THEORY OF COMPUTING, Volume 9 (13), 2013, pp. 441–470
www.theoryofcomputing.org

SPECIAL ISSUE: APPROX-RANDOM 2012

Optimal Hitting Sets for Combinatorial Shapes*

Aditya Bhaskara Devendra Desai Srikanth Srinivasan[†]

Received November 5, 2012; Revised April 16, 2013; Published May 25, 2013

Abstract: We consider the problem of constructing explicit Hitting Sets for *combinatorial shapes*, a class of statistical tests first studied by Gopalan, Meka, Reingold, and Zuckerman (STOC 2011). These generalize many well-studied classes of tests, including symmetric functions and combinatorial rectangles. Generalizing results of Linial, Luby, Saks, and Zuckerman (Combinatorica 1997) and Rabani and Shpilka (SICOMP 2010), we construct explicit hitting sets for combinatorial shapes of size polynomial in the alphabet size, dimension, and the inverse of the error parameter. This is optimal up to polynomial factors. The best previous hitting sets came from the pseudorandom generator construction of Gopalan et al., and in particular had size that was quasipolynomial in the inverse of the error parameter.

Our construction builds on natural variants of the constructions of Linial et al. and Rabani and Shpilka. In the process, we construct fractional perfect hash families and hitting sets for combinatorial rectangles with stronger guarantees. These might be of independent interest.

ACM Classification: F.1.2, F.1.3

AMS Classification: 68Q10, 68Q15, 68R10, 68W20

Key words and phrases: derandomization, expanders, explicit construction, hitting sets, perfect hashing

*An earlier version of this paper appeared in the Proceedings of the 16th International Workshop on Randomization and Computation (RANDOM 2012), pp. 423-434, Springer 2012. The present version contains complete proofs.

[†]This work was done when the author was a postdoctoral researcher at DIMACS, Rutgers University.

 DOI: 10.4086/toc.2013.v009a013

1 Introduction

Randomness is a tool of great importance in computer science and combinatorics. The probabilistic method is highly effective both in the design of simple and efficient algorithms and in demonstrating the existence of combinatorial objects with interesting properties. But the use of randomness also comes with some disadvantages. In the setting of algorithms, introducing randomness adds to the resource requirements of the algorithm, since truly random bits are hard to come by. For combinatorial constructions, *explicit* versions of these objects often turn out to have more structure, which yields advantages beyond the mere fact of their existence (e. g., we know of explicit error-correcting codes that can be efficiently encoded and decoded, but we do not know of an analogue for random linear codes [7]). Thus, it makes sense to ask exactly how powerful probabilistic algorithms and arguments are. Can they be "derandomized," i. e., replaced by deterministic algorithms/arguments of comparable efficiency?[1] There is a long line of research that has addressed this question in various forms [22, 13, 21, 26, 19].

An important line of research in this area is the question of derandomizing randomized space-bounded algorithms. In 1979, Aleliunas et al. [1] demonstrated the power of these algorithms by showing that undirected *s*-*t* connectivity can be solved by randomized algorithms in just $O(\log n)$ space. In order to show that any randomized LOGSPACE computation could be derandomized within the same space requirements, researchers considered the problem of constructing an efficient *ε-pseudorandom generator* (ε-PRG) that would stretch a short random seed to a long pseudorandom string which would be indistinguishable (up to error ε) from strings chosen from the uniform distribution to any LOGSPACE algorithm.[2] In particular, an ε-PRG (for small constant $\varepsilon > 0$) with seed length $O(\log n)$ would allow efficient deterministic simulations of LOGSPACE randomized algorithms since a deterministic algorithm could run over all possible random seeds.

A breakthrough work of Nisan [21] took a massive step towards this goal by giving an explicit ε-PRG for $\varepsilon = 1/\mathrm{poly}(n)$ that stretches $O(\log^2 n)$ truly random bits to an n-bit pseudorandom string for LOGSPACE computations. In the two decades since, however, Nisan's result has not been improved upon at this level of generality. However, many interesting sub-cases of this class of functions have been considered as avenues for progress [23, 15, 17, 16, 14].

In this work, we consider a very natural class of functions known as *combinatorial shapes*. A Boolean function f is an (m,n)-combinatorial shape if it takes n inputs $x_1,\dots,x_n \in [m]$ and computes a symmetric function of Boolean bits y_i that depend on the membership of the inputs x_i in sets $A_i \subseteq [m]$, called *accepting sets*, associated with f. (A function of Boolean bits $y_1,\dots,y_n$ is symmetric if and only if the output depends only on the sum of the input bits.) In particular, ANDs, ORs, modular sums and majorities of subsets of the input alphabet all belong to this class. Until recently, Nisan's result gave the best known seed length for any explicit ε-PRG for this class, even when ε was a constant. In 2011, however, Gopalan et al. [11] gave an explicit ε-PRG for this class with seed length $O(\log(mn) + \log^2(1/\varepsilon))$. This seed length is optimal as a function of m and n but suboptimal as a function of ε, and for the very interesting case of $\varepsilon = 1/n^{O(1)}$, this result does not improve upon Nisan's work.

Is the setting of small error important? We think the answer is yes, for many reasons. The first deals

[1] A "deterministic argument" for the existence of a combinatorial object is one that yields an efficient deterministic algorithm for its construction.

[2] As a function of its random bits, the LOGSPACE algorithm is *read-once*: it scans its input once from left to right.

with the class of combinatorial shapes: many tests from this class accept a random input only with inverse polynomial probability (e. g., the alphabet is $\{0,1\}$ and the test accepts iff the Hamming weight of its n input bits is $n/2$); for such tests, the guarantee that a $1/n^{o(1)}$-PRG gives us is unsatisfactory. Secondly, while designing PRGs for some class of statistical tests with (say) constant error, it often is the case that one needs PRGs with much smaller error—e. g., one natural way of constructing almost-$\log n$ wise independent spaces uses PRGs that fool parity tests [20] to within inverse polynomial error. Thirdly, the reason to improve the dependence on the error is simply because we know that such PRGs exist. Indeed, a randomly chosen function that expands $O(\log n)$ bits to an n-bit string is, w.h.p., an ε-PRG for $\varepsilon = 1/\text{poly}(n)$. Derandomizing this existence proof is a basic challenge in understanding how to eliminate randomness from existence proofs. The tools we gain in solving this problem might help us in solving others of a similar flavor.

Our result Constructing optimal PRGs is usually a hard problem, but there is a well-studied weakening that we consider in this paper: constructing small ε-*hitting sets* (ε-HS). An ε-HS for a class of functions has the property that any function from that class that accepts at least an ε fraction of uniformly random strings accepts at least one of the strings in the hitting set. This is clearly a weaker guarantee than what an ε-PRG gives us. Nevertheless, in many cases, this problem turns out to be very interesting and non-trivial. In particular, a polynomial sized and LOGSPACE computable ε-HS for the class of space-bounded computations would solve the long-standing open question of whether $\mathsf{RL} = \mathsf{L}$.

Our main result is an explicit ε-HS of size $\text{poly}(mn/\varepsilon)$ for the class of combinatorial shapes, which is *optimal*, to within polynomial factors, for all errors. Here, *explicit* means that it can be constructed by a deterministic algorithm in time $\text{poly}(mn/\varepsilon)$ and space $O(\log m + \log n + \log(1/\varepsilon))$.

Theorem 1.1 (Main Result (informal))**.** *For any $m, n \in \mathbb{N}, \varepsilon > 0$, there is an explicit ε-HS for the class of combinatorial shapes of size* $\text{poly}(mn/\varepsilon)$.

Related work As far as we know, ours is the first work to specifically study the problem of constructing hitting sets for combinatorial shapes. However, there has been a substantial amount of research into both PRGs and hitting sets for many interesting subclasses of combinatorial shapes, and also some generalizations.

Naor and Naor [20] constructed ε-PRGs for parity tests of bits (alphabet size 2) with a seed length of $O(\log n + \log(1/\varepsilon))$ that is optimal up to a constant factor [4]; these results were extended by Lovett, Reingold, Trevisan, and Vadhan [16] and Meka and Zuckerman [18] to modular sums (with coefficients) and separately by Watson [27] to parity sets over a larger alphabet, though with suboptimal seed length.

Combinatorial *rectangles*, another subclass of combinatorial shapes, have also been the subject of much attention. A series of works [8, 6, 17] have constructed ε-PRGs for this class of functions: the best such PRG, due to Lu [17], has seed length $O(\log n + \log^{3/2}(1/\varepsilon))$. Linial, Luby, Saks, and Zuckerman [15] constructed optimal hitting sets for this class of tests. We build on many ideas from this work.

We also mention two more recent results that are very pertinent to our work. The first has to do with *linear threshold functions* which are weighted generalizations of threshold symmetric functions of input bits. For this class, Rabani and Shpilka [24] construct an explicit ε-HS of optimal size $\text{poly}(n/\varepsilon)$. They use a bucketing and expander walk construction to build their hitting set. Our construction uses similar ideas.

The final result that we use is the PRG for combinatorial shapes by Gopalan et al. [11] that was mentioned in the introduction. This work directly motivates our results and moreover, we use their PRG as a black-box within our construction.

2 Preliminaries

Definition 2.1 (Combinatorial shapes, rectangles, and thresholds)**.** A function $f : [m]^n \to \{0,1\}$ is an (m,n)-*combinatorial shape* if there exist sets $A_1,\ldots,A_n \subseteq [m]$ and a symmetric function $h : \{0,1\}^n \to \{0,1\}$ such that $f(x_1,\ldots,x_n) = h(1_{A_1}(x_1),\ldots,1_{A_n}(x_n))$.[3] If h is the AND function, we call f an (m,n)-*combinatorial rectangle*. If h is an unweighted threshold function, i. e., h accepts iff $\sum_i 1_{A_i}(x_i) \geq \theta$ for some $\theta \in \mathbb{N}$, then f is said to be an (m,n)-*combinatorial threshold*. We denote by $\mathsf{CShape}(m,n)$, $\mathsf{CRect}(m,n)$, and $\mathsf{CThr}(m,n)$ the class of (m,n)-combinatorial shapes, rectangles, and thresholds respectively.

Notation In many arguments, we will work with a fixed collection of accepting sets $A_1,\ldots,A_n \subseteq [m]$ that will be clear from the context. In such a scenario, for $i \in [n]$, we let $X_i = 1_{A_i}(x_i)$ and denote by X the corresponding membership vector, i. e., the bits $(X_1,\ldots,X_n)$. For $i \in [n]$, let $p_i = |A_i|/m$, $q_i = 1 - p_i$ and $w_i = p_i q_i$. Define the *weight* of a shape f as $w(f) = \sum_i w_i$. Also let $\mu(f) := \sum_i p_i$. For $\theta \in \mathbb{N}$, let $T_\theta : \{0,1\}^n \to \{0,1\}$ be the symmetric function that accepts iff the sum of its inputs is at least θ.

Definition 2.2 (Pseudorandom generators and hitting sets)**.** Let $\mathcal{F} \subseteq \{0,1\}^D$ denote a Boolean function family for some input domain D. A function $G : \{0,1\}^s \to D$ is an ε-pseudorandom generator (ε-PRG) with seed length s for a class of functions $\mathcal{F}$ if for all $f \in \mathcal{F}$,

$$\left| \Pr_{x \in_u \{0,1\}^s}[f(G(x)) = 1] - \Pr_{y \in_u D}[f(y) = 1] \right| \leq \varepsilon\,.$$

An ε-hitting set (ε-HS) for $\mathcal{F}$ is a multi-set H containing only elements from D s.t. for any $f \in \mathcal{F}$, if $\Pr_{x \in_u D}[f(x) = 1] \geq \varepsilon$, then $\exists x \in H$ such that $f(x) = 1$.

Remark 2.3. Whenever we say that there exist *explicit* families of combinatorial objects of some kind, we mean that the object can be constructed by a deterministic algorithm in time polynomial and space logarithmic in the description of the object. It will be clear from the formal descriptions of the hitting sets that they can be constructed this efficiently.

We will use the following known results in our constructions.

Theorem 2.4 (ε-PRGs for $\mathsf{CShape}(m,n)$ [11])**.** *For every $\varepsilon > 0$, there exists an explicit ε-PRG $\mathcal{G}^{m,n,\varepsilon}_{GMRZ}$: $\{0,1\}^s \to [m]^n$ for $\mathsf{CShape}(m,n)$ with seed length $s = O(\log(mn) + \log^2(1/\varepsilon))$.*

Theorem 2.5 (ε-HS for $\mathsf{CRect}(m,n)$ [15])**.** *For every $\varepsilon > 0$, there exists an explicit ε-hitting set $\mathcal{S}^{m,n,\varepsilon}_{LLSZ}$ for $\mathsf{CRect}(m,n)$ of size $\mathrm{poly}(m(\log n)/\varepsilon)$.*

[3] 1_A is the indicator function of the set A.

We will also need a stronger version of Theorem 2.5 for special cases of combinatorial rectangles. Informally, the strengthening says that if the acceptance probability of a "nice" rectangle is $> p$ for some *reasonably large* p, then a close to p fraction of the strings in the hitting set are accepting. Formally, the following is proved later in the paper.

Theorem 2.6 (Stronger HS for $\mathsf{CRect}(m,n)$)**.** *For all constants $c \geq 1$, $m = n^c$, and $\rho \leq c \log n$, there is an explicit set $\mathcal{S}_{\text{rect}}^{n,c,\rho}$ of size $n^{O_c(1)}$ such that for any $\mathcal{R} \in \mathsf{CRect}(m,n)$ which satisfies the properties:*

1. *$\mathcal{R}$ is defined by A_i, and the rejecting probabilities $q_i := (1 - |A_i|/m)$ which satisfy $\sum_i q_i \leq \rho$,*
2. $\Pr_{x \sim [m]^n}[\mathcal{R}(x) = 1] \geq p \quad (\geq 1/n^c)$

we have

$$\Pr_{x \sim \mathcal{S}_{\text{rect}}^{n,c,\rho}}[\mathcal{R}(x) = 1] \geq \frac{p}{2^{O_c(\rho)}}.$$

Recall that a distribution μ over $[m]^n$ is k-wise independent for $k \in \mathbb{N}$ if for any $S \subseteq [n]$ such that $|S| \leq k$, the marginal $\mu|_S$ is uniform over $[m]^{|S|}$. Also, $\mathcal{G} : \{0,1\}^s \to [m]^n$ is a *k-wise independent probability space over* $[m]^n$ if for uniformly randomly chosen $z \in \{0,1\}^s$, the distribution of $\mathcal{G}(z)$ is k-wise independent.

Fact 2.7 (Explicit k-wise independent spaces, [2])**.** *For any $k, m, n \in \mathbb{N}$, there is an explicit k-wise independent probability space $\mathcal{G}_{\text{k-wise}}^{m,n} : \{0,1\}^s \to [m]^n$ with $s = O(k \log(mn))$.*

We will also use the following result of Even et al. [8].

Theorem 2.8. *Fix any $m, n, k \in \mathbb{N}$. Then, if $f \in \mathsf{CRect}(m,n)$ and μ is any k-wise independent distribution over $[m]^n$, then we have*

$$\left| \Pr_{x \in [m]^n}[f(x) = 1] - \Pr_{x \sim \mu}[f(x) = 1] \right| \leq \frac{1}{2^{\Omega(k)}}.$$

Expanders Recall that a degree-D multigraph $G = (V,E)$ on N vertices is an (N,D,λ)-expander if the second largest (in absolute value) eigenvalue of its normalized adjacency matrix is at most λ. We need the expander graph to be regular in the *weighted* sense, i. e., the uniform distribution should be the graph's stationary distribution. We will use explicit expanders as a basic building block. We refer the reader to the excellent survey of Hoory, Linial, and Wigderson [12] for various related results.

Fact 2.9 (Explicit expanders [12])**.** *Given any $\lambda > 0$ and $N \in \mathbb{N}$, there is an explicit (N,D,λ)-expander where $D = (1/\lambda)^{O(1)}$.*

Expanders have found numerous applications in derandomization. A central theme in these applications is to analyze random walks on a sequence of expander graphs. Let $G_1, \ldots, G_\ell$ be a sequence of (possibly different) graphs on the *same* vertex set V. Assume G_i ($i \in [\ell]$) is an (N, D_i, λ_i)-expander. Fix any $u \in V$ and $y_1, \ldots, y_\ell \in \mathbb{N}$ such that $y_i \in [D_i]$ for each $i \in [\ell]$. Note that $(u, y_1, \ldots, y_\ell)$ naturally defines a "walk" $(v_1, \ldots, v_\ell) \in V^\ell$ as follows: v_1 is the y_1th neighbor of u in G_1 and for each $i > 1$, v_i is the y_ith neighbor of v_{i-1} in G_i. We denote by $\mathcal{W}(G_1, \ldots, G_\ell)$ the set of all tuples $(u, y_1, \ldots, y_\ell)$ as defined above. Moreover, given $w = (u, y_1, \ldots, y_\ell) \in \mathcal{W}(G_1, \ldots, G_\ell)$, we define $v_i(w)$ to be the vertex v_i defined above (we will simply use v_i if the walk w is clear from the context).

We need a variant of a result due to Alon, Feige, Wigderson, and Zuckerman [3]. The lemma as it is stated below is slightly more general than the one given in [3] but it can be obtained by using essentially the same proof and setting the parameters appropriately.

Lemma 2.10. *Let $G_1,\ldots,G_\ell$ be a sequence of graphs defined on the same vertex set V of size N. Assume that G_i is an (N,D_i,λ_i)-expander. Let $V_1,\ldots,V_\ell \subseteq V$ such that $|V_i| \geq p_i N > 0$ for each $i \in [\ell]$. Let $p_0 = 1$. Then, as long as for each $i \in [\ell]$, $\lambda_i \leq (p_i p_{i-1})/8$,*

$$\Pr_{w\in \mathcal{W}(G_1,\ldots,G_\ell)}[\forall i \in [\ell], v_i(w) \in V_i] \geq (0.75)^\ell \prod_{i\in[\ell]} p_i\,. \tag{2.1}$$

Actually, the way we have defined our walk, we do not need the graph G_1. It is there in the statement just to make the notation simpler. In our applications, it is convenient to use the following corollary.

Corollary 2.11. *Let V be a set of N elements, and let $0 < p_i < 1$ for $1 \leq i \leq \ell$ be given. There exists an explicit set of walks $\mathcal{W}$, each of length ℓ, such that for any subsets $V_1,V_2,\ldots,V_\ell$ of V, with $|V_i| \geq p_i N$, there exists a walk $w = w_1 w_2 \ldots w_\ell \in \mathcal{W}$ such that $w_i \in V_i$ for all i. Furthermore, there exist such $\mathcal{W}$ satisfying $|\mathcal{W}| \leq \mathrm{poly}(N,\prod_{i=1}^{\ell}(1/p_i))$.*

This follows from Lemma 2.10 by picking λ_i smaller than $p_i p_{i-1}/8$ for each i. By Fact 2.9, known explicit constructions of expanders require choosing degrees $D_i = 1/\lambda_i^{O(1)}$. The number of walks of length ℓ is $N \cdot \prod_{i=1}^{\ell} D_i$, which gives the bound on $\mathcal{W}$ above.

Hashing Hashing plays a vital role in all our constructions. Thus, we need explicit hash families which have several "good" properties. First, we state a lemma obtained by slightly extending part of a lemma due to Rabani and Shpilka [24], which itself builds on the work of Schmidt and Siegel [25] and Fredman, Komlós, and Szemerédi [10]. The proof appears later in the paper.

Lemma 2.12 (Perfect hash families)**.** *For any $n,t \in \mathbb{N}$, there is an explicit family of hash functions $\mathcal{H}^{n,t}_{\mathrm{perf}} \subseteq [t]^{[n]}$ of size $2^{O(t)}\mathrm{poly}(n)$ such that for any $S \subseteq [n]$ with $|S| = t$, we have*

$$\Pr_{h\in\mathcal{H}^{n,t}_{\mathrm{perf}}}[h \text{ is 1-1 on } S] \geq \frac{1}{2^{O(t)}}\,.$$

The families of functions thus constructed are called *perfect hash families*. We also need a "fractional" version of the above lemma, whose proof is similar to that of the perfect hashing lemma above and is also presented later in the paper.

Lemma 2.13 (Fractional perfect hash families)**.** *For any $n,t \in \mathbb{N}$ such that $t \leq n$, there is an explicit family of hash functions $\mathcal{H}^{n,t}_{\mathrm{frac}} \subseteq [t]^{[n]}$ of size $2^{O(t)}n^{O(1)}$ such that for any $z \in [0,1]^n$ with $\sum_{j\in[n]} z_j \geq 10t$, we have*

$$\Pr_{h\in\mathcal{H}^{n,t}_{\mathrm{frac}}}\left[\forall i \in [t], \sum_{j\in h^{-1}(i)} z_j \in [0.01M, 10M]\right] \geq \frac{1}{2^{O(t)}}\,,$$

where $M = (\sum_{j\in[n]} z_j)/t$.

3 Overview

We first show a lower bound on the size of hitting sets for combinatorial shapes. This lower bound implies that the $\mathrm{poly}(mn/\varepsilon)$ sized ε-HS we construct for the class $\mathsf{CShape}(m,n)$ is optimal up to polynomial factors.

Lemma 3.1. *For any $\varepsilon < 1/3$, any ε-hitting set for $\mathsf{CShape}(m,n)$ must have size $\Omega(\max\{m,n,1/\varepsilon\})$.*

Proof. It is already known from the result of Linial et al. [15, Proposition 4] that any ε-hitting set for even the subclass $\mathsf{CRect}(m,n)$ of $\mathsf{CShape}(m,n)$ must have size at least $\Omega(\max\{m,1/\varepsilon\})$. Thus, we only need to show a lower bound of $\Omega(n)$ for the size of any ε-hitting set for $\mathsf{CShape}(m,n)$ and that will prove the lemma. This we do by essentially showing a lower bound for the case of constructing hitting sets for parity tests over alphabet size 2 and then reducing this problem to the case of larger alphabets.

We need the following, which follows from the fact any set of less than n homogeneous linear equations over $\mathbb{F}_2$ have a non-zero solution.

Fact 3.2. *Given any $\mathcal{T} \subseteq \{0,1\}^n$ such that $|\mathcal{T}| < n$, there is a non-empty set $I \subseteq [n]$ such that for each $b \in \mathcal{T}$ we have $\bigoplus_{i\in I} b_i = 0$.*

Now fix any ε-hitting set $\mathcal{S}$ for $\mathsf{CShape}(m,n)$. We fix some $A \subseteq [m]$ such that $|A| = \lfloor m/2 \rfloor$. Now, for each non-empty $I \subseteq [n]$ we define the statistical test $F_I : [m]^n \to \{0,1\}$ as follows: $F_I(x_1,\ldots,x_n) := \bigoplus_{i\in I} 1_A(x_i)$. Note that for any I, $F_I \in \mathsf{CShape}(m,n)$ since we can write $F_I(x)$ as $f(1_{B_1}(x_1),\ldots,1_{B_n}(x_n))$ where $B_i = A$ for $i \in I$ and $\emptyset$ otherwise and $f : \{0,1\}^n \to \{0,1\}$ is the parity of its n input bits, which is of course a symmetric function.

It is easy to check that for any non-empty $I \subseteq [n]$, the function F_I accepts a random input with probability at least $\lfloor m/2 \rfloor / m \geq 1/3$. Hence, for each non-empty $I \subseteq [n]$, we have an $x \in \mathcal{S}$ such that $F_I(x) = 1$. Equivalently, if we define $\mathcal{T} := \{(1_A(x_1),\ldots,1_A(x_n)) \mid x \in \mathcal{S}\} \subseteq \{0,1\}^n$, then for every non-empty $I \subseteq [n]$, there is some $b \in \mathcal{T}$ such that $\bigoplus_{i\in I} b_i = 1$. But by Fact 3.2, this implies that $|\mathcal{T}| \geq n$. Since $|\mathcal{T}| \leq |\mathcal{S}|$, we have $|\mathcal{S}| \geq n$ as well, which completes the proof. □

We now make a standard simplifying observation that we can throughout assume that m and $1/\varepsilon$ are $n^{O(1)}$. Thus, we only need to construct hitting sets of size $n^{O(1)}$ in this case.

Lemma 3.3. *Assume that for some $c \geq 1$, and $m \leq n^c$, there is an explicit $1/n^c$-HS for $\mathsf{CShape}(m,n)$ of size $n^{O_c(1)}$. Then, for any $m,n,\in \mathbb{N}$ and $\varepsilon > 0$, there is an explicit ε-HS for $\mathsf{CShape}(m,n)$ of size $\mathrm{poly}(mn/\varepsilon)$.*

Proof. Fix $c \geq 1$ so that the assumptions of the lemma hold. Note that when $m > n^c$, we can increase the number of coordinates to $n' = m$. Now, an ε-HS for $\mathsf{CShape}(m,n')$ is also an ε-HS for $\mathsf{CShape}(m,n)$, because we can ignore the final $n'-n$ coordinates and this will not affect the hitting set property. Similarly, when $\varepsilon < 1/n^c$, we can again increase the number of coordinates to n' that satisfies $\varepsilon \geq 1/(n')^c$ and the same argument follows. In each case, by assumption we have an ε-HS of size $(n')^{O_c(1)} = \mathrm{poly}(mn/\varepsilon)$ and thus, the lemma follows. □

From now on, we will assume $m, 1/\varepsilon = n^{O(1)}$. Next, we prove an important lemma which shows how to obtain hitting sets for $\mathsf{CShape}(m,n)$ starting with hitting sets for $\mathsf{CThr}(m,n)$. This reduction crucially uses the fact that combinatorial shapes consist of only *symmetric* tests—it fails to hold, for instance, for natural "weighted" generalizations of combinatorial shapes. Hitting sets for combinatorial thresholds turn out to be easier to construct by appealing to the recent results of Gopalan et al. [11].

Lemma 3.4. *Suppose that for every $\varepsilon > 0$ there exists an explicit ε-HS for $\mathsf{CThr}(m,n)$ of size $F(m,n,1/\varepsilon)$. Then there exists an explicit ε-HS for $\mathsf{CShape}(m,n)$ of size $(n+1)\cdot F^2(m,n,(n+1)/\varepsilon)$.*

Proof. Suppose we can construct hitting sets for $\mathsf{CThr}(m,n)$ and parameter ε' of size $F(m,n,1/\varepsilon')$, for all $\varepsilon' > 0$. Now consider some $f \in \mathsf{CShape}(m,n)$, defined using sets $A_1,\dots,A_n$ and symmetric function h. Since h is symmetric, it depends only on the *number* of 1's in its input. In particular, there is a $W \subseteq [n]\cup\{0\}$ such that for $a \in \{0,1\}^n$ we have $h(a) = 1$ iff $|a| \in W$. Now if $\Pr_x[f(x) = 1] \geq \varepsilon$, there must exist a $w \in W$ such that

$$\Pr_x[|\{i \in [n] \mid 1_{A_i}(x_i) = 1\}| = w] \geq \frac{\varepsilon}{|W|} \geq \frac{\varepsilon}{n+1}\,.$$

Now consider the function $f_w^+ \in \mathsf{CThr}(m,n)$ defined by the same accepting sets $A_1,\dots,A_n$ and threshold function T_w (so f_w^+ accepts iff *at least* w of its inputs x_i satisfy $x_i \in A_i$), and the function $f_w^- \in \mathsf{CThr}(m,n)$ defined by the complement accepting sets $\overline{A_1},\dots,\overline{A_n}$ and threshold function T_{n-w} (so f_w^- accepts iff *at most* w of its inputs x_i satisfy $x_i \in A_i$). We have that *both* f_w^+ and f_w^- have accepting probability at least $\varepsilon/(n+1)$, and thus an $\varepsilon/(n+1)$-HS $\mathcal{S}$ for $\mathsf{CThr}(m,n)$ must have "accepting" elements $y,z \in [m]^n$ for f_w^- and f_w^+ respectively.

The key idea is now the following. Suppose we started with the string y and moved to string z by flipping the coordinates one at a time, i. e., the sequence of strings would be:

$$(y_1\ y_2\ \cdots\ y_n),(z_1\ y_2\ \cdots\ y_n),(z_1\ z_2\ \cdots\ y_n),\dots,(z_1\ z_2\ \cdots\ z_n)\,.$$

In this sequence the number of "accepted" indices (i. e., i for which $1_{A_i}(x_i) = 1$) changes by at most one in each "step." To start with, since y was accepting for f_w^-, the number of accepting indices was at most w, and in the end, the number is at least w (since z is accepting for f_w^+), and hence one of the strings must have precisely w accepting indices, and this string would be accepting for f!

Thus, we can construct an ε-HS for $\mathsf{CShape}(m,n)$ as follows. Let $\mathcal{S}$ denote an explicit $(\varepsilon/(n+1))$-HS for $\mathsf{CThr}(m,n)$ of size $F(m,n,(n+1)/\varepsilon)$. For any $y,z \in \mathcal{S}$, let $\mathcal{I}_{y,z}$ be the set of $n+1$ "interpolated" strings obtained above. Define $\mathcal{S}' = \bigcup_{y,z\in\mathcal{S}} \mathcal{I}_{y,z}$. As we have argued above, $\mathcal{S}'$ is an ε-HS for $\mathsf{CShape}(m,n)$. It is easy to check that $\mathcal{S}'$ has the size claimed. □

Outline of the constructions In what follows, we focus on constructing hitting sets for $\mathsf{CThr}(m,n)$. We will describe the construction of two families of hitting sets: the first is for the "high weight" case – $w(f) := \sum_i w_i \geq C\log n$ for some large constant C, and the second for the case $w(f) < C\log n$. The final hitting set is a union of the ones for the two cases.

The high weight case (Section 4.1) is conceptually simpler, and illustrates the important tools. A main tool in both cases is a "fractional" version of the perfect hashing lemma, which, though a consequence of folklore techniques, does not seem to be known in this generality (Lemma 2.13).

The proof of the low weight case is technically more involved, so we first present the solution in the special case when all the sets A_i are "small," i. e., we have $p_i \leq 1/2$ for all i (Section 4.2). This case illustrates the main techniques we use for the general low weight case. The special case uses the perfect hashing lemma (which appears, for instance in derandomization of "color coding"—a trick introduced in [5], which our proof in fact bears a resemblance to).

The general case (Section 4.3), in which p_i are arbitrary, is more technical: here we need to do a "two level" hashing. The top level is by dividing into buckets, and in each bucket we get the desired "advantage" using a generalization of hitting sets for combinatorial rectangles (which itself uses hashing: Theorem 2.6).

Finally we describe the main tools used in our construction. The stronger hitting set construction for special combinatorial rectangles is discussed in Section 5, the perfect and fractional perfect hash family constructions are discussed in Section 6, and the proof of the expander walk lemma appears in Section 7. We end with some interesting open problems.

4 Hitting sets for combinatorial thresholds

As described above, we first consider the high weight case (i. e., $w(f) \geq C \log n$ for some large absolute constant C). Next, we consider the low weight case, with an additional restriction that each of the accepting probabilities $p_i \leq 1/2$. This serves as a good starting point to explain the *general* low weight case, which we get to in Section 4.3. In each section, we outline our construction and then analyze it for a generic combinatorial threshold $f : [m]^n \to \{0,1\}$ (subject to weight constraints) defined using sets $A_1, \ldots, A_n \subseteq [m]$. The theorem we finally prove in the section is as follows.

Theorem 4.1. *For any constant $c \geq 1$, the following holds. Suppose $m, 1/\varepsilon \leq n^c$. For the class of functions $\mathsf{CThr}(m,n)$, there exists an explicit ε-hitting set of size $n^{O_c(1)}$.*

The main result of the paper, which we state formally below, follows directly from the statements of Theorem 4.1 and Lemmas 3.3 and 3.4.

Theorem 4.2. *For any $m, n \in \mathbb{N}$ and $\varepsilon > 0$, there is an explicit ε-hitting set for $\mathsf{CShape}(m,n)$ of size* $\mathrm{poly}(mn/\varepsilon)$.

4.1 High weight case

In this section we will prove the following:

Theorem 4.3. *For any $c \geq 1$, there is a $C > 0$ such that for $m, 1/\varepsilon \leq n^c$, there is an explicit ε-HS of size $n^{O_c(1)}$ for the class of functions in $\mathsf{CThr}(m,n)$ of weight at least $C \log n$.*

Fix a combinatorial threshold f where the associated accepting sets are $A_1, \ldots, A_n$ and the symmetric function is T_θ, for θ such that the probability of acceptance for independent, uniformly random inputs is at least $1/n^c$. For convenience, define $\mu := \mu(f)$, and $W := w(f)$. We have $W \geq C \log n$ for a large constant C (it needs to be *large* compared to c, as seen below).

First, recall that we denote by X the membership vector for an input $x \in [m]^n$, i. e., X denotes the bits $(X_1, \ldots, X_n) = (1_{A_1}(x_1), \ldots, 1_{A_n}(x_n))$. Since $\Pr_x[T_\theta(X) = 1] > \varepsilon$ $(\geq 1/n^c)$, by Chernoff bounds we have that $\theta \leq \mu + 2\sqrt{cW \log n}$.

Outline The main idea is the following: we first divide the indices $[n]$ into $\log n$ buckets using a hash function h (from a *fractional perfect hash family*, see Lemma 2.13). This is to ensure that the w_i get distributed somewhat uniformly. Second, we aim to obtain an *advantage* of roughly $2\sqrt{cW/\log n}$ in each of the buckets (advantage is with respect to the mean in each bucket): i. e., for each $i \in [\log n]$, we choose the indices x_j ($j \in h^{-1}(i)$) such that we get

$$\sum_{j\in h^{-1}(i)} X_j \geq \sum_{j\in h^{-1}(i)} p_j + 2\sqrt{\frac{cW}{\log n}}$$

with reasonable probability. Third, we ensure that the above happens for all buckets *simultaneously* (with probability > 0) so that the advantages add up, giving a total advantage of $2\sqrt{cW\log n}$ over the mean, which is what we intended to obtain. In the second step (i. e., in each bucket), we can prove that the desired advantage occurs with *constant* probability for *uniformly randomly and independently* chosen $x_j \in [m]$ and then derandomize this choice by the result of Gopalan et al. [11] (Theorem 2.4). Finally, in the third step, we cannot afford to use independent random bits in different buckets (this would result in a seed length of $\Theta(\log^2 n)$)—thus we need to use expander walks to save on randomness.

Construction and analysis Let us now describe the three steps in detail. We note that these steps parallel the results of Rabani and Shpilka [24].

The first step is straightforward: we pick a hash function from a perfect fractional hash family $\mathcal{H}^{n,\log n}_{\text{frac}}$. From Lemma 2.13, we obtain

Claim 4.4. *For every set of weights w, there exists an $h \in \mathcal{H}^{n,\log n}_{\text{frac}}$ such that for all $1 \leq i \leq \log n$, we have $W/(100\log n) \leq \sum_{j\in h^{-1}(i)} w_j \leq (100W)/\log n$.*

The rest of the construction is done starting with each $h \in \mathcal{H}^{n,\log n}_{\text{frac}}$. Thus for analysis, suppose that we are working with an h satisfying the inequality from the above claim. For the second step, we first prove that for independent random $x_i \in [m]$, we have a constant probability of getting an *advantage* of $2\sqrt{cW/\log n}$ over the mean in each bucket.

Lemma 4.5. *Let S be the sum of k independent random variables X_i, with $\Pr[X_i = 1] = p_i$, let $c' \geq 0$ be a constant, and let $\sum_i p_i(1-p_i) = \sigma^2$, for some σ satisfying $\sigma \geq 20e^{c'^2}$. Define $\mu := \sum_i p_i$. Then $\Pr[S > \mu + c'\sigma] \geq \alpha$, and $\Pr[S < \mu - c'\sigma] \geq \alpha$, for some constant α depending on c'.*

The proof is straightforward, but it is instructive to note that in general, a random variable (in this case, S) need not deviate "much more" (in this case, a c' factor more) than its standard deviation: we have to use the fact that S is the sum of independent random variables. This is done by an application of the Berry-Esséen theorem [9].

Proof. We recall the standard Berry-Esséen theorem [9].

Fact 4.6 (Berry-Esseen)**.** *Let $Y_1, \dots, Y_n$ be independent random variables satisfying*

$$\forall i,\, \mathbb{E}[Y_i] = 0\,, \quad \sum \mathbb{E}[Y_i^2] = \sigma^2 \quad \textit{and} \quad \forall i,\, |Y_i| \leq \beta\sigma\,.$$

Then the following error bound holds for any $t \in \mathbb{R}$,

$$\left|\Pr\left[\sum Y_i > t\right] - \Pr\left[N(0,\sigma^2) > t\right]\right| \leq \beta\,.$$

We can now apply this to $Y_i := X_i - p_i$ (so as to make $\mathbb{E}[Y_i] = 0$). Then $\mathbb{E}[Y_i^2] = p_i(1-p_i)^2 + (1-p_i)p_i^2 = p_i(1-p_i)$, thus the total variance is still $\geq \sigma^2$. Since $|Y_i| \leq 1$ for all $i \in [n]$, this means we have the condition $|Y_i| \leq \beta\sigma$ for $\beta \leq e^{-c'^2}/20$. Now for the Gaussian, a computation shows that we have $\Pr[N(0,\sigma^2) > c'\sigma] > e^{-c'^2}/10$. Thus from our bound on β, we get $\Pr[\sum Y_i > c'\sigma] > e^{-c'^2}/20$, which we pick to be α. This proves the lemma. □

Assume now that we choose $x_1, \ldots, x_n \in [m]$ independently and uniformly at random. For each bucket $i \in [\log n]$ defined by the hash function h, we let $\mu_i = \sum_{j \in h^{-1}(i)} p_j$ and $W_i = \sum_{j \in h^{-1}(i)} p_j(1-p_j) = \sum_{j \in h^{-1}(i)} w_j$. Recall that Claim 4.4 assures us that for $i \in [\log n]$, $W_i \geq W/(100 \log n) \geq C/100$. Let $X^{(i)}$ denote $\sum_{j \in h^{-1}(i)} X_j$. Then, for any $i \in [\log n]$, we have

$$\Pr\left[X^{(i)} > \mu_i + 2\sqrt{\frac{cW}{\log n}}\right] \geq \Pr\left[X^{(i)} > \mu_i + \sqrt{400c} \cdot \sqrt{W_i}\right]\,.$$

We can now apply Lemma 4.5 (with σ^2 being W_i): if C is a large enough constant so that $\sqrt{W_i} \geq \sqrt{C}/10 \geq 20e^{400c}$, then for uniformly randomly chosen $x_1, \ldots, x_n \in [m]$ and each bucket $i \in [\log n]$, we have

$$\Pr\left[X^{(i)} \geq \mu_i + 2\sqrt{cW/\log n}\right] \geq \alpha\,,$$

where $\alpha > 0$ is some fixed constant depending on c. When this event occurs for *every* bucket, we obtain $\sum_{j \in [n]} X_j \geq \mu + 2\sqrt{cW \log n} \geq \mu + \theta$. We now show how to sample such an $x \in [m]^n$ with a small number of random bits.

Let $\mathcal{G} : \{0,1\}^s \to [m]^n$ denote the PRG of Gopalan et al. [11] from Theorem 2.4 with parameters m, n, and error $\alpha/2$ i. e., $\mathcal{G}_{GMRZ}^{m,n,\alpha/2}$. Note that since α is a constant depending on c, we have $s = O_c(\log n)$. Moreover, since we know that the success probability with independent random x_j ($j \in h^{-1}(i)$) for obtaining the desired advantage is at least α, we have for any $i \in [\log n]$ and $y^{(i)}$ randomly chosen from $\{0,1\}^s$,

$$\Pr_{x^{(i)} = \mathcal{G}(y^{(i)})}\left[X^{(i)} > \mu_i + 2\sqrt{\frac{cW}{\log n}}\right] \geq \alpha/2\,.$$

This only requires seed length $O_c(\log n)$ per bucket.

Thus we are left with the third step: here for each bucket $i \in [\log n]$, we would ideally like to have (independent) seeds which generate the corresponding $x^{(i)}$ (and each of these PRGs has a seed length of $O_c(\log n)$). Since we cannot afford $O_c(\log^2 n)$ total seed length, we instead do the following: consider the PRG $\mathcal{G}$ defined above. As mentioned above, since $\alpha = \Omega_c(1)$, the seed length needed here is only $O_c(\log n)$. Let $\mathcal{S}$ be the range of $\mathcal{G}$ (viewed as a multi-set of strings: $\mathcal{S} \subseteq [m]^n$). From the above, we have that for the ith bucket, the probability $x \sim \mathcal{S}$ exceeds the threshold on indices in bucket i is at least $\alpha/2$. Now there are $\log n$ buckets, and in each bucket, the probability of "success" is at least $\alpha/2$. We can thus appeal to the "expander walk" lemma of Alon et al. [3] (see preliminaries, Lemma 2.10 and Corollary 2.11).

This means the following: we consider an explicitly constructed expander on a graph with vertices being the elements of $\mathcal{S}$, and the degree being a constant depending on α. We then perform a random walk of length $\log n$ (the number of buckets). Let $s_1, s_2, \ldots, s_{\log n}$ be the strings (from $\mathcal{S}$) we see in the walk. We form a new string in $[m]^n$ by picking values for indices in bucket i, from the string s_i. By Corollary 2.11, with non-zero probability, this will succeed for *all* $1 \le i \le \log n$, and this gives the desired advantage.

The seed length for generating the walk is $O(\log|\mathcal{S}|) + O_c(1) \cdot \log n = O_c(\log n)$. Combining (or in some sense, *composing*) this with the hashing earlier completes the construction.

4.2 Low weight case with small accepting sets

We now prove Theorem 4.1 for the case of thresholds f satisfying $w(f) = O(\log n)$. Also we will make the simplifying assumption (which we will get rid of in the next subsection) that the accepting sets of f, namely $A_1, \ldots, A_n \subseteq [m]$, are of small size.

Theorem 4.7. *Fix any $c \ge 1$. For any $m = n^c$, there exists an explicit $1/n^c$-HS $\mathcal{S}^{n,c}_{\mathrm{low},1} \subseteq [m]^n$ of size $n^{O_c(1)}$ for functions $f \in \mathsf{CThr}(m,n)$ such that $w(f) \le c \log n$ and $p_i \le 1/2$ for each $i \in [n]$.*

Let us fix a function $f(x) = T_\theta(X)$ (recall that X denotes the membership vector for x) that accepts with good probability: $\Pr_x[T_\theta(X) = 1] \ge \varepsilon$. Since $w(f) \le c \log n$ and $w_i = p_i(1-p_i) \ge p_i/2$ for each $i \in [n]$, it follows that $\mu \le 2c \log n$. Thus by a Chernoff bound and the fact that $\varepsilon = 1/n^c$, we have that $\theta \le c' \log n$ for some $c' = O_c(1)$.

Outline Suppose we fix a $1 \le \theta \le c' \log n$. The idea is to use a hash function h from a *perfect hash family* (Lemma 2.12) mapping $[n] \mapsto [\theta]$. The aim is now to obtain a contribution of 1 to the sum $\sum_j X_j$ from each bucket. By using a pairwise independent space in each bucket $B_i := h^{-1}(i)$, we get the desired contribution with probability $\mu_i = \sum_{j \in B_i} p_j$. Thus in order to succeed overall, we require $\prod_i \mu_i$ to be large (at least $1/\mathrm{poly}(n)$). By a reason similar to color coding (see [5]), this condition will turn out to be true when we bucket using a perfect hash family. As before, even when this is true, we cannot use independent hashes in each bucket, we take a hash function over $[n]$, and do an expander walk. The final twist is that in the expander walk, we cannot use a constant degree expander, because we do not have a constant probability of success in each bucket—all we know is that the product of the probabilities is at least $1/n^{c''}$. Thus we use a sequence of expanders on the same vertex set with the *product of the degrees* being a specific value. We observe that there are only polynomially many possible sequences of degrees, and this will complete the proof. We note that the last trick was implicitly used in the work of [15].

Construction Let us formally describe a hitting set construction for a fixed θ. (The final set $\mathcal{S}^{n,c}_{\mathrm{low},1}$ will be a union of these for all $1 \le \theta \le c' \log n$ along with the hitting set of [15].)

Step 1: Let $\mathcal{H}^{n,\theta}_{\mathrm{perf}} = \{h : [n] \to [\theta]\}$ be a perfect hash family as in Lemma 2.12. The size of the hash family is $2^{O(\theta)}\mathrm{poly}(n) = n^{O_{c'}(1)} = n^{O_c(1)}$. For each hash function $h \in \mathcal{H}^{n,\theta}_{\mathrm{perf}}$ divide $[n]$ into θ buckets $B_1, \ldots, B_\theta$ (so $B_i = h^{-1}(i)$).

Step 2: We will plug in a pairwise independent space in each bucket. Let $\mathcal{G}^{m,n}_{\text{2-wise}} : \{0,1\}^s \to [m]^n$ denote the generator of a pairwise independent space. Note that the seed length for any bucket is $s = O(\log n)$.

Step 3: The seed for the first bucket is chosen uniformly at random and seeds for the subsequent buckets are chosen by a walk on expanders with varying degrees. For each $i \in [\theta]$ we choose every possible η_i' such that $1/\eta_i'$ is a power of 2 and $\prod_i \eta_i' \geq 1/n^{O_c(1)}$, where the constant implicit in the $O_c(1)$ will become clear in the analysis of the construction below. There are at most poly(n) such choices for all η_i''s in total.[4] We then take a $(2^s, D_i, \lambda_i)$-expander H_i on vertices $\{0,1\}^s$ with degree $D_i = \mathrm{poly}(1/(\eta_i'\eta_{i-1}'))$ (let $\eta_0' = 1$) and $\lambda_i \leq \eta_i'\eta_{i-1}'/8$ (by Fact 2.9, such explicit expanders exist). Now for any $u \in \{0,1\}^s$, $\{y_i \in [D_i]\}_{i=1}^{\theta}$, let $(u, y_1, \ldots, y_\theta) \in \mathcal{W}(H_1, \ldots, H_\theta)$ be a θ-step walk. For all starting seeds $z_0 \in \{0,1\}^s$ and all possible $y_i \in [d_i]$, we construct the input $x \in [m]^n$ such that for all $i \in [\theta]$, we have $x|_{B_i} = \mathcal{G}^{m,n}_{\text{2-wise}}(v_i(z_0, y_1, \ldots, y_\theta))|_{B_i}$.

Size. We have $|\mathcal{S}^{n,c}_{\text{low},1}| = c' \log n \cdot n^{O_c(1)} \cdot \prod_i D_i$, where the $c' \log n$ factor is due to the choice of θ, the $n^{O_c(1)}$ factor is due to the size of the perfect hash family, the number of choices of $(\eta_1', \ldots, \eta_\theta')$, and the choice of the first seed, and an additional $n^{O(1)} \cdot \prod_i D_i$ factor is the number of expander walks. Simplifying, $|\mathcal{S}^{n,c}_{\text{low},1}| = n^{O_c(1)} \prod D_i = n^{O_c(1)} \prod_i (\eta_i')^{-O(1)} \leq n^{O_c(1)}$, where the last inequality is due to the choice of η_i''s.

Analysis We follow the outline. First, by a union bound we know that

$$\Pr_{x \sim [m]^n}[T_\theta(X) = 1] \leq \sum_{|S|=\theta} \prod_{i \in S} p_i \quad \text{and hence} \quad \sum_{|S|=\theta} \prod_{i \in S} p_i \geq \varepsilon\,.$$

Second, if we hash the indices $[n]$ into θ buckets at random and consider one S with $|S| = \theta$, the probability that the indices in S are "uniformly spread" (one into each bucket) is $1/2^{O(\theta)}$. By Lemma 2.12, this property is also true if we pick h from the explicit perfect hash family $\mathcal{H}^{n,\theta}_{\text{perf}}$.

Formally, given an $h \in \mathcal{H}^{n,\theta}_{\text{perf}}$, define $\alpha_h = \prod_{i \in [\theta]} \sum_{j \in B_i} p_j$. Over a uniform choice of h from the family $\mathcal{H}^{n,\theta}_{\text{perf}}$, we can conclude that

$$\mathop{\mathbb{E}}_h[\alpha_h] \geq \sum_{|S|=\theta} \prod_{i \in S} p_i \cdot \Pr_h[h \text{ is 1-1 on } S] \geq \frac{\varepsilon}{2^{O(\theta)}} \geq \frac{1}{n^{O_c(1)}}\,.$$

Thus there must exist an h that satisfies $\alpha_h \geq 1/n^{O_c(1)}$.

We fix such an h. For a bucket B_i, define $\eta_i = \Pr_{x \in \mathcal{G}^{m,n}_{\text{2-wise}}}[\sum_{j \in B_i} X_j \geq 1]$. Now for a moment, let us analyze the construction assuming *independently seeded* pairwise independent spaces in each bucket. Then the success probability, namely the probability that *every* bucket B_i has a non-zero $\sum_{j \in B_i} X_j$ is equal to $\prod_i \eta_i$. The following claim gives a lower bound on this probability.

Claim 4.8. *For the function h satisfying $\alpha_h \geq 1/n^{O_c(1)}$, we have $\prod_{i \in [\theta]} \eta_i \geq 1/n^{O_c(1)}$.*

Proof. For a bucket B_i, define $\mu_i = \sum_{j \in B_i} p_j$. Further, call a bucket B_i as being *good* if $\mu_i \leq 1/2$, otherwise call the bucket *bad*. For the bad buckets,

$$\prod_{B_i \text{ bad}} \mu_i \leq \prod_{B_i \text{ bad}} e^{\mu_i} = \exp\left(\sum_{B_i \textit{ bad}} \mu_i\right) \leq e^{\mu} \leq n^{O_c(1)}\,. \tag{4.1}$$

[4] This is equivalent to writing $F := O_c(1) \cdot \log n$ as a sum of θ non-negative integers, which can be done in at most $\binom{F+\theta}{\theta} \leq \mathrm{poly}(n)$ ways.

From the choice of h and the definition of α_h we have

$$\frac{1}{n^{O_c(1)}} \leq \prod_{i\in[\theta]} \mu_i = \prod_{B_i \text{ bad}} \mu_i \prod_{B_i \text{ good}} \mu_i \leq n^{O_c(1)} \prod_{B_i \text{ good}} \mu_i \Rightarrow \prod_{B_i \text{ good}} \mu_i \geq \frac{1}{n^{O_c(1)}}, \tag{4.2}$$

where we have used equation (4.1) for the second inequality.

Now let us analyze the η_i's. For a good bucket B_i, by inclusion-exclusion,

$$\eta_i = \Pr_x\left[\sum_{j\in B_i} X_j \geq 1\right] \geq \sum_{j\in B_i} p_j - \sum_{j,k\in B_i: j<k} p_j p_k \geq \mu_i - \frac{\mu_i^2}{2} \geq \frac{\mu_i}{2}. \tag{4.3}$$

For a bad bucket, $\mu_i > 1/2$. But since all p_i's are $\leq 1/2$, it is not hard to see that there must exist a non empty subset $B_i' \subset B_i$ satisfying $1/4 \leq \mu_i' := \sum_{j\in B_i'} p_j \leq 1/2$. We now can use equation (4.3) on the good bucket B_i' to get the bound on the bad bucket B_i as follows:

$$\eta_i \geq \Pr_x\left[\sum_{j\in B_i'} X_j \geq 1\right] \geq \frac{\mu_i'}{2} \geq \frac{1}{8}. \tag{4.4}$$

So finally,

$$\prod_{i\in[\theta]} \eta_i \geq \prod_{B_i \text{ bad}} \frac{1}{8} \prod_{B_i \text{ good}} \frac{\mu_i}{2} \geq \frac{1}{2^{O(\theta)}} \frac{1}{n^{O_c(1)}} = \frac{1}{n^{O_c(1)}},$$

where we have used (4.3) and (4.4) for the first inequality and (4.2) for the second inequality. □

If now the seeds for $\mathcal{G}_{\text{2-wise}}^{m,n}$ in each bucket are chosen according to the expander walk with the degrees of the expander graphs suitably related to the probability vector $(\eta_1, \dots, \eta_\theta)$, then by Lemma 2.10 the success probability becomes at least $(1/2^{O(\theta)}) \prod_i \eta_i \geq 1/n^{O_c(1)}$, using Claim 4.8 for the final inequality.

However, we do not know this probability vector and we cannot try all possible such vectors, since there are too many of them. Instead, we get a closest guess $(\eta_1', \dots, \eta_\theta')$ such that for all $i \in [\theta]$, $1/\eta_i'$ is a power of 2 and $\eta_i \geq \eta_i' \geq \eta_i/2$. Again, by Lemma 2.10 the success probability becomes at least $(1/2^{O(\theta)}) \prod_i \eta_i' \geq (1/2^{O(\theta)})^2 \prod_i \eta_i \geq 1/n^{O_c(1)}$, using Claim 4.8 for the final inequality. Note that this also tells us that it is sufficient to guess η_i' such that $\prod_i (1/\eta_i') \leq n^{O_c(1)}$.

4.3 General low weight case

The general case (where the p_i's are arbitrary) is more technical: here we need to do a "two level" hashing. The top level is by dividing into buckets, and in each bucket we get the desired "advantage" using a generalization of hitting sets for combinatorial rectangles (which itself uses hashing) from [15]. The theorem we prove for this case can be stated as follows.

Theorem 4.9. *Fix any $c \geq 1$. For any $m \leq n^c$, there exists an explicit $1/n^c$-HS $\mathcal{S}_{\text{low}}^{n,c} \subseteq [m]^n$ of size $n^{O_c(1)}$ for functions $f \in \mathsf{CThr}(m,n)$ such that $w(f) \leq c\log n$.*

Construction We describe $\mathcal{S}_{\text{low}}^{n,c}$ by demonstrating how to sample a random element x of this set. The number of possible random choices bounds $|\mathcal{S}_{\text{low}}^{n,c}|$. We define the sampling process in terms of certain constants c_i that depend on c in a way that will become clear later in the proof. Assuming this, it will be clear that $|\mathcal{S}_{\text{low}}^{n,c}| = n^{O_c(1)}$.

Step 1: Choose at random $t \in \{0, \dots, 12c\log n\}$. If $t = 0$, then we simply output a random element x of $\mathcal{S}_{LLSZ}^{m,n,1/n^{c_1}}$ for some constant c_1. The number of choices for t is $O_c(\log n)$ and if $t = 0$, the number of choices for x is $n^{O_c(1)}$. The number of choices for non-zero t are bounded subsequently.

Step 2: Choose $h \in \mathcal{H}_{\text{perf}}^{n,t}$ uniformly at random. The number of choices for h is $n^{O_c(1)} \cdot 2^{O(t)} = n^{O_c(1)}$.

Step 3: Choose at random non-negative integers $\rho_1, \dots, \rho_t$ and $a_1, \dots, a_t$ such that $\sum_i \rho_i \le c_2 \log n$ and $\sum_i a_i \le c_3 \log n$. For any constants c_2 and c_3, the number of choices for $\rho_1, \dots, \rho_t$ and $a_1, \dots, a_t$ is $n^{O_c(1)}$.

Step 4: Choose a set V such that $|V| = n^{O_c(1)} = N$ and identify V with $\mathcal{S}_{\text{rect}}^{n,c_4,\rho_i}$ for some constant $c_4 \ge 1$ and each $i \in [t]$ in some arbitrary way (we assume w.l.o.g. that the sets $\mathcal{S}_{\text{rect}}^{n,c_4,\rho_i}$ $(i \in [t])$ all have the same size). Fix a sequence of expander graphs $(G_1, \dots, G_t)$ with vertex set V where G_i is an (N, D_i, λ_i)-expander with $\lambda_i \le 1/(10 \cdot 2^{a_i} \cdot 2^{a_{i-1}})$ and $D_i = 2^{O(a_i + a_{i-1})}$, where $a_0 = 0$ (this is possible by Fact 2.9). Choose $w \in \mathcal{W}(G_1, \dots, G_t)$ uniformly at random. For each $i \in [t]$, the vertex $v_i(w) \in V$ gives us some $x^{(i)} \in \mathcal{S}_{\text{rect}}^{n,c_4,\rho_i}$. Finally, we set $x \in [m]^n$ so that $x|_{h^{-1}(i)} = x^{(i)}|_{h^{-1}(i)}$. The total number of choices in this step is bounded by $|\mathcal{W}(G_1, \dots, G_t)| \le N \cdot \prod_i D_i \le n^{O_c(1)} \cdot 2^{O(\sum_i a_i)} = n^{O_c(1)}$.

Thus, the number of random choices (and hence $|\mathcal{S}_{\text{low}}^{n,c}|$) is at most $n^{O_c(1)}$.

Analysis We will now prove Theorem 4.9. The analysis once again follows the outline of Section 4.2.

For brevity, we will denote $\mathcal{S}_{\text{low}}^{n,c}$ by $\mathcal{S}$. Fix any $A_1, \dots, A_n \subseteq [m]$ and a threshold test $f \in \mathsf{CThr}(m,n)$ such that $f(x) := T_\theta(X)$ for some $\theta \in \mathbb{N}$ (where X denotes the membership vector $(X_1, \dots, X_n)$ based on the A_i's). We assume that f has low weight and good acceptance probability on uniformly random input: that is, $w(f) \le c\log n$ and $\Pr_{x \in [m]^n}[f(x) = 1] \ge 1/n^c$. For each $i \in [n]$, let p_i denote $|A_i|/m$ and q_i denote $1 - p_i$. We call A_i small if $p_i \le 1/2$ and large otherwise. Let $U = \{i \,|\, A_i \text{ is small}\}$ and $V = [n] \setminus U$. Note that $w(f) = \sum_i p_i q_i \ge \sum_{i \in U} p_i/2 + \sum_{i \in V} q_i/2$.

Also, given $x \in [m]^n$, let $Y(x) = \sum_{i \in U} X_i$ and $\overline{Z}(x) = \sum_{i \in V}(1 - X_i) = \sum_{i \in V} 1_{\overline{A_i}}(x_i)$. We have $\sum_i X_i = Y(x) + (|V| - \overline{Z}(x))$ for any x. We would like to show that $\Pr_{x \in \mathcal{S}}[f(x) = 1] > 0$. Instead we show the following stronger statement:

$$\Pr_{x \in \mathcal{S}}\left[\overline{Z}(x) = 0 \wedge Y(x) \ge \theta - |V|\right] > 0\,. \tag{4.5}$$

To do this, we first need the following simple claim.

Claim 4.10. $\Pr_{x \in [m]^n}\left[\overline{Z}(x) = 0 \wedge Y(x) \ge \theta - |V|\right] \ge 1/n^{c_1}$, *for* $c_1 = O(c)$.

Proof. Clearly, we have

$$\Pr_{x \in [m]^n}\left[\overline{Z}(x) = 0 \wedge Y(x) \ge \theta - |V|\right] = \Pr_{x \in [m]^n}\left[\overline{Z}(x) = 0\right] \cdot \Pr_{x \in [m]^n}\left[Y(x) \ge \theta - |V|\right].$$

We lower bound each term separately by $1/n^{O(c)}$.

To bound the first term, note that $\Pr_{x\in[m]^n}\left[\overline{Z}(x)=0\right]=\prod_{i\in V}(1-q_i)=\exp\{-O(\sum_{i\in V}q_i)\}$ where the last inequality follows from the fact that $q_i<1/2$ for each $i\in V$ and $(1-x)\geq e^{-2x}$ for $x\in[0,1/2]$. Now, since each $q_i<1/2$, we have $q_i\leq 2w_i$ for each $i\in V$ and hence, $\sum_{i\in V}q_i=O(w(f))=O(c\log n)$. The lower bound on the first term follows.

To bound the second term, we note that $\Pr_{x\in[m]^n}[Y(x)\geq\theta']$ can only decrease as θ' increases. Thus, we have

$$\begin{aligned}\Pr_{x\in[m]^n}[Y(x)\geq\theta-|V|]&=\sum_{i\geq 0}\Pr_{x\in[m]^n}[Y(x)\geq\theta-|V|]\cdot\Pr_{x\in[m]^n}\left[\overline{Z}(x)=i\right]\\&\geq\sum_{i\geq 0}\Pr_{x\in[m]^n}\left[Y(x)\geq(\theta-|V|+i)\wedge\overline{Z}(x)=i\right]\\&=\Pr_{x\in[m]^n}\left[\sum_{i\in[n]}X_i\geq\theta\right]\geq 1/n^c\,.\end{aligned}$$ □

To show that $\Pr_{x\in\mathcal{S}}\left[\overline{Z}(x)=0\wedge Y(x)\geq\theta-|V|\right]>0$, we define a sequence of "good" events whose conjunction occurs with positive probability and which together imply that $\overline{Z}(x)=0$ and $Y(x)\geq\theta-|V|$.

Event $\mathcal{E}_1$: $t=\max\{\theta-|V|,0\}$. To argue that $\mathcal{E}_1$ occurs with positive probability, we need to show that $\theta-|V|\leq 12c\log n$. To see this, note that we are given that $f(x)=T_\theta(X)$ accepts a uniformly random x with probability at least $1/n^c$, and by Chernoff bounds, we must have $\theta-\mathbb{E}_x[\sum_i X_i]\leq 10c\log n$. Since $\mathbb{E}_x[\sum_i X_i]\leq\sum_{i\in U}p_i+\sum_{i\in V}p_i\leq 2w(f)+|V|$, we see that $\theta-|V|\leq 12c\log n$. We condition on this choice of t.

Note that by Claim 4.10, we have $\Pr_{x\in[m]^n}\left[\overline{Z}(x)=0\wedge Y(x)\geq t\right]\geq 1/n^{c_1}$. We will show that this event occurs with positive probability even if x is drawn from $\mathcal{S}$ as described above, and this will prove (4.5). If $t=0$, then the condition that $Y(x)\geq t$ is trivial and hence the above event reduces to $\overline{Z}(x)=0$, which is just a combinatorial rectangle and hence, there is an $x\in\mathcal{S}_{LLSZ}^{m,n,1/n^{c_1}}$ with $f(x)=1$ and we are done. Therefore, for the rest of the proof we assume that $t\geq 1$.

Event $\mathcal{E}_2$: Given $h\in\mathcal{H}_{\text{perf}}^{n,t}$, define α_h to be the quantity $\prod_{i\in[t]}\left(\sum_{j\in h^{-1}(i)\cap U}p_j\right)$. Note that by Lemma 2.12, for large enough constant c_1' depending on c, we have

$$\begin{aligned}\mathop{\mathbb{E}}_{h\in\mathcal{H}_{\text{perf}}^{n,t}}[\alpha_h]&\geq\sum_{T\subseteq U:|T|=t}\prod_{j\in T}p_j\Pr_h[h\text{ is 1-1 on }T]\\&\geq\frac{1}{2^{O(t)}}\sum_{T\subseteq U:|T|=t}\prod_{j\in T}p_j\\&\geq\frac{1}{2^{O(t)}}\Pr_x[Y(x)\geq t]\qquad\text{(by union bound)}\\&\geq\frac{1}{n^{c_1'}}\,.\end{aligned}$$

Event $\mathcal{E}_2$ is simply that $\alpha_h\geq 1/n^{c_1'}$. By averaging, there is such a choice of h. Fix such a choice.

Event $\mathcal{E}_3$: We say that this event occurs if for each $i\in[t]$, we have

$$\rho_i=\left\lceil\sum_{j\in h^{-1}(i)\cap U}p_j+\sum_{k\in h^{-1}(i)\cap V}q_k\right\rceil+1\,.$$

To see that this event can occur, we only need to verify that for this choice of ρ_i, we have $\sum_i \rho_i \leq c_2 \log n$ for a suitable constant c_2 depending on c. But this straight away follows from the fact that $\sum_{j\in U} p_j + \sum_{k\in V} q_k \leq 2w(f) \leq 2c\log n$. Fix this choice of ρ_i ($i \in [t]$).

To show that there is an $x \in \mathcal{S}$ such that $\overline{Z}(x) = 0$ and $Y(x) \geq t$, our aim is to show that there is an $x \in \mathcal{S}$ with $\overline{Z}_i(x) := \sum_{j\in h^{-1}(i)\cap V}(1-X_j) = 0$ and $Y_i(x) := \sum_{j\in h^{-1}(i)\cap U} X_j \geq 1$ for each $i \in [t]$. To show that this occurs, we first need the following claim.

Claim 4.11. *Fix $i \in [t]$. Let $p'_i = \Pr_{x\in\mathcal{S}^{n,c_4,\rho_i}_{\mathrm{rect}}}\left[\overline{Z}_i(x) = 0 \wedge Y_i(x) \geq 1\right]$. Then, $p'_i \geq (\sum_{j\in h^{-1}(i)\cap U} p_j)/2^{c'_4\rho_i}$, for large enough constants c_4 and c'_4 depending on c.*

Proof. We assume that $p_j > 0$ for every $j \in h^{-1}(i)\cap U$ (the other j do not contribute anything to the right hand side of the inequality above).

The claim follows from the fact that the event $\overline{Z}_i(x) = 0 \wedge Y_i(x) \geq 1$ is implied by any of the *pairwise disjoint* rectangles $R_j(x) = X_j \wedge \bigwedge_{j\neq k\in h^{-1}(i)\cap U}(1-X_k) \wedge \bigwedge_{\ell\in h^{-1}(i)\cap V} X_\ell$ for $j \in h^{-1}(i)\cap U$. Thus, we have

$$p'_i = \Pr_{x\in\mathcal{S}^{n,c_4,\rho_i}_{\mathrm{rect}}}\left[\overline{Z}_i(x) = 0 \wedge Y_i(x) \geq 1\right] \geq \sum_{j\in h^{-1}(i)\cap U} \Pr_{x\in\mathcal{S}^{n,c_4,\rho_i}_{\mathrm{rect}}}\left[R_j(x) = 1\right]. \tag{4.6}$$

Note that the sum of the rejecting probabilities of the individual sets in the combinatorial rectangle R_j is upper bounded by $\sum_{k\in h^{-1}(i)\cap U\setminus\{j\}} p_k + \sum_{\ell\in h^{-1}(i)\cap V} q_\ell + q_j$, which is in turn is at most $\sum_{k\in h^{-1}(i)\cap U} p_k + \sum_{\ell\in h^{-1}(i)\cap V} q_\ell + 1 \leq \rho_i$ by the fact that event $\mathcal{E}_3$ holds. Moreover, $\rho_i \leq \sum_{s\in[t]} \rho_s \leq c_2 \log n$. Below, we choose $c_4 \geq c_2$ and so we have $\rho_i \leq c_4 \log n$.

Note also that for each $j \in h^{-1}(i)\cap U$, we have

$$\begin{aligned} P_j &:= \Pr_{x\in[m]^n}\left[R_j(x) = 1\right] \\ &\geq p_j \prod_{k\in h^{-1}(i)\cap U}(1-p_k) \prod_{\ell\in h^{-1}(i)\cap V}(1-q_\ell) \\ &\geq p_j \exp\{-2(\sum_k p_k + \sum_\ell q_\ell)\} \geq p_j \exp\{-2\rho_i\}, \end{aligned}$$

where the second inequality follows from the fact that $(1-x) \geq e^{-2x}$ for any $x \in [0, 1/2]$. In particular, for large enough constant $c_4 > c_2$, we see that $P_j \geq 1/m \cdot 1/n^{O(c)} \geq 1/n^{c_4}$.

Thus, by Theorem 2.6, we have for each j, $\Pr_{x\in\mathcal{S}^{n,c_4,\rho_i}_{\mathrm{rect}}}\left[R_j(x) = 1\right] \geq P_j/2^{O_c(\rho_i)}$; since $P_j \geq p_j/2^{O(\rho_i)}$, we have

$$\Pr_{x\in\mathcal{S}^{n,c_4,\rho_i}_{\mathrm{rect}}}\left[R_j(x) = 1\right] \geq p_j/2^{(O_c(1)+O(1))\rho_i} \geq p_j/2^{c'_4\rho_i}$$

for a large enough constant c'_4 depending on c. This bound, together with (4.6), proves the claim. □

The above claim immediately shows that if we plug in *independent* $x^{(i)}$ chosen at random from $\mathcal{S}^{n,c_4,\rho_i}_{\mathrm{rect}}$ in the indices in $h^{-1}(i)$, then the probability that we pick an x such that $\overline{Z}(x) = 0$ and $Y(x) \geq t$ is at least

$$\begin{aligned} \prod_i p'_i &\geq 1/2^{O_c(\sum_{i\in[t]}\rho_i)} \prod_{i\in[t]}\left(\sum_{j\in h^{-1}(i)\cap U} p_j\right) \\ &= 1/2^{O_c(\log n)} \cdot \alpha_h \geq 1/n^{O_c(1)}. \end{aligned} \tag{4.7}$$

However, the $x^{(i)}$ we actually choose are not independent but picked according to a random walk $w \in \mathcal{W}(G_1, \ldots, G_t)$. But by Lemma 2.10, we see that for this event to occur with positive probability, it suffices to have $\lambda_i \leq p'_{i-1} p'_i / 8$ for each $i \in [t]$ (let $p'_0 = 1$). To satisfy this, it suffices to have $1/2^{a_i} \leq p'_i \leq 1/2^{a_i - 1}$ for each i. This is exactly the definition of the event $\mathcal{E}_4$.

Event $\mathcal{E}_4$: For each $i \in [t]$, we have $1/2^{a_i} \leq p'_i \leq 1/2^{a_i-1}$. For this to occur with positive probability, we only need to check that $\sum_{i \in [t]} \lceil \log(1/p'_i) \rceil \leq c_3 \log n$ for large enough constant c_3. But from (4.7), we have

$$\begin{aligned}\sum_i \lceil \log(1/p'_i) \rceil &\leq \left(\sum_i \log(1/p'_i) \right) + t \\ &\leq O_c(\log n) + O(c \log n) \leq c_3 \log n\end{aligned}$$

for large enough constant c_3 depending on c. This shows that $\mathcal{E}_4$ occurs with positive probability and concludes the analysis.

Proof of Theorem 4.1 The theorem follows easily from Theorems 4.3 and 4.9. Fix constant $c \geq 1$ such that $m, 1/\varepsilon \leq n^c$. For $C > 0$ a constant depending on c, we obtain hitting sets for thresholds of weight at least $C \log n$ from Theorem 4.3 and for thresholds of weight at most $C \log n$ from Theorem 4.9. Their union is an ε-HS for all of $\mathsf{CThr}(m,n)$.

5 Stronger hitting sets for combinatorial rectangles

As mentioned in the introduction, [15] give ε-hitting set constructions for combinatorial rectangles, even for $\varepsilon = 1/\mathrm{poly}(n)$. However in our applications, we require something slightly stronger—in particular, we need a set $\mathcal{S}$ such that $\Pr_{x \sim \mathcal{S}}(x \text{ in the rectangle}) \geq \varepsilon$ (roughly speaking). We however need to fool only special kinds of rectangles, given by the two conditions in the following theorem.

Theorem 5.1 (Theorem 2.6 restated). *For all constants $c \geq 1$, $m = n^c$, and $\rho \leq c \log n$, there is an explicit set $\mathcal{S}_{\mathrm{rect}}^{n,c,\rho}$ of size $n^{O_c(1)}$ such that for any $\mathcal{R} \in \mathsf{CRect}(m,n)$ which satisfies the properties:*

1. *$\mathcal{R}$ is defined by A_i, and the rejecting probabilities q_i satisfy $\sum_i q_i \leq \rho$ and*
2. *$p := \Pr_{x \sim [m]^n}[\mathcal{R}(x) = 1] \geq 1/n^c$,*

we have

$$\Pr_{x \sim \mathcal{S}_{\mathrm{rect}}^{n,c,\rho}}[\mathcal{R}(x) = 1] \geq \frac{p}{2^{O_c(\rho)}} .$$

To outline the construction, we keep in mind a rectangle $\mathcal{R}$ defined by sets A_i, and write $p_i = |A_i|/m$, $q_i = 1 - p_i$. W.l.o.g., we assume that $\rho \geq 10$. The outline of the construction is as follows:

1. We guess an integer $r \leq \rho/10$ (supposed to be an estimate for $\sum_i q_i/10$).
2. Then we use a fractional hash family $\mathcal{H}_{\mathrm{frac}}^{n,r}$ to map the indices into r buckets. This ensures that each bucket has roughly a constant weight.

3. In each bucket, we show that taking $O(1)$-wise independent spaces (Fact 2.7) ensures a success probability (i. e., the probability of being inside $\mathcal{R}$) depending on the weight of the bucket.

4. We then combine the distributions for different buckets using expander walks (this step has to be done with more care now, since the probabilities are different across buckets).

Steps (1) and (2) are simple: we try all choices of r, and the "right" one for the hashing in step (2) to work is $r = \sum_i q_i/10$; the probability that we make this correct guess is at least $1/\rho \gg 1/2^{\rho}$. In this case, by the fractional hashing lemma, we obtain a hash family $\mathcal{H}_{\text{frac}}^{n,r}$, which has the property that for an h drawn from it, we have

$$\Pr\left[\sum_{j\in h^{-1}(i)} q_j \in [1/100, 100] \text{ for all } i\right] \geq \frac{1}{2^{O_c(r)}} \geq \frac{1}{2^{O_c(\rho)}}.$$

Step (3) is crucial, and we prove the following:

Claim 5.2. *There is an absolute constant $a \in \mathbb{N}$ such that the following holds. Let $A_1, \dots, A_k$ be the accepting sets of a combinatorial rectangle $\mathcal{R}$ in $\mathsf{CRect}(m,k)$, and let $q_1, \dots, q_k$ be* rejecting *probabilities as defined earlier, with $\sum_i q_i \leq C$, for some constant $C \geq 1$. Let $\mathcal{S}$ be the support of an aC-wise independent distribution on $[m]^k$ (in the sense of Fact 2.7). Then*

$$\Pr_{x\in\mathcal{S}}[\mathcal{R}(x) = 1] \geq \frac{\prod_i (1-q_i)}{2}.$$

Proof. We observe that if $\sum_i q_i \leq C$, then at most $2C$ of the q_i are $\geq 1/2$. Let B denote the set of such indices. Now consider $\mathcal{S}$, an aC-wise independent distribution over $[m]^k$. Let us restrict to the vectors in the distribution for which the coordinates corresponding to B are in the rectangle $\mathcal{R}$. Because the family is aC-wise independent, the number of such vectors is precisely a factor $\prod_{i\in B}(1-q_i)$ of the support of $\mathcal{S}$.

Now, even after fixing the values at the locations indexed by B, the chosen vectors still form a $(a-2)C$-wise independent distribution. Thus by Theorem 2.8, we have that the distribution δ-approximates, i. e., maintains the probability of any event (in particular the event that we are in the rectangle $\mathcal{R}$) to an additive error of $\delta = 2^{-\Omega((a-2)C)} < (1/2)e^{-2\sum_i q_i} < (1/2)\prod_{i\notin B}(1-q_i)$ for large enough a. (In the last step, we used the fact that if $x < 1/2$, then $(1-x) > e^{-2x}$). Thus if we restrict to coordinates outside B, we have that the probability that these indices are "accepting" for $\mathcal{R}$ is at least $(1/2)\prod_{i\notin B} p_i$ (because we have a very good additive approximation).

Combining the two, we get that the overall accepting probability is $(1/2)\prod_i(1-q_i)$, finishing the proof of the claim. □

Let us now see how the claim fits into the argument. Let $B_1, \dots, B_r$ be the sets of indices of the buckets obtained in Step (2). Claim 5.2 now implies that if we pick an aC-wise independent family on all the n positions (call this $\mathcal{S}$), the probability that we "succeed" on B_i is at least $(1/2)\prod_{j\in B_i}(1-q_j)$. For convenience, let us write $P_i = (1/2)\prod_{j\in B_i}(1-q_j)$. We wish to use an expander walk argument as before—however this time the probabilities P_i of success are different across the buckets.

The idea is to estimate P_i for each i, up to a *sufficiently small* error. Let us define $L = \lceil c\log n\rceil$. Note that $L \geq \log(1/p)$, since $p \geq 1/n^c$ (where p is as in the statement of Theorem 2.6). Now, we estimate

$\log(1/P_i)$ by the smallest integer multiple of $L' := \lfloor L/r \rfloor \geq 10$ which is larger than it: call it $\alpha_i \cdot L'$. Since $\sum_i \log(1/P_i)$ is at most L, we have $\sum_i \alpha_i L' \leq 2L$, or $\sum_i \alpha_i \leq 3r$. Since the sum is over r indices, there are at most $2^{O(r)}$ choices for the α_i we need to consider. Each choice of the α_i's gives an estimate for P_i (which is also a *lower bound* on P_i). More formally, set $\rho_i = e^{-\alpha_i L'}$, so we have $P_i \geq \rho_i$ for all i.

Finally, let us construct graphs G_i (for $1 \leq i \leq r$) with the vertex set being $\mathcal{S}$ (the aC-wise independent family), and G_i having a degree depending on ρ_i (we do this for each choice of the ρ_i's). By the expander walk lemma (Lemma 2.10), we obtain an overall probability of success of at least $\prod_i P_i/2^{O(r)}$ for the "right" choice of the ρ_i's. Since our choice is right with probability at least $2^{-O(r)}$, we obtain a success probability in Steps (3) and (4) of at least $\prod_i P_i/2^{O(r)} \geq p/2^{O(r)} \geq p/2^{O(\rho)}$. In combination with the success probability of $1/2^{O_c(\rho)}$ above for Steps (1) and (2), this gives us the claimed overall success probability.

Finally, we note that the total seed length we have used in the process is $O_c(\log n + \sum_i \log(1/\rho_i))$, which can be upper bounded by $O_c(\log n + L) = O_c(\log n)$.

6 Perfect and fractional hash families

The first step in all of our constructions has been hashing into a smaller number of buckets. To this effect, we need an explicit construction of hash families which have several "good" properties. We will first look at the perfect hash lemma, which appears in a slightly different form in Rabani and Shpilka [24]. Most of our proof follows along the lines of their proof, except for when we use expanders to derandomize the seeds used for second level hashes. The proof is provided for completeness and also serves as a warm up for the similar, but more involved construction of *fractional* perfect hash families later discussed.

Lemma 2.12 (restated)**.** For any $n, t \in \mathbb{N}$, there is an explicit family of hash functions $\mathcal{H}^{n,t}_{\text{perf}} \subseteq [t]^{[n]}$ of size $2^{O(t)}\text{poly}(n)$ such that for any $S \subseteq [n]$ with $|S| = t$, we have

$$\Pr_{h \in \mathcal{H}^{n,t}_{\text{perf}}}[h \text{ is 1-1 on } S] \geq \frac{1}{2^{O(t)}}.$$

Proof. We begin with the formal construction of $\mathcal{H}^{n,t}_{\text{perf}}$. To sample a random $h \in \mathcal{H}^{n,t}_{\text{perf}}$, we do the following:

Step 1 (Top-level hashing): We choose a pairwise independent hash function $h_1 : [n] \to [t]$ by choosing a random seed to generator $\mathcal{G}^{t,n}_{\text{2-wise}}$. By Fact 2.7, this requires $O(\log n + \log t) = O(\log n)$ bits.

Step 2 (Guessing bucket sizes): We choose at random $y_1, \ldots, y_t \in \mathbb{N}$ so that $\sum_i y_i \leq 4t$. It can be checked that the number of possibilities for $y_1, \ldots, y_t$ is only $2^{O(t)}$.

Step 3 (Second-level hashing): For each $i \in [t]$, we fix an explicit pairwise independent family of hash functions mapping $[n]$ to $[y_i]$ given by $\mathcal{G}^{y_i,n}_{\text{2-wise}}$. We assume w.l.o.g. that each such generator has some fixed seed length $s = O(\log n)$ (if not, increase the seed length of each to the maximum seed length among them). Let $V = \{0,1\}^s$. Using Fact 2.9, fix a sequence $(G_1, \ldots, G_t)$ of t many $(2^s, D, \lambda)$-expanders on set V with $D = O(1)$ and $\lambda \leq 1/100$. Choosing $w \in \mathcal{W}(G_1, \ldots, G_t)$ uniformly at random, set $h_{2,i} : [n] \to [y_i]$ to be $\mathcal{G}^{y_i,n}_{\text{2-wise}}(v_i(w))$. Define $h_2 : [n] \to [4t]$ as follows:

$$h_2(j) = \left(\sum_{i < h_1(j)} y_i\right) + h_{2,h_1(j)}(j)\,. \tag{6.1}$$

Given the random choices made in the previous steps, the function h_2 is completely determined by $|\mathcal{W}(G_1,\ldots,G_{10t})|$, which is $2^{O(t)}\cdot\mathrm{poly}(n)$.

Step 4 (Folding): We choose uniformly at random $I'\subseteq[4t]$ such that $|I'|=t$. We now fix an arbitrary map $f:[4t]\to[t]$ such that f is a bijection on I' and define $h(j):=f(h_2(j))$. The number of choices in this step is the number of possibilities for I' which is $2^{O(t)}$.

Since the number of possibilities for the random choices made in the above four steps, is bounded by $2^{O(t)}n^{O(1)}$, we see that $|\mathcal{H}^{n,t}_{\mathrm{perf}}|$ is at most $2^{O(t)}n^{O(1)}$.

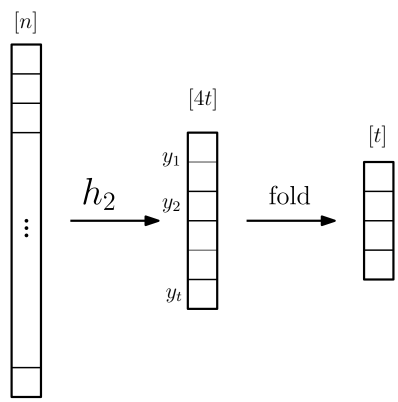

Figure 1: The basic framework of the perfect hash family construction.

We now show that a random $h\in\mathcal{H}^{n,t}_{\mathrm{perf}}$ has the properties stated in the lemma. Assume h is sampled as above. Fix $S\subseteq[n]$ such that $|S|=t$.

For $i\in[t]$, define the random variable $X_i=|h_1^{-1}(i)\cap S|$ and let $X=\sum_{i\in[t]}X_i^2$. An easy computation shows that $\mathbb{E}_{h_1}[X]\le 2t$. Let $\mathcal{E}_1$ denote the event that $X\le 4t$. By Markov's inequality, $\Pr_{h_1}[\mathcal{E}_1]\ge 1/2$. We condition on a choice of h_1 so that $\mathcal{E}_1$ occurs.

We now analyze the second step. We say that event $\mathcal{E}_2$ holds if for each $i\in[t]$, $y_i=X_i^2$. We claim that $\Pr[\mathcal{E}_2]\ge 1/2^{O(t)}$. Since the number of random choices in Step 2 is only $2^{O(t)}$, it suffices to argue that $\sum_{i\in[t]}X_i^2\le 4t$. But this follows since we have conditioned on $\mathcal{E}_1$. We now condition on random choices in Step 2 so that $\mathcal{E}_2$ occurs as well.

For the third step, given $i\in[t]$, we call the hash function $h_{2,i}$ *collision-free* if $h_{2,i}$ is 1-1 on the set $h_1^{-1}(i)\cap S$. Since $h_{2,i}$ is chosen from a pairwise independent hash family, given any distinct $j_1,j_2\in h_1^{-1}(i)\cap S$, the probability that $h_{2,i}(j_1)=h_{2,i}(j_2)$ is $1/y_i=1/X_i^2$. By a simple union bound, we can upper bound the probability that $h_{2,i}$ is not collision-free as follows:

$$\begin{aligned}\Pr_{h_{2,i}}[h_{2,i}\text{ not collision-free}] &= \Pr_{h_{2,i}}\left[\exists j_1\ne j_2\in h_1^{-1}(i)\cap S:\ h_{2,i}(j_1)=h_{2,i}(j_2)\right]\\ &\le \binom{X_i}{2}\cdot\frac{1}{X_i^2}\le\frac{1}{2}\,.\end{aligned}$$

Let $\mathcal{E}_3$ denote the event that for each $i\in[t]$, $h_{2,i}$ is collision-free. Since the seeds for the various hash functions have been chosen using a suitable expander walk, by Lemma 2.10, we see that

$$\Pr[\mathcal{E}_3] = \Pr_{w \in \mathcal{W}(G_1,\ldots,G_t)} [\forall i \in [t]: \ h_{2,i} \text{ collision-free}] \geq 1/2^{O(t)}.$$

We condition on a choice for the hash functions $h_{2,1},\ldots,h_{2,t}$ so that $\mathcal{E}_3$ also occurs. Note that given h_1 and $h_{2,i}$ for $i \in [t]$, the hash function $h_2 : [n] \to [4t]$ is completely determined.

Moreover, conditioned on the events $\mathcal{E}_1, \mathcal{E}_2$, and $\mathcal{E}_3$, we claim that h_2 is 1-1 on S. To see this, consider any distinct $j_1, j_2 \in S$. If $h_1(j_1) \neq h_1(j_2)$, then we have $h_2(j_1) \neq h_2(j_2)$ straight away from (6.1). On the other hand, if $h_1(j_1) = h_1(j_2) = i$, then from (6.1) we see that $|h_2(j_1) - h_2(j_2)| = |h_{2,i}(j_1) - h_{2,i}(j_2)| \neq 0$, where the last inequality follows from the fact that $\mathcal{E}_3$ holds and hence each $h_{2,i}$ is collision-free. This shows that h_2 is 1-1 on S. Let I denote the set $h_2(S)$ which is a subset of $[4t]$ of size exactly t.

Let $\mathcal{E}_4$ denote the event that $I' = I$. The probability that $\mathcal{E}_4$ occurs is exactly $\binom{4t}{t}^{-1} = 1/2^{O(t)}$. When $\mathcal{E}_4$ occurs as well, the function f maps I bijectively to $[t]$ and hence the function $h = f \circ h_2$ maps S bijectively to $[t]$ as well, which is exactly what we want.

Thus the probability of sampling such a "good" h is at least $\Pr[\mathcal{E}_1 \wedge \mathcal{E}_2 \wedge \mathcal{E}_3 \wedge \mathcal{E}_4] = 1/2^{O(t)}$, which proves the lemma. □

The construction of the fractional perfect hash family is almost analogous to the construction of the perfect hash family above, though the details are somewhat more involved, as we have to ensure that each bucket in the hash has roughly equal weight.

Lemma 2.13 (restated)**.** For any $n,t \in \mathbb{N}$ such that $t \leq n$, there is an explicit family of hash functions $\mathcal{H}^{n,t}_{\text{frac}} \subseteq [t]^{[n]}$ of size $2^{O(t)}n^{O(1)}$ such that for any $z \in [0,1]^n$ such that $\sum_{j\in[n]} z_j \geq 10t$, we have

$$\Pr_{h \in \mathcal{H}^{n,t}_{\text{frac}}} \left[\forall i \in [t], 0.01 \frac{\sum_{j\in[n]} z_j}{t} \leq \sum_{j \in h^{-1}(i)} z_j \leq 10 \frac{\sum_{j\in[n]} z_j}{t} \right] \geq \frac{1}{2^{O(t)}}.$$

Proof. For $S \subseteq [n]$, we define $z(S)$ to be $\sum_{j\in S} z_j$. By assumption, we have $z([n]) \geq 10t$. Without loss of generality, we assume that $z([n]) = 10t$ (otherwise, we work with $\tilde{z} = (10t/z([n]))z$ which satisfies this property; since we prove the lemma for $\tilde{z}$, it is true for z as well). We thus need to construct $\mathcal{H}^{n,t}_{\text{frac}}$ such that

$$\Pr_{h \in \mathcal{H}^{n,t}_{\text{frac}}} \left[\forall i \in [t], z(h^{-1}(i)) \in [0.1, 100] \right] \geq \frac{1}{2^{O(t)}}.$$

We describe the formal construction by describing how to sample a random element h of $\mathcal{H}^{n,t}_{\text{frac}}$. To sample a random $h \in \mathcal{H}^{n,t}_{\text{frac}}$, we do the following:

Step 1 (Top-level hashing): We choose a pairwise independent hash function $h_1 : [n] \to [10t]$ by choosing a random seed to generator $\mathcal{G}^{t,n}_{\text{2-wise}}$. By Fact 2.7, this requires $O(\log n + \log t) = O(\log n)$ bits.

Step 2 (Guessing bucket sizes): We choose at random a subset $I' \subseteq [10t]$ of size exactly t and $y_1,\ldots,y_{10t} \in \mathbb{N}$ so that $\sum_i y_i \leq 30t$. It can be checked that the number of possibilities for I' and $y_1,\ldots,y_{10t}$ is only $2^{O(t)}$.

Step 3 (Second-level hashing): By Fact 2.7, for each $i \in [10t]$, we have an explicit pairwise independent family of hash functions mapping $[n]$ to $[y_i]$ given by $\mathcal{G}^{y_i,n}_{\text{2-wise}}$. We assume w.l.o.g. that each such generator has some fixed seed length $s = O(\log n)$ (if not, increase the seed length of each to the maximum seed length among them). Let $V = \{0,1\}^s$. Using Fact 2.9, fix a sequence $(G_1,\ldots,G_{10t})$ of $10t$ many $(2^s,D,\lambda)$-expanders on set V with $D = O(1)$ and $\lambda \le 1/100$. Choosing $w \in \mathcal{W}(G_1,\ldots,G_{10t})$ uniformly at random, set $h_{2,i} : [n] \to [y_i]$ to be $\mathcal{G}^{y_i,n}_{\text{2-wise}}(v_i(w))$. Define $h_2 : [n] \to [30t]$ as follows:

$$h_2(j) = \left(\sum_{i<h_1(j), i \notin I'} y_i \right) + h_{2,h_1(j)}(j)\,.$$

Given the random choices made in the previous steps, the function h_2 is completely determined by $|\mathcal{W}(G_1,\ldots,G_{10t})|$, which is $2^{O(t)} \cdot n^{O(1)}$.

Step 4 (Folding): This step is completely deterministic given the random choices made in the previous steps. We fix an arbitrary map $f : (I' \times \{0\}) \cup ([10t] \times \{1\}) \to [t]$ with the following properties: (a) f is 1-1 on $I' \times \{0\}$, (b) f is 10-to-1 on $[10t] \times \{1\}$. We now define $h : [n] \to [t]$. Define $h(j)$ as

$$h(j) = \begin{cases} f(h_1(j),0) & \text{if } h_1(j) \in I', \\ f(h_2(j),1) & \text{otherwise.} \end{cases}$$

It is easy to check that $|\mathcal{H}^{n,t}_{\text{frac}}|$, which is the number of possibilities for the random choices made in the above steps, is bounded by $2^{O(t)} n^{O(1)}$, exactly as required.

We now show that a random $h \in \mathcal{H}^{n,t}_{\text{frac}}$ has the properties stated in the lemma. Assume h is sampled as above. We analyze the construction step-by-step. First, we recall the following easy consequence of the Paley-Zygmund inequality:

Fact 6.1. *For any non-negative random variable Z we have*

$$\Pr[Z \ge 0.1\,\mathbb{E}[Z]] \ge 0.9\,\frac{(\mathbb{E}[Z])^2}{\mathbb{E}[Z^2]}\,.$$

Consider h_1 sampled in the first step. Define, for each $i \in [10t]$, the random variables $X_i = z(h_1^{-1}(i))$ and $Y_i = \sum_{j_1 \ne j_2 : h_1(j_1) = h_1(j_2) = i} z_{j_1} z_{j_2}$, and let $X = \sum_{i \in [10t]} X_i^2$ and $Y = \sum_{i \in [10t]} Y_i$. An easy calculation shows that $X = \sum_{j \in [n]} z_j^2 + Y \le 10t + Y$. Hence, $\mathbb{E}_{h_1}[X] \le 10t + \mathbb{E}_{h_1}[Y]$ and moreover

$$\mathop{\mathbb{E}}_{h_1}[Y] = \sum_{j_1 \ne j_2} z_{j_1} z_{j_2} \Pr_{h_1}[h_1(j_1) = h_1(j_2)] \le \frac{z([n])^2}{10t} = 10t\,.$$

Let $\mathcal{E}_1$ denote the event that $Y \le 20t$. By Markov's inequality, this happens with probability at least $1/2$. We condition on any choice of h_1 so that $\mathcal{E}_1$ occurs. Note that in this case, we have $X \le 10t + Y \le 30t$.

Let $Z = X_i$ for a randomly chosen $i \in [10t]$. Clearly, we have $\mathbb{E}_i[Z] = (1/10t)\sum_i X_i = 1$ and also $\mathbb{E}_i[Z^2] = (1/10t)\sum_i X_i^2 = (1/10t)X \le 3$. Thus, Fact 6.1 implies that for random $i \in [n]$, we have $\Pr_i[Z \ge 0.1] \ge 0.3$. Markov's Inequality tells us that $\Pr_i[Z > 10] \le 0.1$. Putting things together, we see that if we set $I = \{i \in [10t] \mid X_i \in [0.1, 10]\}$, then $|I| \ge 0.2 \times 10t = 2t$. The elements of I will be referred to as the *medium-sized buckets*.

We now analyze the second step. We say that event $\mathcal{E}_2$ holds if (a) I' contains *only* medium-sized buckets, and (b) for each $i \in [10t]$, $y_i = \lceil Y_i \rceil$. We claim that $\Pr[\mathcal{E}_2] \ge 1/2^{O(t)}$. Since the number of random choices in Step 2 is only $2^{O(t)}$, it suffices to argue that there are more than t many medium-sized buckets and that $\sum_{i\in[10t]}\lceil Y_i\rceil \le 30t$. The former follows from the lower bound on $|I'|$ above, and the latter from the fact that $\sum_{i\in[10t]}\lceil Y_i\rceil \le 10t+\sum_i Y_i \le 30t$. We now condition on random choices in Step 2 so that both $\mathcal{E}_1$ and $\mathcal{E}_2$ occur.

For the third step, given $i \notin I'$, we say that hash function $h_{2,i}$ is *collision-free* if for each $k \in [y_i]$, we have $z(S_{i,k}) \le 2$ where $S_{i,k} = h_{2,i}^{-1}(k)\cap h_1^{-1}(i)$. The following simple claim shows that this condition is implied by the condition that for each k, $Y_{i,k} := \sum_{j_1\neq j_2\in S_{i,k}} z_{j_1}z_{j_2} \le 2$.

Claim 6.2. *For any $\alpha_1,\ldots,\alpha_m \in [0,1]$, if $\sum_j \alpha_j > 2$, then $\sum_{j_1\neq j_2}\alpha_{j_1}\alpha_{j_2} > 2$.*

Proof. Follows from the fact that $\sum_{j_1\neq j_2}\alpha_{j_1}\alpha_{j_2} = (\sum_j \alpha_j)^2 - \sum_j \alpha_j^2 \ge (\sum_j \alpha_j)^2 - (\sum_j \alpha_j)$, where the last inequality follows from the fact that $\alpha_1,\ldots,\alpha_m \in [0,1]$. □

For the sake of analysis, assume first that the hash functions $h_{2,i}$ ($i \in [10t]$) are chosen to be pairwise independent and *independent of each other*. Now fix any $i \in [10t]$ and $k \in [y_i]$. Then, since $h_{2,i}$ is chosen to be pairwise-independent, we have

$$\mathbb{E}[Y_{i,k}] = \sum_{j_1\neq j_2 : h_1(j_1)=h_1(j_2)=i} z_{j_1}z_{j_2} \Pr_{h_{2,i}}[h_{2,i}(j_1)=h_{2,i}(j_2)=k] = Y_i/y_i^2 \le 1/y_i\,.$$

In particular, by Markov's inequality, $\Pr[Y_{i,k}\ge 2] \le 1/2y_i$. Thus, by a union bound over k, we see that the probability that a uniformly random pairwise independent hash function $h_{2,i}$ is collision-free is at least $1/2$.

Now, let us consider the hash functions $h_{2,i}$ as defined in the above construction. Let $\mathcal{E}_3$ denote the event that for each $i \notin I'$, $h_{2,i}$ is collision-free. Hence, by Lemma 2.10, we see that

$$\Pr[\mathcal{E}_3] = \Pr_{w\in\mathcal{W}(G_1,\ldots,G_{10t})}\left[\forall i \in [10t]\setminus I' :\ h_{2,i} \text{ collision-free}\right] \ge 1/2^{O(t)}\,.$$

Thus, we have established that $\Pr[\mathcal{E}_1\wedge\mathcal{E}_2\wedge\mathcal{E}_3] \ge 1/2^{O(t)}$. We now see that when these events occur, then the sampled h satisfies the properties we need. Fix such an h and consider $i \in [t]$.

Since f is a bijection on $I'\times\{0\}$, we see that there must be an $i'\in I'$ such that $f(i',0)=i$. Since $i'\in I'$ and the event $\mathcal{E}_2$ occurs, it follows that i' is a medium-sized bucket. Thus, $z(h^{-1}(i)) \ge z(h_1^{-1}(i')) \ge 0.1$. Secondly, since $\mathcal{E}_3$ occurs, we have

$$z(h^{-1}(i)) = z(h_1^{-1}(i')) + \sum_{(\ell,1)\in f^{-1}(i)} z(h_2^{-1}(\ell)\setminus h_1^{-1}(I')) \le 10 + 10\max_{i\in[10t],k\in[y_i]} z(S_{i,k}) \le 100\,,$$

where the final inequality follows because $\mathcal{E}_3$ holds. This shows that for each i, we have $z(h^{-1}(i)) \in [0.1, 100]$ and hence h satisfies the required properties. This concludes the proof of the lemma. □

7 Expander walks

In this section we prove Lemma 2.10. For convenience we restate it below.

Lemma 2.10 (restated)**.** Let $G_1,\dots,G_\ell$ be a sequence of graphs defined on the same vertex set V of size N. Assume that G_i is an (N,D_i,λ_i)-expander. Let $V_1,\dots,V_\ell \subseteq V$ such that $|V_i| \geq p_i N > 0$ for each $i \in [\ell]$. Let $p_0 = 1$. Then, as long as for each $i \in [\ell]$, $\lambda_i \leq (p_i p_{i-1})/8$,

$$\Pr_{w \in \mathcal{W}(G_1,\dots,G_\ell)}[\forall i \in [\ell], v_i(w) \in V_i] \geq (0.75)^\ell \prod_{i \in [\ell]} p_i\,. \tag{7.1}$$

Without loss of generality, we can assume that each subset V_i ($i \in [\ell]$) has size *exactly* $p_i N$.

Let us consider an ℓ step random walk starting at a uniformly random starting vertex in V, in which step i is taken in the graph G_i. The probability distribution after ℓ steps is now given by $A_\ell A_{\ell-1}\dots A_1\mathbf{1}_N$, where $\mathbf{1}_N$ denotes the vector $(1/N,\dots,1/N)$, and A_i is the normalized adjacency matrix of the graph G_i.

Now, we are interested in the probability that a walk satisfies the property that its ith vertex is in set V_i for each i. For $\ell = 1$, for example, this is precisely the L_1 weight of the set V_1, in the vector $A_1\mathbf{1}_N$. More generally, suppose we define the operator I_S to be one which takes a vector and returns the "restriction" to S (and puts zero everywhere else), then the probability that the walk ends up in set S after one step is $\|I_{V_1}A_1\mathbf{1}_N\|_1$. In general, it is easy to see that the probability that the ith vertex in the walk is in V_i for all $1 \leq i \leq t$ is precisely $\|I_{V_t}A_t I_{V_{t-1}}A_{t-1}\dots I_{V_1}A_1\mathbf{1}_N\|_1$. We will call the vector of interest $u_{(t)}$, for convenience, and bound $\|u_{(t)}\|_1$ inductively.

Intuitively, the idea will be to show that $u_{(t)}$ should be a vector with a 'reasonable mass', and is distributed "roughly uniformly" on the set V_t. Formally, we will show the following inductive statement. Define $u_{(0)} = \mathbf{1}_N$.

Lemma 7.1. *For all $1 \leq t \leq \ell$, we have the following two conditions:*

$$\|u_{(t)}\|_1 \geq \frac{3p_t}{4}\|u_{(t-1)}\|_1\,, \tag{7.2}$$

$$\|u_{(t)}\|_2 \leq \frac{2}{\sqrt{p_t N}}\|u_{(t)}\|_1\,. \tag{7.3}$$

Note that the second equation informally says that the mass of $u_{(t)}$ is distributed roughly equally on a set of size $p_t N$. Lemma 2.10 now follows by induction using eq. (7.2) and the fact that $\|u_{(0)}\|_1 = 1$.

The proof of Lemma 7.1 is also by induction, but we will need a bit of simple notation before we start. Let us define $u^\parallel$ and $u^\perp$ to be the components of a vector u which are parallel and perpendicular (respectively) to the vector $\mathbf{1}_N$. Thus we have $u = u^\parallel + u^\perp$ for all u. The following lemma is easy to see.

Claim 7.2. *For any N-dimensional vector x with all positive entries, we have $\|x^\parallel\|_1 = \|x\|_1$. Furthermore, $x^\parallel$ is an N-dimensional vector with each entry $\|x\|_1/N$.*

Proof. The "furthermore" part is by the definition of $x^\parallel$, and the first part follows directly from it. □

We can now prove Lemma 7.1. We will use the fact that $A_i\mathbf{1}_N = \mathbf{1}_N$ for each i, and that $\|A_i u\|_2 \leq \lambda\|u\|_2$ for u orthogonal to $\mathbf{1}_N$.

Proof of Lemma 7.1. For $t = 1$, we have $u_{(1)} = I_{V_1}A_1\mathbf{1}_N = I_{V_1}\mathbf{1}_N$, and thus we have $\|u_{(1)}\|_1 = p_1$, and we have $\|u_{(1)}\|_2 = p_1/\sqrt{p_1 N}$, and thus the claims are true for $t = 1$. Now suppose $t \geq 2$, and that they are true for $t-1$.

For the first part, we observe that

$$\|u_{(t)}\|_1 = \|I_{V_t} A_t u_{(t-1)}\|_1 \geq \|I_{V_t} A_t u^{\|}_{(t-1)}\|_1 - \|I_{V_t} A_t u^{\perp}_{(t-1)}\|_1 . \tag{7.4}$$

The first term is equal to $\|I_{V_t} u^{\|}_{(t-1)}\|_1 = p_t \|u_{(t-1)}\|_1$, because I_{V_t} preserves $p_t N$ indices, and each has a contribution of $\|u_{(t-1)}\|_1/N$, by Claim 7.2.

The second term can be upper bounded as

$$\|I_{V_t} A_t u^{\perp}_{(t-1)}\|_1 \leq \sqrt{N} \|I_{V_t} A_t u^{\perp}_{(t-1)}\|_2 \leq \sqrt{N} \cdot \lambda_t \|u_{(t-1)}\|_2 \leq \frac{2\lambda_t \sqrt{N}}{\sqrt{p_{t-1} N}} \|u_{(t-1)}\|_1 ,$$

where we used the inductive hypothesis in the last step. From the condition $\lambda_t \leq p_t p_{t-1}/8$, we have that the term above is bounded above by $p_t \|u_{(t-1)}\|_1/4$. Combining this with equation (7.4), the first inequality follows.

The second inequality is proved similarly. Note that for this part we can even assume the first inequality for t, i. e., $\|u_{(t)}\|_1 \geq (3/4) p_t \|u_{(t-1)}\|_1$. We will call this (*). By the triangle inequality,

$$\|u_{(t)}\|_2 \leq \|I_{V_t} A_t u^{\|}_{(t-1)}\|_2 + \|I_{V_t} A_t u^{\perp}_{(t-1)}\|_2 . \tag{7.5}$$

The first term is the L_2 norm of a vector with support V_t, and each entry $\|u_{(t-1)}\|_1/N$, from Claim 7.2 we have that the first term is equal to

$$\frac{\|u_{(t-1)}\|_1}{N} \cdot \sqrt{p_t N} \leq \frac{4\|u_{(t)}\|_1}{3\sqrt{p_t N}} ,$$

with the inequality following from (*). The second term can be bounded by

$$\lambda_t \|u_{(t-1)}\|_2 \leq \frac{2\lambda_t}{\sqrt{p_{t-1} N}} \|u_{(t-1)}\|_1 \leq \frac{\sqrt{p_t p_{t-1}}}{4\sqrt{p_{t-1} N}} \|u_{(t-1)}\|_1 \leq \frac{1}{3\sqrt{p_t N}} \|u_{(t)}\|_1 .$$

Here we first used the inductive hypothesis, and then our choice of λ_t, followed by (*). Plugging these into equation (7.5), we obtain the second inequality.

This completes the inductive proof of the two inequalities. □

8 Open problems

We have used a two-level hashing procedure to construct hitting sets for combinatorial thresholds of low weight. It would be nice to obtain a simpler construction avoiding the use of an "inner" hitting set construction.

An interesting direction is to extend our methods to weighted variants of combinatorial shapes: functions which accept an input x iff $\sum_i \alpha_i 1_{A_i}(x_i) = S$ where $\alpha_i \in \mathbb{R}_{\geq 0}$. The difficulty here is that having hitting sets for this sum being $\geq S$ and $\leq S$ do not imply a hitting set for "$= S$." The simplest open case here is $m = 2$, all A_i being $\{1\}$, and α_i integers in $[1, 10n]$, for example. It would also be interesting to prove formally that such weighted versions can capture much stronger computational classes.

Acknowledgements

The authors are very grateful to the anonymous referees for correcting various errors and deficiencies in an earlier version of the paper and also simplifying some of the notation and proofs.

References

[1] ROMAS ALELIUNAS, RICHARD M. KARP, RICHARD J. LIPTON, LÁSZLÓ LOVÁSZ, AND CHARLES RACKOFF: Random walks, universal traversal sequences, and the complexity of maze problems. In *Proc. 20th FOCS*, pp. 218–223. IEEE Comp. Soc. Press, 1979. [doi:10.1109/SFCS.1979.34] 442

[2] 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] 445

[3] NOGA ALON, URIEL FEIGE, AVI WIGDERSON, AND DAVID ZUCKERMAN: Derandomized graph products. *Comput. Complexity*, 5(1):60–75, 1995. [doi:10.1007/BF01277956] 446, 451

[4] NOGA ALON, ODED GOLDREICH, JOHAN HÅSTAD, AND RENÉ PERALTA: Simple construction of almost k-wise independent random variables. *Random Structures & Algorithms*, 3(3):289–304, 1992. Preliminary version in FOCS'90. [doi:10.1002/rsa.3240030308] 443

[5] NOGA ALON, RAPHAEL YUSTER, AND URI ZWICK: Color-coding. *J. ACM*, 42(4):844–856, 1995. Preliminary version in STOC'94. [doi:10.1145/210332.210337] 449, 452

[6] ROY ARMONI, MICHAEL E. SAKS, AVI WIGDERSON, AND SHIYU ZHOU: Discrepancy sets and pseudorandom generators for combinatorial rectangles. In *Proc. 37th FOCS*, pp. 412–421. IEEE Comp. Soc. Press, 1996. [doi:10.1109/SFCS.1996.548500] 443

[7] AVRIM BLUM, ADAM KALAI, AND HAL WASSERMAN: Noise-tolerant learning, the parity problem, and the statistical query model. *J. ACM*, 50(4):506–519, 2003. Preliminary version in STOC'00. [doi:10.1145/792538.792543] 442

[8] GUY EVEN, ODED GOLDREICH, MICHAEL LUBY, NOAM NISAN, AND BOBAN VELIČKOVIĆ: Efficient approximation of product distributions. *Random Structures & Algorithms*, 13(1):1–16, 1998. Preliminary version in STOC'92. [doi:10.1002/(SICI)1098-2418(199808)13:1<1::AID-RSA1>3.0.CO;2-W] 443, 445

[9] WILLIAM FELLER: *An Introduction to Probability Theory and its Applications, Vol 2*. Wiley, 1971. 450

[10] MICHAEL L. FREDMAN, JÁNOS KOMLÓS, AND ENDRE SZEMERÉDI: Storing a sparse table with 0(1) worst case access time. *J. ACM*, 31(3):538–544, 1984. Preliminary version in FOCS'82. [doi:10.1145/828.1884] 446

[11] PARIKSHIT GOPALAN, RAGHU MEKA, OMER REINGOLD, AND DAVID ZUCKERMAN: Pseudorandom generators for combinatorial shapes. In *Proc. 43rd STOC*, pp. 253–262. ACM Press, 2011. [doi:10.1145/1993636.1993671] 442, 444, 448, 450, 451

[12] SHLOMO HOORY, NATHAN LINIAL, AND AVI WIGDERSON: Expander graphs and their applications. *Bulletin of the AMS*, 43(4):439–561, 2006. [doi:10.1090/S0273-0979-06-01126-8] 445

[13] RUSSELL IMPAGLIAZZO AND AVI WIGDERSON: $P = BPP$ if E requires exponential circuits: Derandomizing the XOR lemma. In *Proc. 29th STOC*, pp. 220–229. ACM Press, 1997. [doi:10.1145/258533.258590] 442

[14] MICHAL KOUCKÝ, PRAJAKTA NIMBHORKAR, AND PAVEL PUDLÁK: Pseudorandom generators for group products: extended abstract. In *Proc. 43rd STOC*, pp. 263–272. ACM Press, 2011. [doi:10.1145/1993636.1993672] 442

[15] NATHAN LINIAL, MICHAEL LUBY, MICHAEL E. SAKS, AND DAVID ZUCKERMAN: Efficient construction of a small hitting set for combinatorial rectangles in high dimension. *Combinatorica*, 17(2):215–234, 1997. Preliminary version in STOC'93. [doi:10.1007/BF01200907] 442, 443, 444, 447, 452, 454, 458

[16] SHACHAR LOVETT, OMER REINGOLD, LUCA TREVISAN, AND SALIL P. VADHAN: Pseudorandom bit generators that fool modular sums. In *Proc. 13th Internat. Workshop on Randomization and Computation (RANDOM'09)*, pp. 615–630. Springer, 2009. [doi:10.1007/978-3-642-03685-9_46] 442, 443

[17] CHI-JEN LU: Improved pseudorandom generators for combinatorial rectangles. *Combinatorica*, 22(3):417–434, 2002. Preliminary version in ICALP'98. [doi:10.1007/s004930200021] 442, 443

[18] RAGHU MEKA AND DAVID ZUCKERMAN: Small-bias spaces for group products. In *Proc. 13th Internat. Workshop on Randomization and Computation (RANDOM'09)*, pp. 658–672, 2009. [doi:10.1007/978-3-642-03685-9_49] 443

[19] ROBIN A. MOSER AND GÁBOR TARDOS: A constructive proof of the general Lovász Local Lemma. *J. ACM*, 57(2):11, 2010. Preliminary version in STOC'09. [doi:10.1145/1667053.1667060] 442

[20] JOSEPH NAOR AND MONI NAOR: Small-bias probability spaces: Efficient constructions and applications. *SIAM J. Comput.*, 22(4):838–856, 1993. Preliminary version in STOC'90. [doi:10.1137/0222053] 443

[21] NOAM NISAN: Pseudorandom generators for space-bounded computation. *Combinatorica*, 12(4):449–461, 1992. Preliminary version in STOC'90. [doi:10.1007/BF01305237] 442

[22] NOAM NISAN AND AVI WIGDERSON: Hardness vs randomness. *J. Comput. System Sci.*, 49(2):149–167, 1994. [doi:10.1016/S0022-0000(05)80043-1] 442

[23] NOAM NISAN AND DAVID ZUCKERMAN: Randomness is linear in space. *J. Comput. System Sci.*, 52(1):43–52, 1996. Preliminary version in STOC'93. [doi:10.1006/jcss.1996.0004] 442

[24] YUVAL RABANI AND AMIR SHPILKA: Explicit construction of a small ε-net for linear threshold functions. *SIAM J. Comput.*, 39(8):3501–3520, 2010. Preliminary version in STOC'09. [doi:10.1137/090764190] 443, 446, 450, 460

[25] JEANETTE P. SCHMIDT AND ALAN SIEGEL: The analysis of closed hashing under limited randomness (extended abstract). In *Proc. 22nd STOC*, pp. 224–234. ACM Press, 1990. [doi:10.1145/100216.100245] 446

[26] RONEN SHALTIEL AND CHRISTOPHER UMANS: Pseudorandomness for approximate counting and sampling. *Comput. Complexity*, 15(4):298–341, 2006. Preliminary version in CCC'05. [doi:10.1007/s00037-007-0218-9] 442

[27] THOMAS WATSON: Pseudorandom generators for combinatorial checkerboards. *Comput. Complexity*, pp. 1 – 43, 2012. Preliminary version in CCC'11. [doi:10.1007/s00037-012-0036-6] 443

AUTHORS

Aditya Bhaskara
Postdoctoral Researcher
EPFL
bhaskara@cs.princeton.edu
http://www.cs.princeton.edu/~bhaskara/

Devendra Desai
Ph. D. student
Rutgers University
devdesai@cs.rutgers.edu
http://www.cs.rutgers.edu/~devdesai/

Srikanth Srinivasan
Assistant Professor
IIT Bombay
srikanth@math.iitb.ac.in
http://math.iitb.ac.in/~srikanth/

ABOUT THE AUTHORS

ADITYA BHASKARA graduated from Princeton University in 2012; his advisor was Moses Charikar. His thesis was on finding dense structures in graphs and matrices. His research interests are in approximation algorithms, and in the use of tools from probability and convex geometry in theoretical CS. He did his undergraduate studies at IIT Bombay; he was advised by Abhiram Ranade and Ajit Diwan, who helped shape his interests in algorithms and theoretical computer science.

DEVENDRA (DEV) DESAI is a Ph. D. student at Rutgers University, advised by Mario Szegedy. His research interests include approximation algorithms, randomized algorithms, derandomization, hardness of approximation, and combinatorics. During his undergraduate days in Pune, India, he was mentored in algorithm analysis by Udayan Kanade, who to this day offers free lectures in various math and computer science areas to anyone who shows up. Dev's interest in theoretical CS was further strengthened during his master's studies at IIT Kharagpur. In his free time he likes to take short walks and, when stationary, likes listening to 70's rock and Hindi music.

SRIKANTH SRINIVASAN got his undergraduate degree from the Indian Institute of Technology Madras, where his interest in the theory side of CS was piqued under the tutelage of N. S. Narayanswamy. Subsequently, he obtained his Ph. D. from The Institute of Mathematical Sciences, Chennai, in 2011; his advisor was V. Arvind. His research interests span all of TCS (in theory), but in practice are limited to circuit complexity, derandomization, and related areas of mathematics. He enjoys running and pretending to play badminton.