \ldots, S_k$ be the partition of the set of variables obtained from Lemma 4. Let $M_i = 2^{w(S_i)} = \prod_{j \in S_i} m_j$. For every $i \in [k]$ with $|\mathbf{x}_i| = 1$, let G_i be a trivial generator with seed length $\log(2M_i/\epsilon)$ as in Proposition 10. For every $i \in [k]$ with $|\mathbf{x}_i| > 1$, let G_i be a generator as in Proposition 11 of quality $1 - \epsilon$ against every class $F(W_i) \subseteq F(M_i, d)$.

We claim that $G = \bigotimes_i G_i$ is a generator against a cover of $F(\mathbf{m}, d)$ as desired. The seed length of G is $r(G) = \sum_i r(G_i) \leq \sum_i \log M_i + O(k \log 1/\epsilon) = \log M + O(\delta \log M)$. Since $\log M_i \leq \frac{4}{k} \log M$ for all $i \in [k]$ with $|\mathbf{x}_i| > 1$, the advice of G is $a(G) = \max_i a(G_i) \leq O(\frac{1}{k} \log M + \log(n/\epsilon \cdot \log d)) = O(\delta^{-1} \log \log M + \log(n \log d))$. Furthermore, we can construct G in time $k \cdot 2^{a(G)} = \text{poly}(\log^{1/\delta} M, n \log d)$.

By Lemma 3, G has quality $1 - \epsilon$ against every class of the form $\prod_i F(W_i)$ with $F(W_i) \subseteq F(M_i, d)$. It follows that the advice-less generator $\bar{G}$ obtained from G has density $1/2$ against $F(\mathbf{m}, d)$ (Fact 12). □

A.2 An improved reduction from multivariate to univariate

In this section we improve an identity-preserving reduction from multivariate polynomials to univariate polynomials due to Klivans and Spielman [KS01] with regards to randomness. The savings in randomness are exponential in the total degree d. This is why this improvement might be of independent interest. We used it in the proof of Proposition 11.

Lemma 13 *Let $\epsilon > 0$ and $W \subseteq K[x_0, \ldots, x_{n-1}]$ be a set of m monomials, each of total degree at most d. For positive integers k and p, let $t = 2m^2 n/\epsilon$ and $\sigma = \sigma_{k,p}$ be the substitution from $K[\mathbf{x}]$ to $K[X]$ such that $\sigma(x_i) = X^{\lfloor k^i \rfloor_p}$. If k is picked uniformly at random from $[t]$ and p is picked uniformly at random from the primes in $\{1, \ldots, t \cdot 10 \log^2 dt\}$, then*

$$\Pr_{k,p} (\forall w, w' \in W.\ \sigma(w) = \sigma(w') \implies w = w') \geq 1 - \epsilon.$$

Proof. Suppose $W = \{\mathbf{x}^{\mathbf{a}_1}, \ldots, \mathbf{x}^{\mathbf{a}_m}\}$, where $\mathbf{x}^{\mathbf{a}_i}$ denotes the monomial $\prod_{j=0}^{n-1} x_j^{a_{ij}}$. For all distinct $i, i' \in [m]$, we have $\Pr_{k\in[t]}(\sum_j a_{ij}k^j = \sum_j a_{i'j}k^j) \leq n/t$, because the non-zero polynomial $\sum_j (a_{ij} - a_{i'j})k^j$ has at most n roots and there are t choices for k. For every $k \in [t]$, the number $\sum_j (a_{ij} - a_{i'j})k^j$ has at most $\log(2dt^n) \leq n \log dt$ distinct prime factors, because $|\sum_j (a_{ij} - a_{i'j})k^j| \leq \sum_j (a_{ij} + a_{i'j})t^n \leq 2dt^n$. The interval $\{1, \ldots, t \cdot 10 \log^2 dt\}$ contains at least $t \log dt$ prime numbers (for t large enough). Hence,

$$\Pr_{k,p}(\sigma(\mathbf{x}^{\mathbf{a}_i}) = \sigma(\mathbf{x}^{\mathbf{a}_{i'}})) \leq \Pr_{k,p}(\textstyle\sum_j a_{ij}k^j = \sum_j a_{i'j}k^j \mod p) \leq \frac{n}{t} + \frac{n \log dt}{t \log dt} \leq 2n/t = \epsilon/m^2.$$

By the union bound, the probability that none of the events $\sigma(w) = \sigma(w')$ happen for distinct $w, w' \in W$ is at least $1 - \binom{m}{2} \cdot \epsilon/m^2 \geq 1 - \epsilon$. □

B Optimality of the Lower Bound

The dimension based lower bound we discussed in the introduction is easily seen to be optimal in the following sense. Suppose that $F \cup \{0\}$ is indeed a linear subspace of $K[x_1, \ldots, x_n]$. Let $H \subseteq K^n$ be a *minimal* hitting set of positive density against F (if one exists). Then for every point $\mathbf{x} \in H$, there is a polynomial $f_{\mathbf{x}}$ in F such that $f_{\mathbf{x}}(\mathbf{x}) \neq 0$ but $f_{\mathbf{x}}(\mathbf{x}') = 0$ for all other points $\mathbf{x}' \in H$. It follows that the space spanned by the set of polynomials $\{f_{\mathbf{x}}\}_{x\in H}$ has dimension $|H|$. Thus, $F \cup \{0\}$ has dimension at least $|H|$. (And by the lower bound, $F \cup \{0\}$ has dimension exactly $|H|$.)

Also the following is true. If $F \cup \{0\}$ is a subspace of dimension h, then h randomly chosen points from a large enough range $[R]^n$ form a hitting set against F with high probability.

C Generating a prime number efficiently

An n-bit prime number can be generated using $2n$ random bits. The idea is to test n^2 pairwise independent numbers deterministically for primality [AKS04]. The Chebyshev bound gives us a success probability of $1 - o(1)$. Using probability amplification on expander graphs, the error probability can be brought down to any $\epsilon > 0$ using an additional $\log(1/\epsilon)$ random bits and time polynomial in $\log(1/\epsilon)$.

In the context of Section 3.1, we were interested in an algorithm with cubic runtime. There currently is no deterministic primality test with this runtime, but the randomized Rabin-Miller test can be used instead. This will result in $O(n + \log(1/\epsilon))$ random bits.

Remark 3 (on Proposition 7) If we could efficiently generate an N-bit prime number with $o(N)$ random bits, then the advice length in Proposition 7 could be reduced accordingly to $o(\log D)$ bits. This would directly imply an asymptotically optimal and efficient construction. But, any method to compute an N-bit prime number in time $\text{poly}(N)$ that we are aware of requires $\Omega(N)$ random bits. Computing an N-bit prime number efficiently with $o(N)$ random bits (or no random bits at all) is an intriguing open problem. Cramer's conjecture about prime gaps states that the gap between two consecutive N-bit prime numbers is polynomially bounded in N. Hence, it implies a deterministic polynomial time algorithm for finding an N-bit prime number. However, mathematical research seems far from proving such a result. Even if we assume the Generalized Riemann Hypothesis, the gaps between N-bit primes are only known to be bounded by $2^{N/2} \cdot \text{poly}(N)$. And even if this were a density result, it would only imply an algorithm using $N/2$ random bits.