




# Fooling Near-Maximal Decision Trees

William M. Hoza   

Department of Computer Science, The University of Chicago, IL, USA

Zelin Lv   

Department of Computer Science, The University of Chicago, IL, USA

---

## Abstract

For any constant  $\alpha > 0$ , we construct an explicit pseudorandom generator (PRG) that fools  $n$ -variate decision trees of size  $m$  with error  $\varepsilon$  and seed length  $(1 + \alpha) \cdot \log_2 m + O(\log(1/\varepsilon) + \log \log n)$ . For context, one can achieve seed length  $(2 + o(1)) \cdot \log_2 m + O(\log(1/\varepsilon) + \log \log n)$  using well-known constructions and analyses of small-bias distributions, but such a seed length is trivial when  $m \geq 2^{n/2}$ . Our approach is to develop a new variant of the classic concept of almost  $k$ -wise independence, which might be of independent interest. We say that a distribution  $X$  over  $\{0, 1\}^n$  is  $k$ -wise  $\varepsilon$ -probably uniform if every Boolean function  $f$  that depends on only  $k$  variables satisfies  $\mathbb{E}[f(X)] \geq (1 - \varepsilon) \cdot \mathbb{E}[f]$ . We show how to sample a  $k$ -wise  $\varepsilon$ -probably uniform distribution using a seed of length  $(1 + \alpha) \cdot k + O(\log(1/\varepsilon) + \log \log n)$ .

Meanwhile, we also show how to construct a set  $H \subseteq \mathbb{F}_2^n$  such that every feasible system of  $k$  linear equations in  $n$  variables over  $\mathbb{F}_2$  has a solution in  $H$ . The cardinality of  $H$  and the time complexity of enumerating  $H$  are at most  $2^{k+o(k)+\text{polylog } n}$ , whereas small-bias distributions would give a bound of  $2^{2k+O(\log(n/k))}$ .

By combining our new constructions with work by Chen and Kabanets (TCS 2016), we obtain nontrivial PRGs and hitting sets for linear-size Boolean circuits. Specifically, we get an explicit PRG with seed length  $(1 - \Omega(1)) \cdot n$  that fools circuits of size  $2.99 \cdot n$  over the  $U_2$  basis, and we get a hitting set with time complexity  $2^{(1-\Omega(1)) \cdot n}$  for circuits of size  $2.49 \cdot n$  over the  $B_2$  basis.

**2012 ACM Subject Classification** Theory of computation  $\rightarrow$  Pseudorandomness and derandomization

**Keywords and phrases** almost  $k$ -wise independence, decision trees, pseudorandom generators

**Digital Object Identifier** 10.4230/LIPIcs.APPROX/RANDOM.2025.35

**Category** RANDOM

**Related Version** *Full Version*: <https://ecc.weizmann.ac.il/report/2025/003/> [28]

**Acknowledgements** We thank Avishay Tal for valuable comments on a draft of this paper and for a discussion about the Fourier spectra of decision trees. We thank Frederic Koehler for pointing out the connection with Huber’s contamination model. We thank Alicia Torres Hoza for helpful comments on drafts of this paper. Zelin Lv thanks Aaron Potechin for valuable discussions.

## 1 Introduction

How many coin flips does it take to sample  $n$  bits that appear random from the perspective of an observer who only looks at  $0.9 \cdot n$  of the bits?

### 1.1 Almost $k$ -wise uniformity and $k$ -wise probable uniformity

*Almost  $k$ -wise uniformity* is a well-studied concept that provides one possible way of formalizing the question posed above.

► **Definition 1** (Almost  $k$ -wise uniformity). *Let  $X$  be a distribution over  $\{0, 1\}^n$ , let  $k \in [n]$ , and let  $\varepsilon \in [0, 1]$ . We say that  $X$  is  $\varepsilon$ -almost  $k$ -wise uniform if, for every size- $k$  set  $S \subseteq [n]$ , the total variation distance between  $X_S$  and  $U_k$  is at most  $\varepsilon$ . Here  $X_S$  denotes the projection of  $X$  to the coordinates in  $S$ , and  $U_k$  denotes the uniform distribution over  $\{0, 1\}^k$ . If  $\varepsilon = 0$ ,*



© William M. Hoza and Zelin Lv;

licensed under Creative Commons License CC-BY 4.0

Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2025).

Editors: Alina Ene and Eshan Chattopadhyay; Article No. 35; pp. 35:1–35:24



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

we simply say that  $X$  is  $k$ -wise uniform. An  $(\varepsilon$ -almost)  $k$ -wise uniform generator is a function  $G: \{0, 1\}^s \rightarrow \{0, 1\}^n$  such that  $G(U_s)$  is  $(\varepsilon$ -almost)  $k$ -wise uniform. We refer to  $s$  as the seed length of  $G$ .

When  $k \geq (\frac{1}{2} + \Omega(1)) \cdot n$  and  $\varepsilon = 0$ , Karloff and Mansour showed that every  $k$ -wise uniform generator has seed length at least  $n - O(1)$  [31], which might be disappointing. On the bright side, the seed length can be improved if a small positive error ( $\varepsilon > 0$ ) is permitted. Using a connection with “small-bias distributions” [39], Alon, Goldreich, Håstad, and Peralta constructed an explicit<sup>1</sup>  $\varepsilon$ -almost  $k$ -wise uniform generator with seed length  $k + O(\log(k/\varepsilon) + \log \log n)$  [6]. Notably, their seed length is meaningful even for large  $k$  such as  $k = 0.9 \cdot n$ .

In this work, we introduce a new variant of almost  $k$ -wise uniformity, called  *$k$ -wise probable uniformity*, which strengthens Definition 1. There are two equivalent definitions, described below.

► **Definition 2** ( *$k$ -wise probable uniformity*). Let  $X$  be a distribution over  $\{0, 1\}^n$ , let  $k \in [n]$ , and let  $\varepsilon \in [0, 1]$ . We say that  $X$  is  $k$ -wise  $\varepsilon$ -probably uniform if it satisfies either of the following two equivalent conditions.

1. For every size- $k$  set  $S \subseteq [n]$ , there exists a distribution  $E$  over  $\{0, 1\}^k$  such that the distribution  $X_S$  can be written as the mixture distribution  $X_S \equiv (1 - \varepsilon) \cdot U_k + \varepsilon \cdot E$ . That is, the distribution  $X_S$  is identical to the following distribution: With probability  $1 - \varepsilon$ , sample a  $k$ -bit string uniformly at random, and with probability  $\varepsilon$ , sample a string according to  $E$ .
2. For every  $k$ -junta<sup>2</sup>  $f: \{0, 1\}^n \rightarrow \{0, 1\}$ , we have  $\mathbb{E}[f(X)] \geq (1 - \varepsilon) \cdot \mathbb{E}[f]$ , where  $\mathbb{E}[f]$  is a shorthand for  $\mathbb{E}[f(U_n)]$ .

(See Section 3 for a proof that the two conditions above are equivalent.) We say that  $G: \{0, 1\}^s \rightarrow \{0, 1\}^n$  is a  $k$ -wise  $\varepsilon$ -probably uniform generator if  $G(U_s)$  is  $k$ -wise  $\varepsilon$ -probably uniform.

We find the first condition above to be more conceptually appealing. It is clearly a strengthening of  $\varepsilon$ -almost  $k$ -wise uniformity, and it inspires the terminology “ $k$ -wise  $\varepsilon$ -probably uniform.” On the other hand, we find the second condition above to be easier to work with mathematically.

The concept of  $k$ -wise probable uniformity is motivated primarily by an application to fooling decision trees, which we will discuss momentarily, but we also consider it to be an interesting concept in its own right. Using a standard nonconstructive argument (see the full version of this paper [28, §4.4]), one can show that there exists a non-explicit  $k$ -wise  $\varepsilon$ -probably uniform generator with seed length<sup>3</sup>

$$k + \log k + 2 \log(1/\varepsilon) + \log \log(n/k) + O(1). \quad (1)$$

The challenge is to construct an *explicit* generator.

Classic results regarding small-bias generators [39, 6] imply that there is an explicit  $k$ -wise  $\varepsilon$ -probably uniform generator with seed length  $2k + O(\log k + \log(1/\varepsilon) + \log \log n)$ .<sup>4</sup>

<sup>1</sup> We consider a generator  $G$  to be *explicit* if  $G(x)$  can be computed in  $\text{poly}(n)$  time, given the parameters (in this case  $n$ ,  $k$ , and  $\varepsilon$ ) and the seed  $x$ .

<sup>2</sup> A  $k$ -junta is a function  $f$  that depends on at most  $k$  variables.

<sup>3</sup> Throughout this paper,  $\log(\cdot)$  denotes the base-two logarithm.

<sup>4</sup> If  $X$  is  $k$ -wise  $\gamma$ -biased, then  $X$  is  $k$ -wise  $(\gamma \cdot 2^k)$ -probably uniform (see Lemma 17 and Proposition 18). Alon, Goldreich, Håstad, and Peralta construct an explicit  $k$ -wise  $\gamma$ -biased generator with seed length  $2 \log(1/\gamma) + O(\log k + \log \log n)$  [6]. Choose  $\gamma = \varepsilon \cdot 2^{-k}$ .

However, this seed length is unsatisfactory, because it is trivial when  $k \geq n/2$ . Meanwhile, Bshouty used a different approach (the method of conditional probabilities with pessimistic estimators) to construct a generator  $G: \{0, 1\}^s \rightarrow \{0, 1\}^n$  such that

$$(1 - \varepsilon) \cdot \mathbb{E}[f] \leq \mathbb{E}[f(G(U_s))] \leq (1 + \varepsilon) \cdot \mathbb{E}[f]$$

for every Boolean  $k$ -junta  $f$  [15], which is even stronger than Definition 2. Furthermore, his generator's seed length matches Equation (1). However, his generator's time complexity is more than  $\binom{n}{k} \cdot 2^k$  [15]. His generator can therefore be considered "explicit" only when  $k = O(1)$ , whereas we are primarily interested in the case  $k = \Theta(n)$ .

In this work, we present an explicit  $k$ -wise  $\varepsilon$ -probably uniform generator with seed length  $(1 + \alpha) \cdot k + O(\log(1/\varepsilon) + \log \log n)$ , where  $\alpha$  is an arbitrarily small positive constant and the constant hiding under the big- $O$  depends on  $\alpha$ .

► **Theorem 3** (Explicit  $k$ -wise probably uniform generator). *For every  $n, k \in \mathbb{N}$  and  $\varepsilon \in (0, 1)$ , there exists an explicit  $k$ -wise  $\varepsilon$ -probably uniform generator  $G: \{0, 1\}^s \rightarrow \{0, 1\}^n$  with seed length*

$$s = k + O\left(k^{2/3} \cdot \log^{1/3}(k/\varepsilon) + \log(1/\varepsilon) + \log \log n\right).$$

The simpler seed length bound  $(1 + \alpha) \cdot k + O(\log(1/\varepsilon) + \log \log n)$  follows from Theorem 3 by the weighted AM-GM inequality.

## 1.2 Fooling decision trees

Instead of modeling the observer as a  $k$ -junta, we can consider the more powerful model of *depth- $k$  decision trees*. A decision tree  $T$  makes queries to the input  $x$  and then produces a Boolean output value  $T(x)$ . The crucial feature of the decision tree model is that the tree can *adaptively* decide which variable to query next, based on the results of previous queries. (See Definition 11 for a precise definition.) Consequently, the output  $T(x)$  of a depth- $k$  decision tree  $T$  might depend on all  $n$  variables even if  $k \ll n$ . The problem of sampling bits that "appear random" to depth- $k$  decision trees can be formalized using the concept of a *pseudorandom generator (PRG)*.

► **Definition 4** (PRGs). *Let  $X$  be a distribution over  $\{0, 1\}^n$ , let  $f: \{0, 1\}^n \rightarrow \{0, 1\}$ , and let  $\varepsilon \in (0, 1)$ . We say that  $X$  fools  $f$  with error  $\varepsilon$  if*

$$|\mathbb{E}[f(X)] - \mathbb{E}[f]| \leq \varepsilon.$$

*We say that  $G: \{0, 1\}^s \rightarrow \{0, 1\}^n$  is a pseudorandom generator (PRG) that fools  $f$  with error  $\varepsilon$  if  $G(U_s)$  fools  $f$  with error  $\varepsilon$ . The parameter  $s$  is called the seed length of the PRG. If  $\mathcal{F}$  is a class of functions  $f: \{0, 1\}^n \rightarrow \{0, 1\}$ , we say that  $X$  (respectively  $G$ ) fools  $\mathcal{F}$  with error  $\varepsilon$  if  $X$  (respectively  $G$ ) fools every  $f \in \mathcal{F}$  with error  $\varepsilon$ .*

Almost  $k$ -wise uniformity is the special case of Definition 4 in which we take  $\mathcal{F}$  to be the class of all Boolean  $k$ -juntas. The aforementioned concept of small-bias distributions is another special case. By definition, a distribution  $X$  is  *$k$ -wise  $\gamma$ -biased* if it fools all functions of the form  $f(x) = \bigoplus_{i \in S} x_i$ , where  $S \subseteq [n]$  and  $|S| \leq k$ , with error  $\gamma/2$  [39].

To fool decision trees, one could try using a generic small-bias generator. This approach works extremely well in the nonadaptive setting, as mentioned previously. In the adaptive setting, the approach still works fairly well, but it turns out that *the parameters are worse*. Specifically, Kushilevitz and Mansour's analysis [34] implies that if  $X$  is  $k$ -wise  $\gamma$ -biased,

then  $X$  fools depth- $k$  size- $m$  decision trees with error  $\gamma \cdot m$ . Every depth- $k$  decision tree has size at most  $2^k$ , so we can choose  $\gamma = \varepsilon \cdot 2^{-k}$ . By combining this reduction with one of Alon, Goldreich, Håstad, and Peralta’s  $k$ -wise  $\gamma$ -biased generators [6], one can construct an explicit PRG that fools depth- $k$  decision trees with error  $\varepsilon$  and seed length  $2k + O(\log(k/\varepsilon) + \log \log n)$ . This seed length is sufficient for many purposes, but we emphasize that it gives us nothing nontrivial for trees of depth  $k \geq n/2$ .

In this paper, we show how to improve the leading constant from 2 to  $1 + \alpha$  for any constant  $\alpha > 0$ , as a consequence of our new  $k$ -wise  $\varepsilon$ -probably uniform generator. More generally, we prove the following.

► **Theorem 5** (Fooling near-maximal decision trees). *Let  $n, m \in \mathbb{N}$  and  $\varepsilon \in (0, 1)$ . There exists an explicit PRG  $G: \{0, 1\}^s \rightarrow \{0, 1\}^n$  that fools  $n$ -variate decision trees of size  $m$  with error  $\varepsilon$  and seed length*

$$s = \log m + O\left(\log^{2/3} m \cdot \log^{1/3}\left(\frac{\log m}{\varepsilon}\right) + \log(1/\varepsilon) + \log \log n\right).$$

Observe that our PRG is meaningful even for trees of near-maximal size such as  $m = 2^{0.9n}$ . Furthermore, it turns out that Theorem 5 extends to the more powerful model of size- $m$  “subcube partitions.” See Section 5 for further details.

### 1.3 A hitting set for systems of equations over $\mathbb{F}_2$

We also study a certain linear-algebraic variant of  $k$ -wise uniformity. We prove the following.

► **Theorem 6** (Hitting set for systems of equations over  $\mathbb{F}_2$ ). *For every  $n, k \in \mathbb{N}$ , there exists  $H \subseteq \mathbb{F}_2^n$  such that:*

1. *For every  $A \in \mathbb{F}_2^{k \times n}$  and every  $b \in \text{image}(A)$ , there exists  $x \in H$  such that  $Ax = b$ .*
2. *Given the parameters  $n$  and  $k$ , the set  $H$  can be enumerated in time  $T$  (and hence  $|H| \leq T$ ), where  $T = 2^{k+O((k \cdot \log k \cdot \log n)^{2/3} + \log n)}$ .*

We should compare Theorem 6 to what one can get by using a small-bias distribution. One can show that if  $X$  is  $n$ -wise  $\gamma$ -biased, then  $|\Pr[AX = b] - \Pr[AU_n = b]| \leq \gamma$  [34, 9]. If  $b \in \text{image}(A)$ , then  $\Pr[AU_n = b] \geq 2^{-k}$  by the rank-nullity theorem. Therefore, if we choose  $\gamma < 2^{-k}$ , the set  $H := \text{Supp}(X)$  satisfies Item 1 of Theorem 6. Plugging in one of Alon, Goldreich, Håstad, and Peralta’s  $\gamma$ -biased generators [6] would give us  $|H| \leq 2^{2k+O(\log(n/k))}$ . Essentially, Theorem 6 improves the coefficient of  $k$  in the exponent from  $2 - o(1)$  to  $1 + o(1)$ , although our dependence on  $n$  is worse.

Andreev, Baskakov, Clementi, and Rolim previously claimed to prove a similar theorem, with a bound of  $|H| \leq 2^{k+O(\sqrt{n-k} \cdot \log n)}$  [9]. This would be incomparable to Theorem 6: better when  $k \approx n$  and worse when  $k \ll n$ . However, there seems to be a mistake in their analysis.<sup>5</sup>

### 1.4 Applications: Pseudorandomness for linear-size Boolean circuits

Our results are motivated by applications in the area of *circuit complexity*. We consider circuits over the “ $B_2$ ” and “ $U_2$ ” bases. A  $B_2$ -circuit is a circuit in which each gate computes an arbitrary function  $\phi: \{0, 1\}^2 \rightarrow \{0, 1\}$ . A  $U_2$ -circuit is the same, except that gates are not

<sup>5</sup> Andreev, Baskakov, Clementi, and Rolim partition the variables into blocks,  $x = (x_1, \dots, x_s)$ , and they say that the condition  $Ax = b$  can be written as a conjunction of conditions  $A_1x_1 = b_1, \dots, A_sx_s = b_s$  [9, Appendix B, preprint version]. But this is not true in general.

permitted to compute the XOR function or its complement. Chen and Kabanets used “gate elimination” methods to establish, among other results, close connections between linear-size circuits and near-maximal decision trees [19]:

- Every  $U_2$ -circuit of size  $(3-\alpha) \cdot n$  can be simulated by a decision tree of size  $2^{(1-\Omega(\alpha^2)) \cdot n}$  [19].
- Every  $B_2$ -circuit of size  $(2.5-\alpha) \cdot n$  can be simulated by a *parity decision tree*<sup>6</sup> of size  $2^{(1-\Omega(\alpha^2)) \cdot n}$  [19].

They posed the problem of designing PRGs that fool general Boolean circuits [19]. By combining their simulations with our constructions, we are able to solve their problem, at least in part. First of all, we get a PRG that fools  $U_2$ -circuits of size  $(3-\alpha) \cdot n$ :

► **Corollary 7** (Fooling circuits over the  $U_2$  basis). *For every  $n \in \mathbb{N}$  and  $\alpha \in (0, 3)$ , there exists an explicit PRG  $G: \{0, 1\}^s \rightarrow \{0, 1\}^n$  that fools  $n$ -variate  $U_2$ -circuits of size  $(3-\alpha) \cdot n$  with error  $n \cdot 2^{-\Omega(\alpha^6 n)}$  and seed length  $s = (1 - \Omega(\alpha^2)) \cdot n$ .*

**Proof.** By Chen and Kabanets’ work [19], we know every  $U_2$ -circuit of size  $(3-\alpha) \cdot n$  can be simulated by a decision tree of size  $2^{(1-c\alpha^2) \cdot n}$  for some constant  $c > 0$ . By Theorem 5, we can fool such a tree with error  $2^{-c'\alpha^6 n} \cdot n$  and seed length

$$(1 - c\alpha^2) \cdot n + O(n^{2/3} \cdot (c'\alpha^6 n)^{1/3} + c'\alpha^6 n) = n - c\alpha^2 n + O(c'\alpha^2 n).$$

This is  $n - \Omega(\alpha^2 n)$  provided we choose  $c'$  to be a sufficiently small constant based on  $c$ . ◀

Second, we consider  $B_2$ -circuits. We have not managed to construct a genuine PRG that fools  $B_2$ -circuits, but we can at least use Theorem 6 to construct a *hitting set* for  $B_2$ -circuits. A hitting set is a relaxation of a PRG, defined as follows.

► **Definition 8.** *Let  $H \subseteq \{0, 1\}^n$ , let  $\mathcal{F}$  be a class of functions  $f: \{0, 1\}^n \rightarrow \{0, 1\}$ , and let  $\varepsilon \in (0, 1)$ . We say that  $H$  is an  $\varepsilon$ -hitting set for  $\mathcal{F}$  if, for every  $f \in \mathcal{F}$  such that  $\mathbb{E}[f] > \varepsilon$ , there exists  $x \in H$  such that  $f(x) = 1$ .*

► **Corollary 9** (A hitting set for circuits over the  $B_2$  basis). *For every  $n \in \mathbb{N}$  and  $\alpha \in (0, 2.5)$ , there exists a value  $\varepsilon = 2^{-\Omega(\alpha^2 n)}$  and a set  $H \subseteq \{0, 1\}^n$  such that:*

1.  $H$  is an  $\varepsilon$ -hitting set for  $B_2$ -circuits of size  $(2.5-\alpha) \cdot n$ .
2. Given the parameters  $n$  and  $\alpha$ , the set  $H$  can be enumerated in time  $2^{(1-\Omega(\alpha^2)) \cdot n + \tilde{O}(n^{2/3})}$ .

(The proof of Corollary 9 is in Section 6.)

### 1.4.1 Discussion

In general, the main motivation behind PRGs is that many algorithms and protocols rely on a large number of random bits, but producing truly random bits can sometimes be difficult or expensive. We think of randomness as a computational resource, similar to time or space. We try to use as little “true randomness” as possible to sample bits that are “random enough” to run randomized algorithms and protocols without distorting their behavior. With this motivation in mind, we believe that the problem of fooling  $U_2$ -circuits is extremely natural.

<sup>6</sup> A “parity decision tree” is defined like an ordinary decision tree, except that in each step, the tree can query to learn the parity of any subset of the variables, instead of querying just a single variable.

The PRG of Corollary 7 is the first of its kind.<sup>7</sup> Note that the challenge of constructing PRGs that fool Boolean circuits is strictly harder than the notorious challenge of proving circuit lower bounds. In more detail, suppose that one could construct a poly( $m$ )-time computable PRG  $G: \{0, 1\}^{\beta m-1} \rightarrow \{0, 1\}^m$  that fools  $U_2$ -circuits of size  $cm$  with error 0.49, where  $\beta \in (0, 1]$  and  $c > 1$  are constants. Let  $n = \beta m$ , and define  $G': \{0, 1\}^{n-1} \rightarrow \{0, 1\}^n$  by truncating the output of  $G$ . The indicator function for the image of  $G'$  would be an example of a function in NP that cannot be computed by  $U_2$ -circuits of size  $(c/\beta) \cdot n$ . Currently, the best lower bound known on the size of  $U_2$ -circuits computing some function in NP is  $(5 - o(1)) \cdot n$  [30].

Hitting sets are commonly used to solve the so-called ‘‘GAP-SAT’’ problem for  $\mathcal{F}$ , i.e., the problem of distinguishing the case  $f \equiv 0$  from the case  $\mathbb{E}[f] > \varepsilon$ , given  $f \in \mathcal{F}$ . Indeed, if  $H$  is an  $\varepsilon$ -hitting set for  $\mathcal{F}$ , then we can solve GAP-SAT for  $\mathcal{F}$  by computing  $\bigvee_{x \in H} f(x)$ . In this context, we should compare Corollary 9 to prior circuit analysis algorithms. Savinov designed a SAT algorithm for  $B_2$ -circuits of size  $m$  with time complexity  $O(2^{0.389667 \cdot m})$  [46, 35], improving prior work by Nurk [41]. Golovnev, Kulikov, Smal, and Tamaki designed a #SAT algorithm for  $B_2$ -circuits of size  $2.99 \cdot n$  with time complexity  $2^{(1-\Omega(1)) \cdot n}$  [23], improving a result by Chen and Kabanets [19]. These prior algorithms solve problems that are harder than GAP-SAT, and furthermore they can handle circuits that are larger than what Corollary 9 can handle. However, Corollary 9 is superior to these prior results in one respect, namely, we can solve GAP-SAT even if we only have *query access* to the circuit in question. Note that the ‘‘black box’’ nature of hitting sets is crucial in some applications. For example, Cheng and Hoza showed that optimal explicit hitting sets for space-bounded computation would imply  $L = BPL$ , whereas it remains an open problem to prove  $L = BPL$  if we merely assume the existence of an optimal GAP-SAT algorithm for space-bounded computation [20, 43].

## 1.5 Overview of our new constructions

### 1.5.1 Our $k$ -wise probably uniform generator (Theorem 3)

The starting point of our construction is the well-known sampling properties of *pairwise uniform hash functions*. Let  $f: \{0, 1\}^n \rightarrow \{0, 1\}$  be any nonzero  $k$ -junta, or more generally any function such that  $\mathbb{E}[f] \geq 2^{-k}$ . If we sample a hash function  $h: \{0, 1\}^{k+O(\log(1/\varepsilon))} \rightarrow \{0, 1\}^n$  from a pairwise uniform family, then with high probability over the choice of  $h$ , we have

$$\mathbb{E}_x[f(h(x))] \geq (1 - \varepsilon) \cdot \mathbb{E}[f].$$

(This follows from Chebyshev’s inequality.)

We can think of  $h$  as a PRG with an excellent seed length. The only trouble is that sampling  $h$  itself is expensive. In general, sampling a hash function  $h: \{0, 1\}^q \rightarrow \{0, 1\}^\ell$  from a pairwise uniform family costs  $\Theta(q + \ell)$  truly random bits, so in our case, the cost is  $\Theta(n + \log(1/\varepsilon))$  truly random bits, which is much more than we can afford.

<sup>7</sup> To be fair, we should compare Corollary 7 to a different and rather trivial approach that one could use to construct PRGs that fool circuits. In general, if  $h: \{0, 1\}^{n-1} \rightarrow \{0, 1\}$  is average-case hard for circuits of size  $cn$ , then the generator  $G(x) = (x, h(x))$  maps  $n - 1$  bits to  $n$  bits and fools circuits of size  $cn$ . Similarly, the generator  $G'(x, y) = (x, y, h(x), h(y))$  maps  $n' - 2$  bits to  $n'$  bits and fools circuits of size  $(c/2) \cdot n'$ , where  $n' = 2n$ . One can similarly try  $G''(x, y, z) = (x, y, z, h(x), h(y), h(z))$ , etc. One can instantiate this approach with known average-case hardness results for circuits over the  $U_2$  basis or the full binary basis [19, 23]. However, the PRGs that can be constructed using this approach have seed length  $n - O(1)$ . The seed length is what makes Corollary 7 interesting. If  $\alpha$  is constant, then our PRG has linear stretch.

We can slightly decrease the cost of sampling  $h$  by composing with a  $\gamma$ -almost  $k$ -wise uniform generator, where  $\gamma \approx \varepsilon \cdot 2^{-k}$ , with seed length  $\ell = O(k + \log(1/\varepsilon) + \log \log n)$ . Such a generator fools  $f$  with error  $\gamma$ , which is negligible. Now the output length of  $h$  is decreased from  $n$  down to  $\ell$ , hence the cost of sampling  $h$  is “only”  $O(k + \log(1/\varepsilon) + \log \log n)$ . However, this cost is still more than we can afford.

To explain how we bring the cost down to  $o(k)$ , for simplicity’s sake, let us assume that  $\varepsilon = 1/\text{poly}(k)$  and let us neglect  $\log \log n$  terms. We can assume without loss of generality that  $f$  is simply a conjunction of  $k$  literals, because every  $k$ -junta can be written as a sum of such functions. Our approach is to pseudorandomly partition the  $n$  coordinates into  $r = \tilde{\Theta}(k^{1/3})$  buckets:  $[n] = B_1 \cup \dots \cup B_r$ . In expectation, each bucket contains  $k/r$  of the  $k$  relevant variables. With high probability, each bucket has at most  $k_0$  of the variables, where  $k_0 = k/r + \tilde{O}(\sqrt{k/r}) = k/r + \tilde{O}(k^{1/3})$ .

We can write  $f(x) = f_1(x) \wedge \dots \wedge f_r(x)$ , where  $f_i(x)$  only depends on variables in  $B_i$ , so  $f_i$  is a  $k_0$ -junta. We sample a hash function  $h: \{0, 1\}^{k_0 + O(\log k)} \rightarrow \{0, 1\}^n$  such that with high probability over the choice of  $h$ , we have

$$\mathbb{E}_x[f_i(h(x))] \geq \left(1 - \frac{1}{\text{poly}(k)}\right) \cdot \mathbb{E}[f_i].$$

For each bucket  $B_i$  independently, we sample  $x$  at random and put  $h(x)$  in  $B_i$ . Crucially, we reuse the same hash function  $h$  for all of the buckets, which is justified by a simple union bound. The cost of sampling  $h$  is  $O(k_0) = \tilde{O}(k^{2/3})$  truly random bits, and the cost of sampling the  $x$  values is

$$r \cdot (k_0 + O(\log k)) = k + \tilde{O}(k^{2/3}).$$

A more careful calculation, also taking into account the cost of sampling the partition  $[n] = B_1 \cup \dots \cup B_r$ , leads to the seed length bound that appears in Theorem 3.

Observe that in this construction, there are some “bad events” that occur with probability roughly  $\varepsilon$ , namely, we might get a “bad” partition of the variables into buckets or we might get a “bad” hash function  $h$ . Let  $B$  be the union of these bad events. To analyze the impact of these bad events, let  $X$  be the output distribution of our generator and let  $f$  be an arbitrary Boolean  $k$ -junta. Then

$$\mathbb{E}[f(X)] = \underbrace{\Pr[B] \cdot \mathbb{E}[f(X) \mid B]}_{(*)} + \Pr[\neg B] \cdot \mathbb{E}[f(X) \mid \neg B].$$

The quantity marked  $(*)$  is certainly nonnegative, which allows us to prove  $\mathbb{E}[f(X)] \geq (1 - \varepsilon) \cdot \mathbb{E}[f]$ . On the other hand, note that the quantity marked  $(*)$  might be much larger than  $\mathbb{E}[f]$ , and hence we are not able to prove an upper bound of the form  $\mathbb{E}[f(X)] \leq (1 + \varepsilon) \cdot \mathbb{E}[f]$ . Thankfully, such an upper bound is not necessary for our applications.

### 1.5.2 Our hitting set for systems of equations over $\mathbb{F}_2$ Theorem 6)

The first step of the proof of Theorem 6 is to apply a *rank condenser* due to Forbes and Guruswami [22]. This allows us to assume without loss of generality that  $k \geq \Omega(n/\log n)$ . The next step is to partition the variables into  $t$  equal-sized blocks, each containing  $n/t$  variables, where  $t \approx n^{2/3}$ . This induces a partition of the columns of  $A$ :  $A = [A_1 \ A_2 \ \dots \ A_t]$ . Let  $k_i$  be the contribution of  $A_i$  to the rank of  $A$ , so  $k_1 + \dots + k_t \leq k$ . A lemma by Andreev, Clementi, and Rolim says that if  $H_\ell$  is a hitting set for systems of  $\ell$  equations in  $n/t$  variables, then there is some  $x \in H_{k_1} \times \dots \times H_{k_t}$  such that  $Ax = b$  [10]. We construct  $H_\ell$  for every  $\ell$  by a simple brute-force algorithm, which we can afford because the number of variables is small, and then we output the union of  $H_{k_1} \times \dots \times H_{k_t}$  over all possible partitions  $k = k_1 + \dots + k_t$ .

## 1.6 Limitations of $k$ -wise $\gamma$ -biased generators

A great deal of effort has been spent trying to optimize the constant factors in the seed lengths of small-bias generators [39, 5, 6, 12, 15, 51, 13]. Researchers have also developed many sophisticated techniques for proving that small-bias generators fool various models of computation; see Hatami and Hoza’s survey for a few examples [27]. The reader might reasonably wonder whether one could have proven our results by simply improving known constructions or analyses of  $k$ -wise  $\gamma$ -biased distributions. We prove that the answer is no. In more detail, in the full version of this paper [28, §6], we present examples showing that if the support of every  $k$ -wise  $\gamma$ -biased distribution is a 0.49-hitting set for  $U_2$ -circuits of size  $2n$ , then  $k \geq \frac{2}{3} \cdot n$  and  $\gamma \leq O(2^{-n/2})$ . Then, we observe that Karloff and Mansour’s work [31] can be extended to prove the following lower bound on the seed length of  $k$ -wise  $\gamma$ -biased generators in the regime  $k \geq (\frac{1}{2} + \Omega(1)) \cdot n$ .

► **Theorem 10** (Seed length lower bound for  $k$ -wise  $\gamma$ -biased generators). *Let  $G: \{0, 1\}^s \rightarrow \{0, 1\}^n$  be a  $k$ -wise  $\gamma$ -biased generator, where  $k = \lfloor (1/2 + \alpha) \cdot n \rfloor$  for some  $\alpha \in (0, 1/2]$ . Then*

$$s \geq \min\{n, 2 \log(1/\gamma)\} - \log(1/\alpha) - O(1).$$

Consequently, if one tries using a generic  $k$ -wise  $\gamma$ -biased generator to hit  $U_2$ -circuits of size  $2n$ , then the seed length will inevitably be at least  $n - O(1)$ . Thus, the concept of  $k$ -wise  $\gamma$ -biased distributions is inherently too weak to prove Corollaries 7 and 9. In turn, this implies that the concept of  $k$ -wise  $\gamma$ -bias is also too weak to prove our main results (Theorems 3, 5, and 6), of which Corollaries 7 and 9 are applications.<sup>8</sup>

For context, a sequence of prior works [44, 21, 4, 6, 2, 3, 15] has shown that every  $k$ -wise  $\gamma$ -biased generator  $G: \{0, 1\}^s \rightarrow \{0, 1\}^n$  has seed length at least

$$\min \left\{ \log \left( \binom{n}{\leq k/2} \right), \quad 2 \log(1/\gamma) + \log \log \left( \binom{n}{\leq k/2} \right) - \log \log(1/\gamma) \right\} - O(1). \quad (2)$$

Equation (2) and Theorem 10 are incomparable in general, but our new Theorem 10 is superior in the parameter regime in which we are interested. In particular, if  $\gamma = O(2^{-n/2})$  and  $k = cn$  for a constant  $1/2 < c < 1$ , then the prior bound Equation (2) is  $(1 - \Omega(1)) \cdot n$ , whereas our new Theorem 10 gives a bound of  $n - O(1)$ .

## 1.7 Related work

### 1.7.1 Approximate forms of $k$ -wise uniformity

Prior researchers have studied several different ways of quantifying what it means for a distribution  $X$  over  $\{0, 1\}^n$  to be “approximately”  $k$ -wise uniform.

- We could require that the total variation distance between  $X_S$  and  $U_k$  is at most  $\varepsilon$  for every size- $k$  set  $S \subseteq [n]$ . This is the definition of an  $\varepsilon$ -almost  $k$ -wise uniform distribution (Definition 1). See, for example, work by Naor and Naor [39] and work by Alon, Goldreich, Håstad, and Peralta [6].
- We could require that  $|\Pr[\bigoplus_{i \in S} X_i = 1] - \Pr[\bigoplus_{i \in S} X_i = 0]| \leq \varepsilon$  for every nonempty set  $S \subseteq [n]$  of size at most  $k$  [39]. This is the definition of a  $k$ -wise  $\varepsilon$ -biased distribution. See, for example, the works mentioned above [39, 6].

<sup>8</sup> Our results are actually quantitatively stronger in various respects; see the full version of this paper [28] for details.

- We could require that the  $\ell_\infty$  distance between  $X_S$  and  $U_k$  is at most  $\varepsilon$  for every size- $k$  set  $S \subseteq [n]$ . See, for example, work by Alon, Goldreich, Håstad, and Peralta [6] and work by Bshouty [15].
- We could require that  $X$  is  $\varepsilon$ -close in total variation distance to some exactly  $k$ -wise uniform distribution  $X'$ . See, for example, work by Alon, Goldreich, and Mansour [7]; work by Alon, Andoni, Kaufman, Matulef, Rubinfeld, and Xie [3]; and work by O'Donnell and Zhao [42].

Despite the attention paid to all of the above variations, we seem to be the first to study the concept of  $k$ -wise probable uniformity.

### 1.7.2 Huber's contamination model

Our notion of “probable uniformity” is similar to Huber's contamination model in the theory of robust statistics [29]. A key difference is that in Huber's model, contamination is applied to an *unknown* distribution, whereas in a  $k$ -wise probably uniform distribution, every  $k$  coordinates are distributed according to a contaminated version of the *uniform* distribution.

### 1.7.3 Universal sets

A set  $H \subseteq \{0, 1\}^n$  is called *k-universal* if, for every size- $k$ -set  $S \subseteq [n]$  and every  $z \in \{0, 1\}^k$ , there exists  $x \in H$  such that  $x_S = z$ . The concept of  $k$ -universal sets has been studied in many prior works going back more than half a century [32, 18, 52, 1, 48, 11, 5, 39, 40, 14]. The best explicit construction, due to Naor, Schulman, and Srinivasan [40], has cardinality  $2^{k+O(\log^2 k)} \cdot \log n$ . Our constructions were inspired by Naor, Schulman, and Srinivasan's universal set construction [40].

The notion of  $k$ -wise probable uniformity can be considered a strengthening of  $k$ -universality, because if  $X$  is  $k$ -wise probably uniform, then the support of  $X$  is  $k$ -universal. Consequently, Theorem 3 implies the existence of an explicit  $k$ -universal set with cardinality  $2^{k+\tilde{O}(k^{2/3})} \cdot \text{polylog } n$ , but this is inferior to Naor, Schulman, and Srinivasan's construction [40].<sup>9</sup> Our  $k$ -wise uniform generator also has similarities with a recent construction of a “biased” variant of universal sets by Harel, Hoza, Vardi, Evron, Srebro, and Soudry [26].

Similarly, the set  $H$  of Theorem 6 is  $k$ -universal, because the condition  $x_S = z$  can be expressed as a system of  $k$  equations. Once again, the cardinality of this set is greater than the cardinality of Naor, Schulman, and Srinivasan's universal set [40].

### 1.7.4 PRGs based on pseudorandom partitions of the variables

The trick of pseudorandomly partitioning the variables into buckets is not new; similar tricks have been used in many prior PRG constructions. For a few examples that are especially similar to our work, see work by Meka and Zuckerman [38], work by Lovett, Reingold, Trevisan, and Vadhan [36], and work by Gopalan, Kane, and Meka [25].

<sup>9</sup> A  $k$ -universal set  $H$  is typically considered “explicit” if the entire set can be computed in  $\text{poly}(|H|)$  time. Our set has stronger explicitness guarantees, which might possibly be of value, but note that Naor, Schulman, and Srinivasan already constructed a  $k$ -universal set of cardinality  $2^{k+o(k)} \cdot \log n$  with similar explicitness guarantees [40].

### 1.7.5 Correlation bounds for general circuit models

In general, PRGs are intimately related to *correlation bounds*, aka average-case hardness. Loosely speaking, correlation bounds are a prerequisite to designing PRGs. See, e.g., Hatami and Hoza’s survey [27, Chapter 4] for further discussion. Chen and Kabanets proved the first correlation bounds for general, unbounded-depth circuit models [19], and our PRG for  $U_2$ -circuits uses their work, as mentioned previously. Golovnev, Kulikov, Smal, and Tamaki subsequently proved better correlation bounds [23].

## 2 Preliminaries

### 2.1 Decision tree models

Below we record the standard definitions of a decision tree and parity decision trees.

► **Definition 11** (Decision trees). *An  $n$ -variate decision tree is a rooted tree  $T$  in which each internal node is labeled with a variable from among  $x_1, \dots, x_n$ ; each internal node has two outgoing edges labeled 0 and 1; and each leaf is labeled either 0 or 1. The tree  $T$  computes a Boolean function  $T: \{0, 1\}^n \rightarrow \{0, 1\}$  defined inductively as follows. If  $T$  consists of a single leaf labeled  $b \in \{0, 1\}$ , then we define  $T(x) \equiv b$ . Otherwise, let  $x_i$  be the variable labeling the root node. Given an input  $x \in \{0, 1\}^n$ , we start at the root node and traverse the outgoing edge labeled with the value  $x_i$ . This leads to a vertex  $u$ , which is the root of a subtree  $T'$ . Then we set  $T(x) = T'(x)$ . The depth of the tree is the length of the longest path from the root to a leaf. The size of the tree is the total number of leaves.*

► **Definition 12** (Parity decision trees [34]). *A parity decision tree on variables  $x_1, \dots, x_n$  is a rooted tree  $T$  defined exactly as in Definition 11, except that each internal node is labeled by a non-empty subset  $S \subseteq [n]$ . The internal node queries  $\bigoplus_{i \in S} x_i$  and has two outgoing edges labeled 0 and 1 corresponding to the value of that parity. Leaves are labeled by output bits in  $\{0, 1\}$ , and evaluation proceeds exactly as for ordinary decision trees. The depth of a parity decision tree is the length of the longest root-to-leaf path, and its size is the number of leaves. Equivalently, a parity decision tree computes a Boolean function*

$$f(x_1, \dots, x_n) = T \left( \bigoplus_{i \in S_1} x_i, \dots, \bigoplus_{i \in S_m} x_i \right),$$

where  $T$  is an ordinary decision tree on  $m$  inputs and  $S_1, \dots, S_m \subseteq [n]$ . computation.

### 2.2 Pairwise uniform hashing

We rely on the standard notion of a *pairwise uniform hashing*, aka “strongly universal hashing,” introduced in Carter and Wegman’s seminal papers [16, 53].

► **Definition 13** (Pairwise uniform families of hash functions). *A family  $\mathcal{H}$  of hash functions  $h: \{0, 1\}^q \rightarrow \{0, 1\}^\ell$  is called pairwise uniform if, for every two distinct  $x, x' \in \{0, 1\}^q$ , if we sample  $h \sim \mathcal{H}$ , then  $(h(x), h(x'))$  is distributed uniformly at random over  $\{0, 1\}^{2\ell}$ .*

► **Theorem 14** (Explicit pairwise uniform families of hash functions). *For every  $q, \ell \in \mathbb{N}$ , there exists an explicit<sup>10</sup> pairwise uniform family  $\mathcal{H}$  of hash functions  $h: \{0, 1\}^q \rightarrow \{0, 1\}^\ell$  such that  $h \in \mathcal{H}$  can be sampled using a seed of length  $O(q + \ell)$ .*

<sup>10</sup>That is, given a seed  $x \in \{0, 1\}^{O(q+\ell)}$  and an input  $y \in \{0, 1\}^q$ , the value  $h_x(y)$  can be computed in  $\text{poly}(q, \ell)$  time, where  $h_x$  is the hash function corresponding to the seed  $x$ .

For example, if we define  $h_{a,b}(x) = a * x + b$ , where  $*$  is convolution mod 2 and  $+$  is bitwise XOR, then  $\{h_{a,b} : a \in \{0, 1\}^{q+\ell-1} \text{ and } b \in \{0, 1\}^\ell\}$  is a pairwise uniform family [37]. The reason pairwise uniform hashing is useful for us is given by the following relative-error sampling lemma.

► **Lemma 15** (Pairwise uniformity sampling lemma). *Let  $\mathcal{H}$  be a pairwise uniform family of hash functions  $h: \{0, 1\}^q \rightarrow \{0, 1\}^\ell$ . Let  $f: \{0, 1\}^\ell \rightarrow \{0, 1\}$  and let  $\mu = \mathbb{E}[f]$ . Then for every  $\varepsilon \in (0, 1)$ ,*

$$\Pr_{h \sim \mathcal{H}} [h \text{ fools } f \text{ with error } \varepsilon \cdot \mu] \geq 1 - \frac{1}{2^q \cdot \varepsilon^2 \cdot \mu}.$$

**Proof.** For each  $x \in \{0, 1\}^q$ , define  $Z_x = f(h(x))$ , so  $Z_x$  is a random variable based on the choice of  $h \sim \mathcal{H}$ . Then  $\mathbb{E}[Z_x] = \mu$  and  $\text{Var}[Z_x] = \mu \cdot (1 - \mu) \leq \mu$ . Furthermore, the variables  $Z_x$  are pairwise independent. Therefore, if we let  $Z = \sum_x Z_x$ , then  $\mathbb{E}[Z] = 2^q \cdot \mu$  and  $\text{Var}[Z] \leq 2^q \cdot \mu$ . Therefore, by Chebyshev's inequality, we have

$$\Pr[|Z - 2^q \cdot \mu| \geq \varepsilon \cdot \mu \cdot 2^q] \leq \frac{\text{Var}[Z]}{\varepsilon^2 \cdot \mu^2 \cdot 2^{2q}} \leq \frac{1}{2^q \cdot \varepsilon^2 \cdot \mu}. \quad \blacktriangleleft$$

### 2.3 Small-bias distributions

We also rely on asymptotically optimal constructions of  $k$ -wise  $\gamma$ -biased generators, which were defined in Section 1.2.

► **Theorem 16** (Explicit  $k$ -wise  $\gamma$ -biased generators [39]). *For every  $n, k \in \mathbb{N}$  and every  $\gamma \in (0, 1)$ , there exists an explicit  $k$ -wise  $\gamma$ -biased generator  $G: \{0, 1\}^s \rightarrow \{0, 1\}^n$  with seed length  $O(\log(k/\gamma) + \log \log n)$ .*

The reason  $k$ -wise  $\gamma$ -biased generators are useful for us is that they satisfy the following two properties.

► **Lemma 17** (Small-bias generators fool juntas and conjunctions of literals [39, 6]). *Let  $X$  be a  $k$ -wise  $\gamma$ -biased distribution over  $\{0, 1\}^n$ . Then  $X$  is  $\varepsilon$ -almost  $k$ -wise uniform, where  $\varepsilon = \gamma \cdot 2^{k/2}$ . Furthermore,  $X$  fools every conjunction of at most  $k$  literals with error  $\gamma$ .*

## 3 Characterizing $k$ -wise probable uniformity

The following proposition shows the equivalence of three ways of defining  $k$ -wise probably uniform distributions.

► **Proposition 18** (Equivalence of three definitions of  $k$ -wise probable uniformity). *Let  $X$  be a distribution over  $\{0, 1\}^n$ , let  $k \in [n]$ , and let  $\varepsilon \in [0, 1]$ . Then the following are equivalent.*

1. *For every  $k$ -junta  $f: \{0, 1\}^n \rightarrow \{0, 1\}$ , we have  $\mathbb{E}[f(X)] \geq (1 - \varepsilon) \cdot \mathbb{E}[f]$ .*
2. *For every size- $k$  set  $S \subseteq [n]$  and every  $z \in \{0, 1\}^k$ , we have  $\Pr[X_S = z] \geq (1 - \varepsilon) \cdot 2^{-k}$ .*
3. *For every size- $k$  set  $S \subseteq [n]$ , there exists a distribution  $E$  over  $\{0, 1\}^k$  such that one can sample from  $X_S$  by sampling from  $U_k$  with probability  $1 - \varepsilon$  and sampling from  $E$  with probability  $\varepsilon$ .*

## 35:12 Fooling Near-Maximal Decision Trees

**Proof.**

- (1  $\implies$  2) Consider the function  $f(x) = 1 \iff x_S = z$ .
- (2  $\implies$  3) If  $\varepsilon = 0$ , then for every  $x \in \{0, 1\}^k$ , we have  $\Pr[X_S = x] \geq 2^{-k}$ , which implies that  $X_S$  is exactly uniform over  $\{0, 1\}^k$ . If  $\varepsilon > 0$ , define  $p: \{0, 1\}^k \rightarrow \mathbb{R}$  by the formula

$$p(x) = \frac{\Pr[X_S = x] - (1 - \varepsilon) \cdot 2^{-k}}{\varepsilon}.$$

Then  $p(x)$  is a probability mass function: it is nonnegative because  $\Pr[X_S = x] \geq (1 - \varepsilon) \cdot 2^{-k}$ , and it sums to 1 because  $X_S$  is a probability distribution. Let  $E$  be corresponding probability distribution.

- (3  $\implies$  1) If  $f$  is a  $k$ -junta, then there is some set  $S \subseteq [n]$  of size  $k$  and some function  $g: \{0, 1\}^k \rightarrow \{0, 1\}$  such that  $f(x) = g(x_S)$  for all  $x \in \{0, 1\}^n$ . Therefore,

$$\mathbb{E}[f(X)] = \mathbb{E}[g(X_S)] = (1 - \varepsilon) \cdot \mathbb{E}[g(U_k)] + \varepsilon \cdot \mathbb{E}[g(E_S)] \geq (1 - \varepsilon) \cdot \mathbb{E}[f]. \quad \blacktriangleleft$$

By definition, if  $X$  satisfies any of the three equivalent conditions in Proposition 18, then  $X$  is  $k$ -wise  $\varepsilon$ -probably uniform. The third condition in Proposition 18 motivates the name “ $k$ -wise probably uniform,” but we find it more mathematically convenient to work with the first two conditions.

### 4 Constructing $k$ -wise probably uniform generators

In this section, we present our new  $k$ -wise probably uniform generator, thereby proving Theorem 3. At the end of this section, for completeness’ sake, we record the standard nonconstructive proof of the existence of nonexplicit  $k$ -wise probably uniform generators with excellent seed lengths.

#### 4.1 A small family of generators, each with a good seed length

As a first step, we begin by constructing a family of generator  $\mathcal{G}$ , such that for any  $k_0$ -junta  $f$ , most generators  $g \in \mathcal{G}$  satisfy  $(1 - \zeta) \cdot \mathbb{E}[f] \leq \mathbb{E}_x[f(g(x))] \leq (1 + \zeta) \cdot \mathbb{E}[f]$ . This construction is based on a combination of pairwise uniform hash functions and  $k$ -wise  $\gamma$ -biased generators.

► **Lemma 19** (Family of generators). *For every  $n, k_0 \in \mathbb{N}$  and  $\zeta \in (0, 1)$ , there exists an explicit family  $\mathcal{G}$  of PRGs  $g: \{0, 1\}^q \rightarrow \{0, 1\}^n$  satisfying the following.*

1. A generator  $g \sim \mathcal{G}$  can be sampled using  $O(k_0 + \log(1/\zeta) + \log \log n)$  truly random bits.
2. Each generator  $g$  in  $\mathcal{G}$  has seed length  $q = k_0 + O(\log(1/\zeta))$ .
3. If  $f: \{0, 1\}^n \rightarrow \{0, 1\}$  is a  $k_0$ -junta with expectation  $\mathbb{E}[f] = \mu$ , then

$$\Pr_{g \sim \mathcal{G}} [g \text{ fools } f \text{ with error } \zeta \cdot \mu] \geq 1 - \zeta.$$

**Proof.** Let  $G_{\text{sb}}: \{0, 1\}^\ell \rightarrow \{0, 1\}^n$  be a  $k$ -wise  $\gamma$ -biased generator where  $\gamma = (\zeta/3) \cdot 2^{-3k_0/2}$  and

$$\ell = O(k_0 + \log(1/\zeta) + \log \log n).$$

Let  $\mathcal{H}$  be a pairwise uniform family of hash functions  $h: \{0, 1\}^q \rightarrow \{0, 1\}^\ell$ . For each hash function  $h$  in  $\mathcal{H}$ , we define a generator  $g(x) = G_{\text{sb}}(h(x))$ . By Theorems 14 and 16, this family is explicit and  $\mathcal{G}$  can be sampled using  $O(k_0 + \log(1/\zeta) + \log \log n)$  truly random bits.

For the correctness proof, define  $f': \{0, 1\}^\ell \rightarrow \{0, 1\}$  by  $f'(y) = f(G_{\text{sb}}(y))$  and let  $\mu' = \mathbb{E}[f']$ . The generator  $G_{\text{sb}}$  fools  $f$  with error  $\gamma \cdot 2^{k_0/2}$  (see Lemma 17), so  $|\mu - \mu'| \leq \zeta/3 \cdot \mu$ . Furthermore,  $\mu \geq 2^{-k_0}$  unless  $f \equiv 0$ , so  $\mu' \geq 2^{-k_0-1}$ . Therefore, by the pairwise uniformity sampling lemma (Lemma 15), we have

$$\Pr_{h \sim \mathcal{H}} [h \text{ fools } f' \text{ with error } (\zeta/3) \cdot \mu'] \geq 1 - \frac{9}{2^q \cdot \zeta^2 \cdot \mu'} \geq 1 - \frac{18 \cdot 2^{k_0}}{2^q \cdot \zeta^2} \geq 1 - \zeta,$$

provided we choose a suitable value  $q = k_0 + O(\log(1/\zeta))$ . Now fix an  $h$  such that the bad event above does not occur, and let  $g$  be the corresponding generator in  $\mathcal{G}$ , i.e.,  $g(x) = G_{\text{sb}}(h(x))$ . Then  $g$  fools  $f$  with error

$$(\zeta/3) \cdot \mu' + |\mu - \mu'| \leq \mu \cdot (\zeta/3) \cdot (2 + \zeta/3) \leq \zeta \cdot \mu. \quad \blacktriangleleft$$

## 4.2 Pseudorandomly partitioning the coordinates into buckets

In this subsection, we explain how to pseudorandomly partition the coordinates into buckets,  $[n] = B_1 \cup \dots \cup B_r$ , such that no single bucket gets too many of the  $k$  coordinates we care about. To be more precise, we construct a *balanced partition generator*, defined as follows.

► **Definition 20** (Balanced partition generator [38]). *A  $(k, k_0, \delta)$ -balanced partition generator is a function  $G_{\text{vars}}: \{0, 1\}^a \rightarrow [r]^n$  such that for every set  $S \subseteq [n]$  with  $|S| \leq k$ , with probability at least  $1 - \delta$  over a uniform random choice of seed  $x \in \{0, 1\}^a$ , for every bucket  $j \in [r]$ , we have  $|\{i \in S : G_{\text{vars}}(x)_i = j\}| \leq k_0$ .*

Definition 20 is due to Meka and Zuckerman, who used the term “balanced hash family” [38, Definition 4.9]. We use the term “balanced partition generator” to avoid confusion with the hash functions that appear in the proof of Lemma 19. Our balanced partition generator will essentially consist of a  $d$ -wise  $\gamma$ -biased generator for appropriate values  $d$  and  $\gamma$ . The analysis will be based on the following bound on the moments of a sum of independent Bernoulli random variables [47].<sup>11</sup>

► **Theorem 21** (Moment bound for a sum of independent Bernoulli random variables [47]). *Let  $X_1, \dots, X_k$  be independent  $\{0, 1\}$ -valued random variables. Let  $X = \sum_{i=1}^k X_i$ , let  $\mu_i = \mathbb{E}[X_i]$ , and let  $\mu = \sum_{i=1}^k \mu_i$ . Then for every even positive integer  $t$ , we have*

$$\mathbb{E}[(X - \mu)^t] \leq \max\{t^t, (t\mu)^{t/2}\}.$$

Theorem 21 can be improved in some parameter regimes [49], but the simple bound in Theorem 21 suffices for our purposes. Using Theorem 21, we now present a tail bound for sums of random variables that satisfy a certain “near  $t$ -wise independence” condition. Similar bounds were proven in several previous papers [36, 17, 50], and our proof is almost identical to their proofs.

► **Corollary 22** (Tail bound for sums of nearly  $t$ -wise independent random variables). *Let  $X_1, \dots, X_k$  be  $\{0, 1\}$ -valued random variables and let  $\mu_1, \dots, \mu_k \in [0, 1]$ . Let  $X = \sum_{i=1}^k X_i$  and  $\mu = \sum_{i=1}^k \mu_i$ . Let  $t$  be an even positive integer, let  $\gamma \in (0, 1)$ , and assume that for every set  $S \subseteq [k]$  with  $|S| \leq t$ , we have*

$$\left| \mathbb{E} \left[ \prod_{i \in S} X_i \right] - \prod_{i \in S} \mu_i \right| \leq \gamma.$$

<sup>11</sup>The exact statement of Theorem 21 does not appear in Schmidt, Siegel, and Srinivasan’s work [47], but it follows from the proof of item “(III)” in their “Theorem 4.”

### 35:14 Fooling Near-Maximal Decision Trees

Then for every  $\Delta > 0$ , we have

$$\Pr[|X - \mu| \geq \Delta] \leq \left(\frac{t}{\Delta}\right)^t + \left(\frac{\sqrt{\mu t}}{\Delta}\right)^t + \gamma \cdot \left(\frac{2k}{\Delta}\right)^t.$$

**Proof.** See the full version of this paper [28, §4.2].  $\blacktriangleleft$

Given Corollary 22, we are ready to construct our balanced partition generator.

► **Lemma 23** (Balanced partition generator). *Let  $n, k, r \in \mathbb{N}$  and  $\delta \in (0, 1)$ . Assume  $r$  is a power of two and  $r \leq k \leq n$ . There exists an explicit  $(k, k_0, \delta)$ -balanced partition generator  $G_{\text{vars}}: \{0, 1\}^a \rightarrow [r]^n$ , where*

$$k_0 = k/r + O\left(\sqrt{k/r \cdot \log(r/\delta)} + \log(r/\delta)\right),$$

with seed length

$$a = O\left(\log(r/\delta) \cdot \log\left(2 \cdot \left\lceil \frac{rk}{\log(r/\delta)} \right\rceil\right) + \log \log n\right).$$

**Proof.** Identify  $[r]^n$  with  $\{0, 1\}^{n \log r}$ . We let  $G_{\text{vars}}$  be a  $(t \log r)$ -wise  $\gamma$ -biased generator for appropriate values  $t = \log(3r/\delta)$  and  $\gamma = \frac{\delta}{3r} \cdot \left(\frac{t}{rk}\right)^{t/2}$ . The seed length bound follows from Theorem 16. For the correctness proof, assume without loss of generality that  $|S| = k$ . Sample  $Z \in [r]^n$  using the generator. Fix any bucket  $j \in [r]$ . For each  $i \in S$ , let  $X_i$  indicate whether  $Z_i = j$ . Then for any set  $T \subseteq S$  with  $|T| \leq t$ , the value  $\prod_{i \in T} X_i$  can be expressed in terms of the underlying bits of  $Z$  as a conjunction of at most  $t \log r$  literals. Therefore, by Lemma 17, we have  $|\mathbb{E}[\prod_{i \in T} X_i] - r^{-|T|}| \leq \gamma$ . Therefore, by Corollary 22, for every  $\Delta > 0$ , we have

$$\Pr\left[\sum_{i \in S} X_i \geq k/r + \Delta\right] \leq \left(\frac{t}{\Delta}\right)^t + \left(\frac{\sqrt{kt/r}}{\Delta}\right)^t + \gamma \cdot \left(\frac{2k}{\Delta}\right)^t.$$

We choose  $\Delta = \max\{2t, 2\sqrt{kt/r}\}$ . Then we get

$$\begin{aligned} \Pr\left[\sum_{i \in S} X_i \geq k/r + \Delta\right] &\leq 2^{-t} + 2^{-t} + \gamma \cdot \left(\sqrt{\frac{rk}{t}}\right)^t \\ &\leq \frac{\delta}{3r} + \frac{\delta}{3r} + \frac{\delta}{3r} \end{aligned}$$

due to our choices of  $t$  and  $\gamma$ . The union bound over  $r$  buckets completes the proof.  $\blacktriangleleft$

For comparison, Lovett, Reingold, Trevisan, and Vadhan constructed an explicit  $(k, k_0, \delta)$ -balanced partition generator for the special case  $k = \Theta(r \cdot \log(1/\delta))$ , with  $k_0 = O(k/r)$  and seed length  $a = O(\log n + \log(r/\delta) \cdot \log(r \cdot \log(1/\delta)))$  [36]. For any  $k$ , one can also use Gopalan, Kane, and Meka's PRG for Fourier shapes [25] to construct a  $(k, k_0, \delta)$ -balanced partition generator with the same value of  $k_0$  as in Lemma 23 and with seed length  $a = \tilde{O}(\log(n/\delta))$ .

### 4.3 The full $k$ -wise probably uniform generator

**Proof of Theorem 3.** Let  $G_{\text{vars}}: \{0, 1\}^a \rightarrow [r]^n$  be the  $(k, k_0, \delta)$ -balanced partition generator from Lemma 23 with parameters  $\delta = \varepsilon/3$  and  $r = (k/\log(k/\varepsilon))^{1/3}$ , or to be more precise,  $r$  is the largest power of two that is at most  $(k/\log(k/\varepsilon))^{1/3}$ . Let  $\mathcal{G}$  be the family of generators  $g: \{0, 1\}^q \rightarrow \{0, 1\}^n$  from Lemma 19, using  $\zeta = \varepsilon/(3r)$  and using the value  $k_0$  from  $G_{\text{vars}}$ . The final generator  $G$  is defined as follows.

1. Sample a partition  $Z = (Z_1, \dots, Z_n) \in [r]^n$  using  $G_{\text{vars}}$ .
2. Sample a generator  $g \sim \mathcal{G}$ .
3. Sample seeds  $X^{(1)}, \dots, X^{(r)} \in \{0, 1\}^q$  independently and uniformly at random.
4. Output  $Y \in \{0, 1\}^n$ , where

$$Y_i = g(X^{(Z_i)})_i$$

for every  $i \in [n]$ .

To prove that this works, let  $f: \{0, 1\}^n \rightarrow \{0, 1\}$  be a conjunction of  $k$  literals, say

$$f(x) = \bigwedge_{i \in S} (x_i \oplus b_i)$$

where  $S \subseteq [n]$ ,  $|S| = k$ , and  $b_i \in \{0, 1\}$  for every  $i \in S$ . We will prove that  $\mathbb{E}[f(X)] \geq (1 - \varepsilon) \cdot 2^{-k}$ , which is sufficient by Proposition 18.

For each bucket  $j \in [r]$ , let  $B_j = Z^{-1}(j)$ . The definition of a balanced partition generator ensures that except with probability  $\varepsilon/3$  over the choice of  $Z$ , we have  $|S \cap B_j| \leq k_0$  for every  $j \in [r]$ . Let  $E_1$  be this “good” event. Fix any choice of  $Z$  such that  $E_1$  occurs.

For each  $j \in [r]$ , define  $f_j: \{0, 1\}^n \rightarrow \{0, 1\}$  by

$$f_j(x) = \bigwedge_{i \in S \cap B_j} (x_i \oplus b_i),$$

so  $f(x) = f_1(x) \wedge \dots \wedge f_r(x)$ . By Lemma 19 and the union bound over the  $r$  buckets, except with probability  $\varepsilon/3$  over the choice of  $g \sim \mathcal{G}$ , we have

$$\mathbb{E}_{x \in \{0, 1\}^q} [f_j(g(x))] \geq \left(1 - \frac{\varepsilon}{3r}\right) \cdot \mathbb{E}[f_j]$$

for every  $j \in [r]$ . Let  $E_2$  be this “good” event. Fix any choice of  $g$  such that  $E_2$  occurs.

For any such fixing of  $Z$  and  $g$ , with respect to the choice of  $X^{(1)}, \dots, X^{(r)}$  alone, we have

$$\begin{aligned} \mathbb{E}_{X^{(1)}, \dots, X^{(r)}} [f(Y)] &= \prod_{j=1}^r \mathbb{E}_{X^{(j)}} [f_j(g(X^{(j)}))] \geq \prod_{j=1}^r \left(1 - \frac{\varepsilon}{3r}\right) \cdot \mathbb{E}[f_j] \\ &= \left(1 - \frac{\varepsilon}{3r}\right)^r \cdot 2^{-k} \geq (1 - \varepsilon/3) \cdot 2^{-k}, \end{aligned}$$

by Bernoulli’s inequality. Therefore, with respect to all the randomness, we have

$$\begin{aligned} \mathbb{E}[f(Y)] &\geq \Pr[f(Y) = 1 \text{ and } E_1 \text{ and } E_2] = \Pr[E_1] \cdot \Pr[E_2 \mid E_1] \cdot \Pr[f(Y) = 1 \mid E_1, E_2] \\ &\geq (1 - \varepsilon/3) \cdot (1 - \varepsilon/3) \cdot (1 - \varepsilon/3) \cdot 2^{-k} \\ &\geq (1 - \varepsilon) \cdot 2^{-k} \end{aligned}$$

by another application of Bernoulli’s inequality.

Now let us bound the seed length. By Lemma 23, the cost of sampling  $Z$  is

$$\begin{aligned} &O\left(\log(r/\varepsilon) \cdot \log\left(2 \cdot \left\lceil \frac{rk}{\log(r/\varepsilon)} \right\rceil\right) + \log \log n\right) \\ &\leq O\left(\log(k/\varepsilon) \cdot \log\left(2 \cdot \left\lceil \frac{k}{\log(k/\varepsilon)} \right\rceil\right) + \log \log n\right) \\ &\leq O\left(\log(k/\varepsilon) \cdot \left(\frac{k}{\log(k/\varepsilon)}\right)^{2/3} + \log(k/\varepsilon) + \log \log n\right) \\ &= O(k^{2/3} \cdot \log^{1/3}(k/\varepsilon) + \log(k/\varepsilon) + \log \log n). \end{aligned}$$

## 35:16 Fooling Near-Maximal Decision Trees

Furthermore, the parameter  $k_0$  is given by

$$k_0 = k/r + O\left(\sqrt{k/r \cdot \log(r/\varepsilon)} + \log(r/\varepsilon)\right) \leq k/r + O\left(\sqrt{k/r \cdot \log(k/\varepsilon)} + \log(k/\varepsilon)\right).$$

Therefore, by Lemma 19, the cost of sampling  $g \sim \mathcal{G}$  is

$$\begin{aligned} & O(k_0 + \log(k/\varepsilon) + \log \log n) \\ &= O(k^{2/3} \cdot \log^{1/3}(k/\varepsilon) + k^{1/3} \cdot \log^{2/3}(k/\varepsilon) + \log(k/\varepsilon) + \log \log n) \\ &= O(k^{2/3} \cdot \log^{1/3}(k/\varepsilon) + \log(k/\varepsilon) + \log \log n). \end{aligned}$$

Finally, the cost of sampling  $X^{(1)}, \dots, X^{(r)}$  is

$$\begin{aligned} r \cdot q &= r \cdot k_0 + O(r \cdot \log(k/\varepsilon)) \\ &= k + O(k^{2/3} \cdot \log^{1/3}(k/\varepsilon) + k^{1/3} \log^{2/3}(k/\varepsilon) + \log(k/\varepsilon)) \\ &= k + O(k^{2/3} \cdot \log^{1/3}(k/\varepsilon) + \log(k/\varepsilon)). \end{aligned} \quad \blacktriangleleft$$

### 5 Implications of $k$ -wise probable uniformity

In this section, we will show that every  $k$ -wise probably uniform distribution fools decision trees. In fact, we will show that such distributions fool a more general model, called the *subcube partition model*.

► **Definition 24** (The subcube partition model). *A subcube partition  $f$  is a collection of terms  $f_1, \dots, f_m$  and values  $b_1, \dots, b_m \in \{0, 1\}$ . Each term  $f_i: \{0, 1\}^n \rightarrow \{0, 1\}$  is a conjunction of literals, and the sets  $f_1^{-1}(1), \dots, f_m^{-1}(1)$  must partition the domain  $\{0, 1\}^n$ . That is, for every  $x \in \{0, 1\}^n$ , we have  $\sum_{i=1}^m f_i(x) = 1$ . The subcube partition computes the function  $f: \{0, 1\}^n \rightarrow \{0, 1\}$  defined by*

$$f(x) = \sum_{i=1}^m f_i(x) \cdot b_i.$$

*The width of a term  $f_i$  is the number of literals in the term. The width of the subcube partition is the maximum width of any term. The size of the subcube partition is the number of terms ( $m$ ).*

Every width- $k$  subcube partition has size at most  $2^k$ , because  $1 = \sum_{i=1}^m \mathbb{E}[f_i] \geq m \cdot 2^{-k}$ . A decision tree of depth  $k$  and size  $m$  can be simulated by a subcube partition of width  $k$  and size  $m$ : for each leaf  $u$ , we construct a term  $f_u$  that indicates whether the tree reaches the leaf  $u$  on a given input. The converse does not hold. In fact, there exist subcube partitions of width  $k$  that cannot be simulated by decision trees of depth  $k^{1.99}$  [45, 33, 24, 8]. We now explain why  $k$ -wise probably uniform generators fool subcube partitions.

► **Lemma 25** ( $k$ -wise probable uniformity fools subcube partitions). *Let  $X$  be a distribution over  $\{0, 1\}^n$  that is  $k$ -wise  $\varepsilon$ -probably uniform. Then:*

- $X$  fools width- $k$  subcube partitions (hence also depth- $k$  decision trees) with error  $\varepsilon$ .
- $X$  fools size- $m$  subcube partitions (hence also size- $m$  decision trees) with error  $\varepsilon + m \cdot 2^{-(k+1)}$ .

**Proof.** Let  $f: \{0, 1\}^n \rightarrow \{0, 1\}$  be a function computed by a subcube partition with terms  $f_1, \dots, f_m$  and values  $b_1, \dots, b_m$ . Let  $S \subseteq [m]$  be the set of terms of width at most  $k$ . We will show that  $X$  fools  $f$  with error  $\varepsilon + \sum_{i \notin S} \mathbb{E}[f_i]$ . To prove it, sample  $R \in \{0, 1\}^n$  uniformly at random. Then

$$\begin{aligned} \mathbb{E}[f(X)] &= \sum_{i=1}^m b_i \cdot \mathbb{E}[f_i(X)] \geq \sum_{i \in S} b_i \cdot \mathbb{E}[f_i(X)] \geq \sum_{i \in S} b_i \cdot (1 - \varepsilon) \cdot \mathbb{E}[f_i] \\ &= (1 - \varepsilon) \cdot \mathbb{E} \left[ \sum_{i \in S} b_i \cdot f_i(R) \right] \geq \mathbb{E} \left[ \sum_{i \in S} b_i \cdot f_i(R) \right] - \varepsilon \\ &= \mathbb{E} \left[ f(R) - \sum_{i \notin S} b_i \cdot f_i(R) \right] - \varepsilon \geq \mathbb{E}[f] - \sum_{i \notin S} \mathbb{E}[f_i] - \varepsilon. \end{aligned}$$

Now we bound the expectation from above. Let  $\bar{f} = 1 - f$ . Since  $\bar{f}$  can also be computed by a subcube partition with the same terms  $f_1, \dots, f_m$ , we have

$$\mathbb{E}[f(X)] = 1 - \mathbb{E}[\bar{f}(X)] \leq 1 - \mathbb{E}[\bar{f}] + \varepsilon + \sum_{i \notin S} \mathbb{E}[f_i] = \mathbb{E}[f] + \varepsilon + \sum_{i \notin S} \mathbb{E}[f_i].$$

The lemma follows, because  $\mathbb{E}[f_i] \leq 2^{-(k+1)}$  whenever  $i \notin S$ . ◀

By combining Theorem 3 (our  $k$ -wise probably uniform generator) with Lemma 25, we now prove the following theorem, which generalizes Theorem 5.

► **Theorem 26** (Fooling near-maximal subcube partitions). *Let  $n, m \in \mathbb{N}$  and  $\varepsilon \in (0, 1)$ . There exists an explicit PRG  $G: \{0, 1\}^s \rightarrow \{0, 1\}^n$  that fools  $n$ -variate subcube partitions of size  $m$  with error  $\varepsilon$  and seed length*

$$s = \log m + O \left( \log^{2/3} m \cdot \log^{1/3} \left( \frac{\log m}{\varepsilon} \right) + \log(1/\varepsilon) + \log \log n \right). \quad (3)$$

**Proof.** We use our  $k$ -wise  $(\varepsilon/2)$ -probably uniform generator, where  $k = \log m + \log(2/\varepsilon)$ . By Lemma 25, the generator fools size- $m$  subcube partitions with error  $\varepsilon/2 + m \cdot 2^{-k} = \varepsilon$ . By Theorem 3, the seed length is

$$k + O(k^{2/3} \cdot \log^{1/3}(k/\varepsilon) + \log(1/\varepsilon) + \log \log n),$$

which, after substituting the choice of  $k$ , simplifies to Equation (3). ◀

## 6 Hitting sets for systems of equations over $\mathbb{F}_2$ and for $B_2$ -circuits

In this section, we present our *hitting set* for systems of equations over  $\mathbb{F}_2$ , thereby proving Theorem 6. Next, we show that such hitting sets can hit circuits over the  $B_2$  basis, thus proving Corollary 9. In the full version of this paper [28], we also present a more explicit construction of *hitting set generators* for systems of equations over  $\mathbb{F}_2$ , where given the seed, we can output the corresponding string in time  $\text{poly}(n)$ .

### 6.1 Rank condenser

First, we use a *rank condenser*, due to Forbes and Guruswami [22] to “condense” the number of variables from  $n$  to  $O(k \cdot \log n)$ .

► **Definition 27** (*k*-rank condenser). Let  $\mathbb{F}$  be a field and let  $n \geq k \geq 1$ . A collection of matrices  $\mathcal{M} \subseteq \mathbb{F}^{n \times n'}$  is a *k*-rank condenser if, for every matrix  $A \in \mathbb{F}^{k \times n}$  with  $\text{rank}(A) = k$ , there exists  $M \in \mathcal{M}$  such that  $\text{rank}(AM) = k$ .

We say that  $\mathcal{M}$  is explicit if, given an index  $i \in [|\mathcal{M}|]$ , the *i*-th matrix of  $\mathcal{M}$  can be constructed in time  $\text{poly}(n)$ .

► **Remark 28.** Stronger “lossless” variants – which bound how many matrices in  $\mathcal{M}$  can cause rank loss or how much total rank loss can occur – have been studied (see, e.g., Forbes and Guruswami [22]). The simpler notion above suffices for our purposes.

The following theorem, due to Forbes and Guruswami, shows that we can construct such condensers explicitly over  $\mathbb{F}_2$  while keeping the output dimension only  $O(k \cdot \log n)$ .

► **Theorem 29.** Let  $n \geq k \geq 1$ . There is an explicit *k*-rank condenser  $\mathcal{M} \subseteq \mathbb{F}_2^{n \times 4k \log n}$  with  $|\mathcal{M}| = \text{poly}(n)$ .

**Proof.** This follows from Forbes and Guruswami’s work [22, Corollary 8.7, preprint version], by setting the parameters appropriately. ◀

## 6.2 Partition the variables

For the sake of brevity, we define the following notation.

► **Definition 30.** Let  $H \subseteq \mathbb{F}_2^n$ . We say that  $H$  hits codimension *k* if, for every affine subspace of codimension *k*, there exists  $x \in H$  in this affine subspace. Equivalently, for every  $A \in \mathbb{F}_2^{k \times n}$  and every  $b \in \text{image}(A)$ , there exists  $x \in H$  such that  $Ax = b$ .

Our goal is to construct an  $H$  that hits codimension *k*. We can split the  $n$  variables into  $\ell$  consecutive blocks of arbitrary size. For any  $A \subseteq \mathbb{F}_2^{k \times n}$ , this induces a column partition, giving a column partition  $A = [A_1 \ A_2 \ \dots \ A_t]$ , where  $A_i \subseteq \mathbb{F}_2^{k \times n_i}$  and  $n_1 + \dots + n_t = n$ . Without loss of generality, we assume that  $A$  has full rank. Write  $k_i$  for the incremental rank contributed by block  $A_i$ , i.e.,  $k_i = \text{rank}([A_1 \ A_2 \ \dots \ A_i]) - \text{rank}([A_1 \ A_2 \ \dots \ A_{i-1}])$ , so  $k_1 + \dots + k_t = k$ . Andreev, Clementi and Rolim [10] stated the result that, if  $H_i \subseteq \mathbb{F}_2^{n_i}$  hits codimension  $k_i$  for every  $i$ , then there is some  $x$  in the Cartesian product  $H_1 \times H_2 \times \dots \times H_t$  such that  $Ax = b$ .

However, they skipped the proof, so we complete the proof in this subsection. We began by showing that after fixing the first  $i - 1$  blocks, the feasible assignments to the  $i$ -th block form an affine subspace of codimension  $k_i$ . In the following, we focus on the case of partitioning  $A$  into two blocks, which will turn out to be sufficient for analyzing the general case.

► **Lemma 31.** Let  $\mathbb{F}$  be a field. Let  $A_1 \in \mathbb{F}^{k \times n_1}$  and  $A_2 \in \mathbb{F}^{k \times n_2}$ . Let  $b \in \text{image}([A_1 \ A_2])$ , and define  $V = \{y \in \mathbb{F}^{n_1} \mid \exists z \in \mathbb{F}^{n_2} \text{ such that } A_1 y + A_2 z = b\}$ . Then  $V$  is an affine space with codimension  $\text{rank}([A_1 \ A_2]) - \text{rank}(A_2)$ .

**Proof.** Since  $b \in \text{image}([A_1 \ A_2])$ , we know there exists  $(y_*, z_*)$  such that  $A_1 y_* + A_2 z_* = b$ . Let  $W = A_1^{-1}(\text{image}(A_2))$ . We claim that  $V = W + y_*$ . Indeed, if  $y \in W + y_*$ , then  $y - y_* \in W$ , so there is some  $z$  such that  $A_1(y - y_*) = A_2 z$ . Hence

$$A_1 y + A_2(z_* - z) = A_1 y_* + A_2 z_* = b,$$

so  $y \in V$ . Conversely, if  $y \in V$ , then there is some  $z$  such that  $A_1 y + A_2 z = b$ , and consequently

$$A_1(y - y_*) = b - A_2 z - A_1 y_* = A_2(z_* - z),$$

showing that  $y - y_* \in W$ , i.e.  $y \in W + y_*$ .

Now we are going to show that  $\text{codim}(W) = \text{rank}([A_1 \ A_2]) - \text{rank}(A_2)$ . Let  $b_1, \dots, b_s$  be a basis of  $W$ . Extend this to a basis  $b_1, \dots, b_{n_1}$  of  $\mathbb{F}^{n_1}$ , and set  $U = \text{span}(b_{s+1}, \dots, b_{n_1})$ , so  $\mathbb{F}^{n_1} = U + W$ . Because  $\ker(A_1) \subseteq W$ , the map  $A_1$  is injective on  $U$ . Hence  $\dim(A_1U) = \dim(U) = \text{codim}(W)$ . On the other hand,  $\dim(A_1U) = \text{rank}([A_1 \ A_2]) - \text{rank}(A_2)$ . ◀

► **Corollary 32** ([10]). *Let  $A = [A_1 \ A_2 \ \dots \ A_t] \in \mathbb{F}^{k \times n}$  be a matrix of rank  $k$ , where each block  $A_i \in \mathbb{F}^{k \times n_i}$  and  $n_1 + \dots + n_t = n$ . For every  $i \in [t]$ , let  $k_i = \text{rank}([A_i \ A_2 \ \dots \ A_t]) - \text{rank}([A_{i+1} \ A_2 \ \dots \ A_t])$ . For each  $i \in [t]$ , let  $H_i \subseteq \mathbb{F}_2^{n_i}$ , and assume  $H_i$  hits codimension  $k_i$ . Then for every  $b \in \text{image}(A)$ , there exists a vector  $x$  in the Cartesian product*

$$H_1 \times H_2 \times \dots \times H_t \subseteq \mathbb{F}_2^n$$

such that  $Ax = b$ .

**Proof.** We prove it by induction on  $\ell$ . In the base case, when  $\ell = 1$ , the corollary follows immediately from the definition of hitting rank  $k_1$ . Now suppose  $\ell > 1$ . Define  $A_{>1} = [A_2 \ A_3 \ \dots \ A_t]$ , and define

$$V = \{y \in \mathbb{F}_2^{n_1} : \exists z \in \mathbb{F}_2^{n-n_1} \text{ such that } A_1y + A_{>1}z = b\}.$$

By Lemma 31,  $V$  is an affine space with codimension  $k_1$ . Therefore, there exists  $y \in H_1 \cap V$ . By the definition of  $V$ ,  $b - A_1y \in \text{image}(A_{>1})$ . Therefore, by induction, there exists  $z \in H_2 \times \dots \times H_t$  such that  $A_{>1}z = b - A_1y$ . Let  $x = (y, z)$ . Then  $x \in H_1 \times \dots \times H_t$ , and  $Ax = A_1y + A_{>1}z = b$ . ◀

### 6.3 Brute-force construction

There is a simple brute-force method for constructing a set that hits codimension  $k$  with the following time complexity.

► **Lemma 33** (Brute-force hitting set for systems of equations). *For every  $n, k \in \mathbb{N}$ , there exists  $H \subseteq \mathbb{F}_2^n$  of size  $2^{k+O(\log(nk))}$  that hits codimension  $k$ , which can be constructed in time  $O(nk \cdot 2^{kn+n+2k})$ .*

The proof of Lemma 33 can be found in the full version of this paper [28]. On its own, the algorithm of Lemma 33 is too slow to prove Theorem 6. However, we will only apply the brute-force method after reducing the length of binary strings we are searching, so we can afford the exponential time cost. Note the size of our construction matches that of the hitting set obtained by the standard probabilistic method. This method is similar to Naor, Schulman, and Srinivasan’s work [40].

### 6.4 Our final hitting set for systems of equations

**Proof of Theorem 6.** Without loss of generality, we assume that  $k \geq \log n$ ; otherwise, we can simply use a small-bias distribution as described in the paragraph following the statement of Theorem 6. First we use Theorem 29 to construct a  $k$ -rank condenser  $\mathcal{M} \subseteq \mathbb{F}_2^{n \times 4k \log n}$ , where  $|\mathcal{M}| = \text{poly}(n)$ . Then we partition the variables into  $t$  blocks of equal size, where  $t \approx k^{2/3}$  (the exact value will be specified later). Without loss of generality, we assume that  $n'/t$  is an integer. For each  $i \in \{0, 1, \dots, n'/t\}$ , we use Lemma 33 to construct  $H_i \subseteq \mathbb{F}_2^{n'/t}$  that hits codimension  $i$ , as defined in Definition 30. Then we combine them by taking a Cartesian product. Thus, the overall construction is

$$H = \bigcup_{\substack{k_1, \dots, k_t \in \mathbb{N} \\ k_1 + \dots + k_t = k}} \{Mx : M \in \mathcal{M} \text{ and } x \in H_{k_1} \times \dots \times H_{k_t}\}.$$

## 35:20 Fooling Near-Maximal Decision Trees

We first prove the construction is efficient (Item 2) and then prove the construction is correct (Item 1).

(Item 2). By Lemma 33, we know the size of each  $H_i$  is  $|H_i| = O(\frac{n'}{t} \cdot i2^i)$ , and the total running time to construct these hitting sets over  $n'/t$  variables is  $\sum_{i=1}^{n'/t} O(2^{n'/t+i(n'/t+2)\frac{n'}{t}i}) \leq O(2^{n'/t+(n'/t)^2+2(n'/t)(\frac{n'}{t})^3}) = 2^{O((n'/t)^2)}$ .

For one partition of  $k$ , namely  $k_1 + \dots + k_t = k$ , we have

$$\begin{aligned} |H_{k_1} \times \dots \times H_{k_t}| &\leq \prod_{i=1}^t k_i \cdot 2^{k_i + O(\log(\frac{n'}{t}))} = 2^{t \cdot O(\log(\frac{n'}{t})) + \sum_{i=1}^t k_i} \cdot \prod_{i=1}^t k_i \\ &\leq 2^{t \cdot O(\log(\frac{n'}{t}) + \log k) + k} \leq 2^{O(t \cdot \log k) + k}. \end{aligned}$$

Since each  $0 \leq k_i \leq k$ , the total number of partitions  $k_1, \dots, k_t$  is at most  $(k+1)^t = 2^{t \log(k+1)} \leq 2^{O(t \cdot \log k)}$ . Thus,

$$\begin{aligned} |H| &\leq |\mathcal{M}| \cdot 2^{t \cdot O(\log k)} \cdot 2^{O(t \cdot \log k) + k} \\ &\leq 2^{O(\log n) + O(t \cdot \log k) + k}. \end{aligned}$$

Thus, the time used by our algorithm is at most

$$2^{O(\log n + t \cdot \log k + (k \log n)^2 / t^2) + k}.$$

By choosing  $t = O\left(\frac{k^2 \log^2 n}{\log k}\right)^{1/3}$ , we get the time used by our algorithm to be

$$2^{k + O((k \log n \log k)^{2/3} + \log n)}.$$

(Item 1). Now consider any  $A \in \mathbb{F}_2^{k \times n}$  and any  $b \in \text{image}(A)$ . Without loss of generality, we assume that  $\text{rank}(A) = k$ . By Theorem 29, we know there is an  $M \in \mathcal{M}$  such that  $\text{rank} A = \text{rank} AM$ , and we denote  $A' := AM$ . Since  $\text{image}(AM) \subseteq \text{image}(A)$  and  $\dim(\text{image}(AM)) = \text{rank}(AM) = \text{rank}(A) = \dim(\text{image}(A))$ , we have  $\text{image}(AM) = \text{image}(A)$ , thus  $b \in \text{image}(AM)$  as well. By partitioning  $A'$  into  $t$  blocks of size  $k \times \frac{n'}{t}$ , we get

$$A' = [A'_1 \ A'_2 \ \dots \ A'_t].$$

Let  $k_i = \text{rank}([A'_1 \ A'_2 \ \dots \ A'_i]) - \text{rank}([A'_1 \ A'_2 \ \dots \ A'_{i-1}])$ . Thus, by Corollary 32, we know that there exists an  $x \in H_{k_1} \times \dots \times H_{k_t}$ , such that  $A'x = b$ , which means  $A(Mx) = b$  and  $Mx \in H$ .  $\blacktriangleleft$

### 6.5 Hitting set for $B_2$ -circuits

In this subsection, we will show that this hitting set can be used for  $B_2$  circuits.

► **Corollary 34** (Restatement of Corollary 9). *For every  $n \in \mathbb{N}$  and  $\alpha \in (0, 2.5)$ , there exists a value  $\varepsilon = 2^{-\Omega(\alpha^2 n)}$  and a set  $H \subseteq \{0, 1\}^n$  such that:*

1.  $H$  is an  $\varepsilon$ -hitting set for  $B_2$ -circuits of size  $(2.5 - \alpha) \cdot n$ .
2. Given the parameters  $n$  and  $\alpha$ , the set  $H$  can be enumerated in time  $2^{(1 - \Omega(\alpha^2)) \cdot n + \tilde{O}(n^{2/3})}$ .

**Proof.** Assume without loss of generality that the queries on every root-to-leaf path in  $f$  are linearly independent. First we note that any parity decision tree  $f$  can be written as  $f = f_1 + \dots + f_\ell$ , where each  $f_i$  corresponds to a path from the root to an accepting leaf

in  $f$ . Note that  $f_i$ 's are disjoint and each  $f_i$  is a conjunction of parities function. The number of parity functions in the conjunction is the depth of this leaf. If  $f$  is a size- $m$  parity decision tree with  $\mathbb{E}[f] > \varepsilon$ , then there is some accepting leaf that is reached with probability greater than  $\varepsilon/m$ . The depth of that leaf must be less than  $\log(m/\varepsilon)$ , because otherwise the probability of reaching it would be smaller. Consequently, if  $H \subseteq \mathbb{F}_2^n$  hits codimension  $\log(m/\varepsilon)$ , then  $H$  is an  $\varepsilon$ -hitting set for size- $m$  parity decision trees.

By Chen and Kabanets' work [19], we know every  $B_2$ -circuit of size  $(2.5 - \alpha) \cdot n$  can be simulated by a parity decision tree of size  $m = 2^{(1-c \cdot \alpha^2) \cdot n}$ . Let  $\varepsilon = 2^{-\frac{c}{2} \cdot \alpha^2 n}$ . Then  $\log(m/\varepsilon) = (1 - (c/2) \cdot \alpha^2)n$ . By Lemma 33, a set that hits codimension  $\log(m/\varepsilon)$  can be enumerated in time

$$2^{\log(m/\varepsilon) + O((\log(m/\varepsilon) \cdot \log \log(m/\varepsilon) \cdot \log n)^{2/3} + \log n)} = 2^{(1 - \Omega(\alpha^2)) \cdot n + \tilde{O}(n^{2/3})}. \quad \blacktriangleleft$$

► **Remark 35.** In this proof, we have used the fact hitting parity decision trees is equivalent to hitting system of equations. We further note that hitting DNF of parities is also equivalent to these two problems.

---

## References

- 1 N. Alon. Explicit construction of exponential sized families of  $k$ -independent sets. *Discrete Math.*, 58(2):191–193, 1986. doi:10.1016/0012-365X(86)90161-5.
- 2 Noga Alon. Perturbed identity matrices have high rank: proof and applications. *Combin. Probab. Comput.*, 18(1-2):3–15, 2009. doi:10.1017/S0963548307008917.
- 3 Noga Alon, Alexandr Andoni, Tali Kaufman, Kevin Matulef, Ronitt Rubinfeld, and Ning Xie. Testing  $k$ -wise and almost  $k$ -wise independence. In *Proceedings of the 39th Annual Symposium on Theory of Computing (STOC)*, pages 496–505, 2007. doi:10.1145/1250790.1250863.
- 4 Noga Alon, László Babai, and Alon Itai. A fast and simple randomized parallel algorithm for the maximal independent set problem. *J. Algorithms*, 7(4):567–583, 1986. doi:10.1016/0196-6774(86)90019-2.
- 5 Noga Alon, Jehoshua Bruck, Joseph Naor, Moni Naor, and Ron M. Roth. Construction of asymptotically good low-rate error-correcting codes through pseudo-random graphs. *IEEE Transactions on Information Theory*, 38(2):509–516, 1992. doi:10.1109/18.119713.
- 6 Noga Alon, Oded Goldreich, Johan Håstad, and René Peralta. Simple constructions of almost  $k$ -wise independent random variables. *Random Structures Algorithms*, 3(3):289–304, 1992. doi:10.1002/rsa.3240030308.
- 7 Noga Alon, Oded Goldreich, and Yishay Mansour. Almost  $k$ -wise independence versus  $k$ -wise independence. *Inform. Process. Lett.*, 88(3):107–110, 2003. doi:10.1016/S0020-0190(03)00359-4.
- 8 Andris Ambainis, Martins Kokainis, and Robin Kothari. Nearly Optimal Separations Between Communication (or Query) Complexity and Partitions. In *Proceedings of the 31st Conference on Computational Complexity (CCC)*, pages 4:1–4:14, 2016. doi:10.4230/LIPIcs.CCC.2016.4.
- 9 Alexander E. Andreev, Juri L. Baskakov, Andrea E. F. Clementi, and José D. P. Rolim. Small pseudo-random sets yield hard functions: New tight explicit lower bounds for branching programs. In *Proceedings of the 26th International Colloquium on Automata, Languages and Programming (ICALP)*, pages 179–189, 1999. preprint: <https://ecc.weizmann.ac.il/report/1997/053/>. doi:10.1007/3-540-48523-6\_15.
- 10 Alexander E. Andreev, Andrea E. F. Clementi, and José D. P. Rolim. Efficient constructions of hitting sets for systems of linear functions. In *Proceedings of the 14th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 387–398, 1997. doi:10.1007/BFb0023475.
- 11 Bernd Becker and Hans-Ulrich Simon. How robust is the  $n$ -cube? *Inform. and Comput.*, 77(2):162–178, 1988. doi:10.1016/0890-5401(88)90056-9.

- 12 Avraham Ben-Aroya and Amnon Ta-Shma. Constructing small-bias sets from algebraic-geometric codes. *Theory Comput.*, 9:253–272, 2013. doi:10.4086/toc.2013.v009a005.
- 13 Guy Blanc and Dean Doron. New Near-Linear Time Decodable Codes Closer to the GV Bound. In *Proceedings of the 37th Annual Computational Complexity Conference (CCC)*, pages 10:1–10:40, 2022. doi:10.4230/LIPIcs.CCC.2022.10.
- 14 Nader H. Bshouty. Testers and their applications [extended abstract]. In *Proceedings of the 5th Conference on Innovations in Theoretical Computer Science (ITCS)*, pages 327–351. ACM, New York, 2014. doi:10.1145/2554797.2554828.
- 15 Nader H. Bshouty. Derandomizing chernoff bound with union bound with an application to  $k$ -wise independent sets, 2016. arXiv:1608.01568.
- 16 J. Lawrence Carter and Mark N. Wegman. Universal classes of hash functions. *J. Comput. System Sci.*, 18(2):143–154, 1979. doi:10.1016/0022-0000(79)90044-8.
- 17 L. Elisa Celis, Omer Reingold, Gil Segev, and Udi Wieder. Balls and bins: smaller hash families and faster evaluation. *SIAM J. Comput.*, 42(3):1030–1050, 2013. doi:10.1137/120871626.
- 18 Ashok K. Chandra, Lawrence T. Kou, George Markowsky, and Shmuel Zaks. On sets of Boolean  $n$ -vectors with all  $k$ -projections surjective. *Acta Inform.*, 20(1):103–111, 1983. doi:10.1007/BF00264296.
- 19 Ruiwen Chen and Valentine Kabanets. Correlation bounds and #SAT algorithms for small linear-size circuits. *Theoret. Comput. Sci.*, 654:2–10, 2016. doi:10.1016/j.tcs.2016.05.005.
- 20 Kuan Cheng and William M. Hoza. Hitting sets give two-sided derandomization of small space. *Theory Comput.*, 18:Paper No. 21, 32, 2022. doi:10.4086/toc.2022.v018a021.
- 21 Benny Chor, Oded Goldreich, Johan Håstad, Joel Freidmann, Steven Rudich, and Roman Smolensky. The bit extraction problem or  $t$ -resilient functions. In *Proceedings of the 26th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 396–407, 1985. doi:10.1109/SFCS.1985.55.
- 22 Michael A. Forbes and Venkatesan Guruswami. Dimension Expanders via Rank Condensers. In Naveen Garg, Klaus Jansen, Anup Rao, and José D. P. Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2015)*, volume 40 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 800–814, Dagstuhl, Germany, 2015. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. preprint: <https://arxiv.org/abs/1411.7455>. doi:10.4230/LIPIcs.APPROX-RANDOM.2015.800.
- 23 Alexander Golovnev, Alexander S. Kulikov, Alexander V. Smal, and Suguru Tamaki. Gate elimination: circuit size lower bounds and #SAT upper bounds. *Theoret. Comput. Sci.*, 719:46–63, 2018. doi:10.1016/j.tcs.2017.11.008.
- 24 Mika Göös, Toniann Pitassi, and Thomas Watson. Deterministic communication vs. partition number. *SIAM J. Comput.*, 47(6):2435–2450, 2018. doi:10.1137/16M1059369.
- 25 Parikshit Gopalan, Daniel M. Kane, and Raghu Meka. Pseudorandomness via the discrete Fourier transform. *SIAM J. Comput.*, 47(6):2451–2487, 2018. doi:10.1137/16M1062132.
- 26 Itamar Harel, William M. Hoza, Gal Vardi, Itay Evron, Nathan Srebro, and Daniel Soudry. Provable tempered overfitting of minimal nets and typical nets, 2024. doi:10.48550/arXiv.2410.19092.
- 27 Pooya Hatami and William Hoza. Paradigms for unconditional pseudorandom generators. *Found. Trends Theor. Comput. Sci.*, 16(1-2):1–210, 2024. doi:10.1561/0400000109.
- 28 William Hoza and Zelin Lv. Fooling near-maximal decision trees. ECCV preprint TR25-003, 2025. URL: <https://eccv.weizmann.ac.il/report/2025/003/>.
- 29 Peter J. Huber. Robust estimation of a location parameter. *Ann. Math. Statist.*, 35:73–101, 1964. doi:10.1214/aoms/1177703732.
- 30 Kazuo Iwama and Hiroki Morizumi. An explicit lower bound of  $5n - o(n)$  for Boolean circuits. In *Proceedings of the 27th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, volume 2420 of *Lecture Notes in Comput. Sci.*, pages 353–364. Springer, Berlin, 2002. doi:10.1007/3-540-45687-2\_29.

- 31 Howard Karloff and Yishay Mansour. On construction of  $k$ -wise independent random variables. *Combinatorica*, 17(1):91–107, 1997. doi:10.1007/BF01196134.
- 32 Daniel J. Kleitman and Joel Spencer. Families of  $k$ -independent sets. *Discrete Math.*, 6:255–262, 1973. doi:10.1016/0012-365X(73)90098-8.
- 33 Robin Kothari, David Racicot-Desloges, and Miklos Santha. Separating decision tree complexity from subcube partition complexity. In *Proceedings of the 19th International Workshop on Randomization and Computation (RANDOM)*, pages 915–930, 2015. doi:10.4230/LIPIcs.APPROX-RANDOM.2015.915.
- 34 Eyal Kushilevitz and Yishay Mansour. Learning decision trees using the Fourier spectrum. *SIAM J. Comput.*, 22(6):1331–1348, 1993. doi:10.1137/0222080.
- 35 A. A. Lialina. On the complexity of unique circuit sat. *J Math Sci*, 247:457–466, 2020. doi:10.1007/s10958-020-04813-1.
- 36 Shachar Lovett, Omer Reingold, Luca Trevisan, and Salil Vadhan. Pseudorandom bit generators that fool modular sums. In *Proceedings of the 13th International Workshop on Randomization and Computation (RANDOM)*, pages 615–630, 2009. doi:10.1007/978-3-642-03685-9\_46.
- 37 Yishay Mansour, Noam Nisan, and Prason Tiwari. The computational complexity of universal hashing. *Theoretical Computer Science*, 107(1):121–133, 1993. doi:10.1016/0304-3975(93)90257-T.
- 38 Raghu Meka and David Zuckerman. Pseudorandom generators for polynomial threshold functions. *SIAM J. Comput.*, 42(3):1275–1301, 2013. doi:10.1137/100811623.
- 39 Joseph Naor and Moni Naor. Small-bias probability spaces: efficient constructions and applications. *SIAM J. Comput.*, 22(4):838–856, 1993. doi:10.1137/0222053.
- 40 Moni Naor, Leonard J. Schulman, and Aravind Srinivasan. Splitters and near-optimal derandomization. In *Proceedings of 36th Annual Conference on Foundations of Computer Science (FOCS)*, pages 182–191, 1995. doi:10.1109/SFCS.1995.492475.
- 41 Sergey Nurk. An  $o(2^{0.4058m})$  upper bound for circuit sat. PDMI technical report, 2009. URL: <http://www.pdmi.ras.ru/preprint/2009/09-10.html>.
- 42 Ryan O’Donnell and Yu Zhao. On Closeness to  $k$ -Wise Uniformity. In *Proceedings of the 22nd International Conference on Randomization and Computation (RANDOM)*, pages 54:1–54:19, 2018. doi:10.4230/LIPIcs.APPROX-RANDOM.2018.54.
- 43 Edward Pyne, Ran Raz, and Wei Zhan. Certified hardness vs. randomness for log-space. In *Proceedings of the 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 989–1007, 2023. doi:10.1109/FOCS57990.2023.00061.
- 44 C. Radhakrishna Rao. Factorial experiments derivable from combinatorial arrangements of arrays. *Suppl. J. Roy. Statist. Soc.*, 9:128–139, 1947. doi:10.2307/2983576.
- 45 Petr Savický. On determinism versus unambiguous nondeterminism for decision trees. ECCC preprint TR02-009, 2002. URL: <https://eccc.weizmann.ac.il/report/2002/009/>.
- 46 S. V. Savinov. Upper bound for circuit sat. Master’s thesis, St. Petersburg Academic University RAS, 2014.
- 47 Jeanette P. Schmidt, Alan Siegel, and Aravind Srinivasan. Chernoff-Hoeffding bounds for applications with limited independence. *SIAM J. Discrete Math.*, 8(2):223–250, 1995. doi:10.1137/S089548019223872X.
- 48 Gadiel Seroussi and Nader H. Bshouty. Vector sets for exhaustive testing of logic circuits. *IEEE Trans. Inform. Theory*, 34(3):513–522, 1988. doi:10.1109/18.6031.
- 49 Maciej Skorski. Tight Chernoff-Like Bounds Under Limited Independence. In *Proceedings of the 26th International Conference on Randomization and Computation (RANDOM)*, pages 15:1–15:14, 2022. doi:10.4230/LIPIcs.APPROX/RANDOM.2022.15.
- 50 Thomas Steinke, Salil Vadhan, and Andrew Wan. Pseudorandomness and Fourier-growth bounds for width-3 branching programs. *Theory Comput.*, 13:Paper No. 12, 50, 2017. doi:10.4086/toc.2017.v013a012.

## 35:24 Fooling Near-Maximal Decision Trees

- 51 Amnon Ta-Shma. Explicit, almost optimal, epsilon-balanced codes. In *Proceedings of the 49th Annual Symposium on Theory of Computing (STOC)*, pages 238–251, 2017. doi:10.1145/3055399.3055408.
- 52 Donald T. Tang and Lin S. Woo. Exhaustive test pattern generation with constant weight vectors. *IEEE Transactions on Computers*, C-32(12):1145–1150, 1983. doi:10.1109/TC.1983.1676175.
- 53 Mark N. Wegman and J. Lawrence Carter. New hash functions and their use in authentication and set equality. *J. Comput. System Sci.*, 22(3):265–279, 1981. Special issue dedicated to Michael Machtey. doi:10.1016/0022-0000(81)90033-7.