Asymptotically Optimal Hitting Sets Against Polynomials

Markus Bläser[1], Moritz Hardt[2], and David Steurer[2]

[1] Saarland University, Saarbrücken, Germany
[2] Princeton University, Princeton, NJ

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 $\log D + \tilde{O}(\log^{1/2} D)$. Here, $\log D = \sum_i \log(d_i + 1)$ is a lower 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 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.
As our main corollary, we obtain the first blackbox polynomial identity tests with an asymptotically optimal randomness consumption.

1 Introduction

Consider a class of polynomials F 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. To give an example of a hitting set, 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 $d_i + 1$ assuming $|K| > d_i$, then the set $H = S_1 \times \cdots \times S_n$ is a hitting set against F of size $D = \prod_i (d_i + 1)$ (see, for instance, [1]). It is easy to argue that the *size* of this hitting set is optimal. But even for $d_i = 1$ the set is so large that we would like to have an efficient implicit representation of it. This would typically be a be a function called *hitting set generator* computable by a small circuit on $\log |S|$ inputs that serve the purpose of a random *seed*. A second observation is that there are polynomials in F vanishing on all except a single point in H. Here, it would be more desirable if the non-roots of any polynomial in F had *high density* in H. This requirement is met by the well-known *Schwartz-Zippel Lemma* [2–4]: If we replace each S_i by a set of $2nd_i$ points (rather than $d_i + 1$ points), then any polynomial in F vanishes on at most half of the points in H. However, the size of H increased to $(2n)^n \cdot \prod d_i$. Even in terms of $\log |S|$ this increase in size is only of lower order for large enough degree d_i. It

is natural to ask whether this increase in size is inherent. It turns out the answer is negative. In this paper, we present efficient constructions of hitting sets for which the quantity $\log |S|$ is asymptotically optimal, but at the same time our hitting sets will have high density in the above sense.

The main motivation for our work is the closely related problem 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 a member of some class $F \subseteq K[x_1, \ldots, x_n]$. Provided with a hitting set generator 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 given that the hitting set has high density. Notice, these steps only require *blackbox access* to the polynomial. That is, we need not make any structural assumption about the input representation of the polynomial.

The study of polynomial identity testing was initiated by the work of DeMillo, Lipton, Schwartz and Zippel [2–4]. Many interesting problems have since turned out to reduce to checking polynomial identities [5–11]. Similarly, several results in complexity theory [12–15] 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: One is giving deterministic identity tests for restricted classes of arithmetic circuits [16–19]. As it turned out, testing general arithmetic circuits for identity in even subexponential deterministic time is linked to circuit lower bounds [20, 16]. The other approach has been to minimize the seed length of hitting set generators against more general classes of polynomials [21–24]. In this work we continue the study of the latter problem. In this case, there is a natural lower bound on the number of random bits required that we are trying to match:

Suppose a class of polynomials $F \subseteq K[x_1, \ldots, x_n]$ contains a linear space $W \subseteq F \cup \{0\}$ of dimension at least d. Then, a dimension argument [22] shows that any hitting set generator of density $1 - \epsilon$ against F requires seed length at least $r \geq \log(d/\epsilon)$. Here, we appealed to the following definition.

Definition 1. *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.*

We are interested in uniform constructions of hitting set generators. That is, we will consider classes of polynomials $F(t)$ given by a parameter t and we want to have a construction algorithm which on input of t and $\epsilon > 0$ constructs a circuit that computes a hitting set generator G of density $1 - \epsilon$ against $F(t)$. The running time of this algorithm will be measured in terms of the description length $|t|$; the runtime also serves as an upper bound on the size of the circuit.

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. Crucial primitives 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.

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

Theorem 1. *Given a degree $\boldsymbol{d}$, we can efficiently construct a hitting set generator G of density $1/2$ against $F(\boldsymbol{d})$ over any field of characteristic zero such that the seed length of G is $\log D + O(\sqrt{\log D} \cdot \log\log D)$.*

Since the quantity D is the dimension of the space $F(\boldsymbol{d}) \cup \{0\}$, the dimension lower bound implies that the seed length is asymptotically optimal for the entire family of parameters $d_1, \ldots, d_n$ where $n, d_i \in \mathbb{N}$. Our result also holds over large enough finite fields. Here, the requirement on the size of the field is roughly the same as in 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.

We also show how to obtain a nearly linear time construction at the cost of slightly more but still asymptotically optimal seed length. More generally we have the trade-off between runtime $O(\log^{1+\delta} D)$ and seed length $\log D + O(\log^{1-\delta} D \cdot \log\log D)$ where $\delta \in (0, 1/2)$. A similar trade-off holds for the required size of finite fields.

Sparse Polynomials. 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* $\boldsymbol{m} = (m_1, \ldots, m_n)$, if f has at most m_i nonzero terms when written as a univariate polynomial in the i-th variable. For a tuple $\boldsymbol{m} = (m_1, \ldots, m_n)$ and an integer $d \in \mathbb{N}$, we define $F(\boldsymbol{m}, d)$ as the class of nonzero sparsity-$\boldsymbol{m}$ polynomials of total degree at most d. Henceforth, let $M = \prod_{i=1}^{n} m_i$.

Theorem 2. *Given sparsity $\boldsymbol{m}$ and degree $d \leq M$, we can efficiently construct a hitting set generator G of density $1/2$ against $F(\boldsymbol{m}, d)$ over any large enough finite field, say, $|K| \geq \mathrm{poly}(Mnd)$, such that the seed length of G is $\log M + O(\sqrt{\log M \cdot \log d} \cdot \log\log M)$.*

The lower bound shows that any hitting set generator of positive density against $F(\boldsymbol{m}, d)$ 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 $\delta > 0$, $\boldsymbol{m}$, and d, we can construct in time polynomial in $n \log d \cdot \log^{1/\delta} M$ a hitting set generator G of density $1/2$ against $F(\boldsymbol{m}, d)$ over any field of characteristic zero such that 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 $F(\boldsymbol{m}, d)$, 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 [23].

1.2 Previous Work

The Schwartz-Zippel Lemma gives a generator against $F(\boldsymbol{d})$ 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 [24] 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 [21] 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. Lewin and Vadhan [22] 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 constructions 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 [10] 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 [4, 25]. However, a particular tool introduced in their work turns out to give us hitting set generators of the optimal seed length $\log D$ over finite fields. This tool will be used and discussed later. We will see

how to achieve asymptotically the same seed length over significantly smaller finite fields (that is, $|K| > D^{o(1)}$ as opposed to $|K| > D$).

When it comes to sparse polynomials, Klivans and Spielman [23] construct a hitting set generator of seed length $O(\log(mnd))$ against the class of n-variate polynomials of total degree d and at most m nonzero terms. This is better than previous work when $\log m = o(n \log d)$. Although we use techniques from this work, our results are strictly speaking incomparable to those of Klivans and Spielman, since we consider a different class of "sparse" polynomials. However, 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, we may assume $M \geq 2^n$, since all variables with $m_i = 1$ can be fixed to an arbitrary nonzero constant.

Below 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 (neglecting constants and polylogarithmic factors). The density is fixed to be a constant, say, $1/2$. We put $q = \log |K|$ when K is finite.

Size/D	Runtime		Source
	$\mathrm{char}(K) = 0$	$\mathrm{char}(K) > 0$	
n^n	$\log D$	$\log D$	Schwartz-Zippel [4, 3, 2]
2^n	$\mathrm{poly}(nd)$	$\mathrm{poly}(dq)$	Chen-Kao, Lewin-Vadhan [21, 22]
1	$2^{O(\log D)}$	$\mathrm{poly}(q \log D)$ for $\lvert K\rvert \geq D$	Kronecker substitution [10]
$D^{1/\log^{1/2} D}$	$\mathrm{poly}(\log D)$	—	This work, Thm. 1
$D^{o(1)}$	$\log D$	$\mathrm{poly}(q \log D)$ for $\lvert K\rvert \geq D^{o(1)}$	This work, cf. Thm. 5
Size/M			
$d \cdot M^c$	$\mathrm{poly}(\log M \cdot \log d)$	$\mathrm{poly}(q \log M)$	Klivans-Spielman [23]
$\log d \cdot M^\delta$	$\mathrm{poly}(\log^{1/\delta} M \cdot \log d)$	—	This work, Thm. 2
$d \cdot M^{\frac{\log^{1/2} d}{\log^{1/2} M}}$	—	$\mathrm{poly}(q \cdot \log M)$	This work, Thm. 3

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 2 (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_1F_2 against which the direct product still has high density.

Definition 3 (Schwartz-Zippel product). *Let $F_1 \subseteq K[\boldsymbol{x}_1]$ and $F_2 \subseteq K[\boldsymbol{x}_2]$ be two classes of polynomials on disjoint sets of variables $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, respectively. Let $n_i = |\boldsymbol{x}_i|$. We define the* Schwartz-Zippel product *F_1F_2 to be the set of polynomials $f \in K[\boldsymbol{x}_1, \boldsymbol{x}_2]$ such that f as a polynomial in $\boldsymbol{x}_2$ has a coefficient $g \in K[\boldsymbol{x}_1]$ satisfying the following two properties: (1) g is a member of F_1, and (2) for every $\boldsymbol{a}_1 \in K^{n_1}$ with $g(\boldsymbol{a}_1) \neq 0 \in K$, the polynomial $f(\boldsymbol{a}_1, \boldsymbol{x}_2) \in K[\boldsymbol{x}_2]$ is a member of F_2.*

Intuitively, this is the same product structure required in the well-known proof of the Schwartz-Zippel Lemma. As desired, the next lemma is an immediate consequence of the definition.

Lemma 1. *Let G_1 and G_2 be two generators, and let $F_1 \subseteq K[\boldsymbol{x}_1]$ and $F_2 \subseteq K[\boldsymbol{x}_2]$ 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_1F_2.*

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

Definition 4 (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,*

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

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

Definition 5 (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 $G_1 \otimes G_2\ (y, z_1z_2) = (G_1(y,z_1), G_2(y,z_2))$. 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 2. *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, each working 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 $\bar{G}$.

A large number of factors k decreases the *relative* amount of advice. On the other hand, the quality of G suffers as k grows. Varying over k gives rise to interesting trade-offs. But once we fix k we want to determine a partition that minimizes the seed length of our construction.

So, suppose we can associate a weight with each variable such that the total weight of a set of 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 3. *Given a positive integer k and a polynomial ring $K[\boldsymbol{x}]$ with non-negative 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. □

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 1. *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 also need the Kronecker substitution as introduced by [10] for our parameters.

Lemma 4. *Let* $\boldsymbol{d} = (d_1, \ldots, d_n) \in \mathbb{N}^n$ *and define the* Kronecker substitution *as* $\mathbf{kr}(X) = (X^{D_1}, \ldots, X^{D_n})$, *where* $D_i = \prod_{j<i}(d_j + 1)$. *Then, for every* $f \in F(\boldsymbol{d})$, *we have that* $f' = f(\mathbf{kr}(X)) \in K[X]$ *is a univariate polynomial of degree at most* $D - 1$ *such that any two distinct monomials* w *and* w' *in* f *map to distinct monomials in* f'. *In particular,* f' *is not identically zero in* $K[X]$.

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 the previous lemma with Proposition 1. 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.

Over fields of characteristic zero the bit size of $\mathbf{kr}(s)$ is at least D which is *exponential* in the desired runtime of our algorithm. It turns out, we can reduce the points of the hitting set modulo a $(\log D)$-bit prime number. However, this step seems to require at least $\log D$ additional random bits. Indeed, any method of computing an N-bit prime number in time $\mathrm{poly}(N)$ 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 would imply such an algorithm. However, even if we assume the Generalized Riemann Hypothesis, the gaps between N-bit primes are only known to be bounded by $2^{N/2} \cdot \mathrm{poly}(N)$. And even if this were a density result, it would only imply an algorithm using $N/2$ random bits.

Surprisingly, we can circumvent this problem by modeling the additional $O(\log D)$ random bits as random advice. This way, we can exploit our framework in order to reduce the random advice to $o(\log D)$ bits and thus achieve an asymptotically optimal result.

Proposition 2. *Let* K *be of characteristic zero. For any degree* $\boldsymbol{d}$ *and any* $\epsilon > 0$ *we can construct a hitting set generator* G *of quality* $1 - \epsilon$ *against* $F(\boldsymbol{d})$ *in time polynomial in* $\log(D/\epsilon)$. *Furthermore,* $r(G) = \log(D/\epsilon) + O(1)$ *and* $a(G) = O(\log(D/\epsilon))$.

Proof (Sketch). 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 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 $\boldsymbol{b}$ which is obtained by reducing $\mathbf{kr}(s)$ component-wise modulo p. We claim whenever $p(y)$ is a prime number, then $G(y, \cdot)$ has density $1 - \epsilon$ against $F(\boldsymbol{d})$. This can be shown by arguing since f is nonzero, we have that $f(\mathbf{kr}(X))$ vanishes on at most $D - 1$ in R points modulo p. The contrapositive of this argument follows from a standard argument involving a Vandermonde matrix modulo p in which we observe that $f(\mathbf{kr}(s)) = f(\boldsymbol{b}) \mod p$. □

We proceed to prove a more general version of Theorem 1.

Theorem 5. *Let $\boldsymbol{d} = (d_1, \ldots, d_n)$ and $\epsilon > 0$. Then, for any $k \in \{1, \ldots, n\}$, we can efficiently construct a hitting set generator of density $1-\epsilon$ against $F(\boldsymbol{d})$ and seed length $\log(D/\epsilon) + O(k \log(k/\epsilon)) + O(\log(D/\epsilon)/k)$. The construction works for any field of characteristic zero and any finite field of size at least $\frac{2}{\epsilon} \cdot k \cdot D^{4/k}$.*

Proof. Define the weight of the variable x_i as $w(i) = \log(d_i + 1)$. Apply Balanced Factors (Lemma 3) with the given choice of k so as to obtain a partition of the coordinates $[n]$ into sets $S_1, \ldots, S_k$. Let $\boldsymbol{d}_i$ denote the restriction of $\boldsymbol{d}$ to the coordinates in S_i. For each $i \in [k]$ we will construct an advised generator G_i against $F(\boldsymbol{d}_i)$ of quality $1 - \epsilon/2k$. If $|S_i| = 1$, then we obtain G_i from Proposition 1. In this case $a(G_i) = 0$. Whenever $|S_i| > 1$, we obtain G_i from Proposition 2 in case K is of characteristic zero. Consider the advised generator $G = \bigotimes_{i \in [k]} G_i$. This is a generator against the Schwartz-Zippel product $\prod_{i \in [k]} F(\boldsymbol{d}_i)$ which is a superset of $F(\boldsymbol{d})$. Its quality follows from Lemma 2. 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 $\log(D_i) = w(S_i) \leq 4w(S)/k = 4 \log D/k$. Therefore, we obtain the desired generator by combining seed and advice of G (see Fact 4). If K is a finite field, we obtain the above G_i directly via 1. The required field size is $\max_i 2kD_i/\epsilon \leq 2kD^{4/k}/\epsilon$. □

For $k = \lceil \sqrt{\log D / \log(1/\epsilon)} \rceil$ we obtain Theorem 1.

Nearly Linear Time. The larger we choose k the more efficient is our construction. Notice the trivial generators from Proposition 1 can be constructed in time linear in their seed length. But to construct a generator from Proposition 2 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 $\tilde{O}(N)$. 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 $N + \tilde{O}(N^{1-\delta})$. It is easy to see that the exponent c need not be larger than 2. The prime number required in the proof of Lemma 4 can be computed once in cubic time (e.g., using the Rabin-Miller primality test) and passed on to all generators. Provided with this prime number, each generator can be constructed in quadratic time.

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 $F(\boldsymbol{m}, d)$ with asymptotically optimal seed length, provided $\log d$ is sufficiently smaller than $\log M$. In the previous section, our basic building blocks were generators against the target class $F(\boldsymbol{d})$ that have optimal seed length, but require some amount of advice. For the target class $F(\boldsymbol{m}, d)$, 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 $F(w, W) \subseteq F(\boldsymbol{m}, d)$. 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 $f \in K[\boldsymbol{x}]$ 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 3. *Given $\boldsymbol{m}$, 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 $F(w, W) \subseteq F(\boldsymbol{m}, d)$.*

The proposition crucially relies on a multivariate to univariate reduction introduced by Klivans and Spielman [23]. This reduction maps a point $b \in K$ to the tuple $(b^{\lfloor k^{i-1} \rfloor_p})_{i \in [n]}$ where p is prime, k is a random number and $\lfloor k^{i-1} \rfloor_p$ denotes the remainder of k^{i-1} modulo p. In our case, p and k will be generated independently using the advice string (so that the substitution depends on the advice). The point b is simply drawn from a large enough range using the seed. Intuitively, the proposition then 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 $F(w, W)$. Precisely, this will happen if the reduction induced by the advice string is "isolating" with respect to w and W. That is, no distinct monomial w' collides with w under the given reduction. But Klivans and Spielman showed that this isolation behavior occurs with high probability.

We remark that that possibly no single advice string above yields a generator that is dense against $F(\boldsymbol{m}, d)$.

Using Proposition 3 as our basic building block, our construction against $F(\boldsymbol{m}, d)$ essentially works as follows. First, we compute a *balanced partition* $(S_1, \ldots, S_k)$ of the coordinates $[n]$ (Lemma 3). 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 $F(\boldsymbol{m}_i, d)$, where $\boldsymbol{m}_i$ is the restriction of $\boldsymbol{m}$ 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 $F(\boldsymbol{m}, d)$. By Lemma 2, G has high quality against any product $\prod_i F(w_i, W_i)$ with $F(w_i, W_i) \subseteq F(\boldsymbol{m}_i, d)$. This implies that $\bar{G}$ has high density against the union of all such products. Finally, $\bar{G}$ has high density

against $F(\boldsymbol{m}, d)$, because every polynomial in $F(\boldsymbol{m}, d)$ is contained in one of the products $\prod_i F(w_i, W_i)$.

The details of the proof of Proposition 3 and Theorem 3 follow along the lines of our discussion and are omitted from this extended abstract. They will appear in the full version of the paper.

Over Fields of Characteristic Zero. Let K be a field of characteristic zero. Lipton and Vishnoi [26] point out the fact that a univariate polynomial with at most m nonzero terms has at most m positive rational roots over K (a consequence of Descartes' Rule of Signs).

Proposition 4. *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.*

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

Proposition 5. *Given $\epsilon > 0$, $\boldsymbol{m}$, 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 $F(W) \subseteq F(\boldsymbol{m}, d)$.*

As in Proposition 3, the above generator first reduces the multivariate polynomial to a univariate one using the same substitution. Then it applies the generator from Proposition 4 against the resulting univariate polynomial. In contrast to the trivial generator, which was used in Proposition 3, this generator has no dependence on the degree of the polynomial. Another difference to Proposition 3 is that the construction time depends polynomially on the magnitude of the prime p number which is used to reduce the degrees in the substitution $(b^{\lfloor k^{i-1} \rfloor_p})_{i \in [n]}$. This is because over characteristic zero, the magnitude of the points blows up exponentially. In the case of sparse polynomials we do not know how to reduce the bit size of the points as we did in Proposition 2.

The doubly logarithmic dependence on d in the advice length is achieved by analyzing the effect of using a *uniformly random* prime in the substitution. This analysis improves the one given in [23] exponentially with respect to d. It is the main technical ingredient for the proof of Proposition 5. The proof of Theorem 3 follows our general framework and uses the previous two propositions as building blocks. Both proofs are omitted from this extended abstract.

References

1. Alon, N.: Combinatorial Nullstellensatz. Comb. Probab. Comput. **8**(1-2) (1999) 7–29
2. DeMillo, R.A., Lipton, R.J.: A probabilistic remark on algebraic program testing. IPL **7** (1978)
3. Zippel, R.: Probabilistic algorithms for sparse polynomials. In: Proc. ISSAC, Berlin, Springer-Verlag (1979) 216–226

4. Schwartz, J.: Fast probabilistic algorithms for verification of polynomial identities. Journal of the ACM **27** (1980) 701–717
5. Chari, S., Rohatgi, P., Srinivasan, A.: Randomness-optimal unique element isolation with applications to perfect matching and related problems. SIAM J. Comput. **24**(5) (1995) 1036–1050
6. Lovász, L.: On determinants, matchings, and random algorithms. In: FCT. (1979) 565–574
7. Mulmuley, K., Vazirani, U.V., Vazirani, V.V.: Matching is as easy as matrix inversion. Combinatorica **7**(1) (1987) 105–113
8. Blum, M., Chandra, A.K., Wegman, M.N.: Equivalence of free boolean graphs can be decided probabilistically in polynomial time. IPL **10** (1980) 80–82
9. Blum, M., Kannan, S.: Designing programs that check their work. J. ACM **42**(1) (1995) 269–291
10. Agrawal, M., Biswas, S.: Primality and identity testing via chinese remaindering. J. ACM **50**(4) (2003) 429–443
11. Agrawal, M., Kayal, N., Saxena, N.: PRIMES is in P. Ann. of Math. (2) **160**(2) (2004) 781–793
12. Shamir, A.: IP = PSPACE. J. ACM **39**(4) (1992) 869–877
13. Lund, C., Fortnow, L., Karloff, H., Nisan, N.: Algebraic methods for interactive proof systems. J. ACM **39**(4) (1992) 859–868
14. Arora, S., Safra, S.: Probabilistic checking of proofs: a new characterization of NP. J. ACM **45**(1) (1998) 70–122
15. Arora, S., Lund, C., Motwani, R., Sudan, M., Szegedy, M.: Proof verification and the hardness of approximation problems. J. ACM **45**(3) (1998) 501–555
16. Agrawal, M.: Proving lower bounds via pseudo-random generators. In: Proc. 25th FSTTCS, Springer (2005) 92–105
17. Dvir, Z., Shpilka, A.: Locally decodable codes with two queries and polynomial identity testing for depth 3 circuits. SIAM J. Comput. **36**(5) (2007) 1404–1434
18. Kayal, N., Saxena, N.: Polynomial identity testing for depth 3 circuits. In: Proc. 21st CCC, IEEE (2006) 9–17
19. Shpilka, A.: Interpolation of depth-3 arithmetic circuits with two multiplication gates. In: Proc. 39th STOC, ACM (2007) 284–293
20. Kabanets, V., Impagliazzo, R.: Derandomizing polynomial identity tests means proving circuit lower bounds. Comput. Complex. **13**(1/2) (2004) 1–46
21. Chen, Z.Z., Kao, M.Y.: Reducing randomness via irrational numbers. SIAM J. Comput. **29**(4) (2000) 1247–1256
22. Lewin, D., Vadhan, S.: Checking polynomial identities over any field: Towards a derandomization? In: Proc. 30th STOC, ACM (1998) 438–437
23. Klivans, A., Spielman, D.A.: Randomness efficient identity testing of multivariate polynomials. In: Proc. 33th STOC, ACM (2001) 216–223
24. Bogdanov, A.: Pseudorandom generators for low degree polynomials. In: Proc. 37th STOC, ACM (2005) 21–30
25. Ibarra, O.H., Moran, S.: Probabilistic algorithms for deciding equivalence of straight-line programs. J. ACM **30**(1) (1983) 217–228
26. Lipton, R., Vishnoi, N.: Deterministic identity testing for multivariate polynomials. In: Proc. SODA, ACM (2003) 756–760