

Explicit Min-wise Hash Families with Optimal Size

Xue Chen*

Shengtang Huang[†]

Xin Li[‡]

Abstract

We study explicit constructions of min-wise hash families and their extension to k -min-wise hash families. Informally, a min-wise hash family guarantees that for any fixed subset $X \subseteq [N]$, every element in X has an equal chance to have the smallest value among all elements in X ; a k -min-wise hash family guarantees this for every subset of size k in X . Min-wise hash is widely used in many areas of computer science such as sketching [Coh16], web page detection [Hen06], and ℓ_0 sampling [CF14]. For applications like similarity estimation [CDF⁺01] and rarity estimation [DM02], the space complexity of their streaming algorithms is roughly equal to the number of random bits used to construct such families.

The classical works by Indyk [Ind01] and Pătraşcu and Thorup [PT16] have shown $\Theta(\log(1/\delta))$ -wise independent families give min-wise hash of multiplicative (relative) error δ , resulting in a construction with $\Theta(\log(1/\delta) \log N)$ random bits. While this is optimal for constant errors, it leaves a gap to the existential bound of $O(\log(N/\delta))$ bits whenever δ is sub-constant, which is needed in several applications. Based on a reduction from pseudorandom generators for combinatorial rectangles by Saks, Srinivasan, Zhou and Zuckerman [SSZZ00], Gopalan and Yehudayoff [GY20] improved the number of bits to $O(\log N \log \log N)$ for polynomially small errors δ . However, no construction with $O(\log N)$ bits (polynomial size family) and sub-constant error was known before.

In this work, we continue and extend the study of constructing (k) -min-wise hash families from pseudorandomness for combinatorial rectangles and read-once branching programs. Our main result gives the first explicit min-wise hash families that use an optimal (up to constant) number of random bits and achieve a sub-constant (in fact, almost polynomially small) error, specifically, an explicit family of k -min-wise hash with $O(k \log N)$ bits and $2^{-O(\frac{\log N}{\log \log N})}$ error. This improves all previous results for any $k = \log^{O(1)} N$ under $O(k \log N)$ bits. Our main techniques involve several new ideas to adapt the classical Nisan-Zuckerman pseudorandom generator to fool min-wise hashing with a *multiplicative* error.

*xuechen1989@ustc.edu.cn, University of Science and Technology of China & Hefei National Laboratory, Hefei 230088, China. Supported by Innovation Program for Quantum Science and Technology 2021ZD0302901, NSFC 62372424, and CCF-HuaweiLK2023006.

[†]peanuttang@mail.ustc.edu.cn, School of the Gifted Young, University of Science and Technology of China.

[‡]lixints@cs.jhu.edu, Johns Hopkins University. Supported by NSF CAREER Award CCF-1845349 and NSF Award CCF-2127575.

1 Introduction

Min-wise hash families play a crucial role in the design of graph algorithms and streaming algorithms. Notable applications include similarity estimation [CDF⁺01], rarity estimation [DM02], data mining [HGKI02, Hen06], sketching [Coh16], and ℓ_0 sampler [CF14, McG14]. In this work, we study explicit constructions of min-wise hash families with small size. In pseudorandomness, this is equivalent to studying the seed length (number of random bits used) to generate a hash function. In big data algorithms, this is the space complexity of applying min-wise hash families.

Following the standard notation of the min-wise hash [BCFM00, Ind01, FPS11], we consider multiplicative (relative) errors with respect to the fair probability in this work. Although this is different from the standard additive errors in pseudorandomness, multiplicative errors are crucial for many algorithmic applications of min-wise hash such as similarity estimation [CDF⁺01] and ℓ_0 sampling [McG14].

Definition 1.1. Let $a = b \pm \delta$ denote $a \in [b - \delta, b + \delta]$. Then a min-wise hash family $\mathcal{H} = \{h : [N] \rightarrow [M]\}$ of error δ satisfies that for any $X \subseteq [N]$ and any $y \in X$,

$$\Pr_{h \sim \mathcal{H}} \left[h(y) < \min_{x \in X \setminus y} h(x) \right] = \frac{1 \pm \delta}{|X|}. \quad (1.1)$$

Moreover, a k -min-wise hash family of error δ satisfies that for any $X \subseteq [N]$ and any $Y \in \binom{X}{\leq k}$,

$$\Pr_{h \sim \mathcal{H}} \left[\max_{y \in Y} h(y) < \min_{x \in X \setminus Y} h(x) \right] = \frac{1 \pm \delta}{\binom{|X|}{|Y|}}. \quad (1.2)$$

We sometimes call $\log |\mathcal{H}|$ the seed length of the hash family.

In this work, we will focus on the case of $M = \Omega(N/\delta)$ such that the uniform distribution over all functions from $[N]$ to $[M]$ satisfies (1.1) and (1.2) [Ind01, FPS11]. Specifically, let U be the uniform distribution over all functions $h : [N] \rightarrow [M]$. Then $M = \Omega(N/\delta)$ implies

$$\Pr_{h \sim U} \left[h(y) < \min_{x \in X \setminus y} h(x) \right] = \frac{1 \pm \delta}{|X|} \text{ and } \Pr_{h \sim U} \left[\max_{y \in Y} h(y) < \min_{x \in X \setminus Y} h(x) \right] = \frac{1 \pm \delta}{\binom{|X|}{|Y|}}. \quad (1.3)$$

This is equivalent to the requirement $|X| = O(\delta N)$ for $M = N$ in previous works [SSZZ00, Ind01, FPS11]. Moreover, most applications of min-wise hash could choose the image size $|M|$ to guarantee (1.3).

Hence, constructing min-wise hash is equivalent to constructing pseudorandom generators (PRGs) with *multiplicative* errors. In this work, we will consider both sides of (1.2) and (1.3) as the targets of our PRGs.

Although (k)-min-wise hash has found a variety of applications in computer science, the primary approaches of explicit constructions are based on t -wise independent hash families [Ind01, FPS11] and pseudorandom generators for combinatorial rectangles [SSZZ00, GY20]. For min-wise hash, Indyk [Ind01] showed that any $O(\log(1/\delta))$ -wise independent family is a min-wise hash family of error δ . Since t -wise independent families need $\Theta(t \log NM)$ random bits, this construction needs $O(\log(1/\delta) \log NM)$ bits. In fact, Pătraşcu and Thorup [PT16] showed a matching lower bound: $\Omega(\log(1/\delta))$ -wise independence is necessary to have error δ . In contrast, non-explicitly it is known that one can use $O(\log(NM/\delta))$ random bits to construct min-wise hash families of error δ . Therefore, although the construction in [Ind01] is optimal for *constant errors*, it fails to be optimal whenever the error is *sub-constant*.

For k -min-wise hash, Feigenblat, Porat and Shiftan [FPS11] showed that t -wise independent family is also a k -min-wise hash family of error δ when $t = O(\log(1/\delta) + k \log \log(1/\delta))$. In turn, this construction needs $O((\log(1/\delta) + k \log \log(1/\delta)) \cdot \log NM)$ random bits, which still leaves a gap to the optimal result of $O(k \log NM + \log(1/\delta))$ bits for sub-constant $\delta = o(1)$.

At the same time, Saks, Srinivasan, Zhou and Zuckerman [SSZZ00] reduced the construction of min-wise hash to pseudorandom generators for combinatorial rectangles of *polynomially small errors*. This reduction translates polynomially small (additive) errors to a multiplicative error like $\delta/|X|$ (described in [Appendix A](#)). Based on this reduction, Gopalan and Yehudayoff [GY20] provided a min-wise hash family of $O(\log NM \log \log NM)$ bits for any polynomially small error. Although this improves the result by Indyk [Ind01] of $O(\log^2 NM)$ bits when δ is polynomially small, it does not provide a construction with $O(\log NM)$ bits even when δ is a constant.

In this work, we study explicit constructions of min-wise hash with small sizes and (almost) polynomially small errors. Our constructions are well motivated, given that in practice, some applications of min-wise hash require small errors in which the seed length becomes the bottleneck on the space complexity of streaming algorithms. For example, a primary application of min-wise hash is ℓ_0 sampling in the streaming model. Many graph streaming algorithms need $(k > 1)$ -min-wise hash with a sub-constant error $\delta = o(1)$ (see Table 1 in [KNP⁺17]) and have space complexity that is equal to the seed length of the min-wise hash family times the number of hashes used [AGM12]. Also, similarity estimation [CDF⁺01, DM02] applies k -min-wise hash of error δ directly to approximate the Jaccard similarity coefficient $\hat{S}(A, B) := \frac{|A \cap B|}{|A \cup B|}$ between two sets A and B as $(1 \pm \delta) \cdot \frac{|A \cap B|}{|A \cup B|} \pm \frac{O(1)}{\delta \cdot k}$, so the space complexity is equal to the seed length of the k -min-wise hash family here.

Furthermore, from a different aspect, given the connection between min-wise hash families and PRGs for combinatorial rectangles shown by Saks, Srinivasan, Zhou and Zuckerman [SSZZ00], a natural direction is to apply the results on the long line of research on PRGs for combinatorial rectangles to construct better min-wise hash families. Constructing pseudorandom generators for combinatorial rectangles have been extensively studied (e.g., [ASWZ96, LLSZ97, Lu02, GMR⁺12, GY20] to name a few) because they are related to fundamental problems in theoretical computer science such as derandomizing logspace computation and approximately counting the number of satisfying assignments of a CNF formula. While early works [ASWZ96, Lu02] in the 90s have already provided PRGs with seed length $O(\log NM)$ and slightly sub-constant errors (e.g., $2^{-\sqrt{\log NM}}$ [ASWZ96] and $2^{-\log^{2/3} NM}$ [Lu02]), no construction of min-wise hash family with $O(\log NM)$ bits and a sub-constant error was known before.

The main bottleneck is that min-wise hash requires a multiplicative error such as $\delta/|X|$. Even for a constant δ , this becomes a polynomially small error like $1/N$ when $|X| = \Omega(N)$. Hence those PRGs for combinatorial rectangles with $O(\log NM)$ seed length do not give a min-wise hash directly. In fact, even after so many years of extensive study on PRGs for combinatorial rectangles, we still don't have explicit constructions of such PRGs with $O(\log NM)$ seed length and $1/(NM)^{O(1)}$ additive error. Therefore, directly applying these PRGs is not enough to get a min-wise hash family with seed length $O(\log NM)$. To address this barrier, in this work we provide several new ideas to extend the Nisan-Zuckerman PRG [NZ96] and construct min-wise hash families with $O(\log NM)$ seed length and almost polynomially small errors.

1.1 Our Results

Our main results are an explicit construction of min-wise hash families with seed length $O(\log N)$ and almost polynomially small errors and its generalization to k -min-wise hash. For ease of exposition,

we assume $M = (N/\delta)^{O(1)}$.

Theorem 1.2. *Given any N , there exists an explicit family of min-wise hash of $O(\log N)$ bits and (multiplicative) error $\delta = 2^{-O\left(\frac{\log N}{\log \log N}\right)}$.*

We remark that the seed length of our min-wise hash is optimal up to constants and the error is almost polynomial up to a $\log \log N$ factor in the exponent. Hence [Theorem 1.2](#) improves previous results [[SSZZ00](#), [Ind01](#), [FPS11](#), [GY20](#)] for seed length $O(\log N)$. In fact, this is the first construction of min-wise hash family with optimal seed length and sub-constant error.

Next we state its generalization to k -min-wise hash.

Theorem 1.3. *Given any $k = \log^{O(1)} N$, there exists an explicit k -min-wise hash family of $O(k \log N)$ bits and (multiplicative) error $\delta = 2^{-O\left(\frac{\log N}{\log \log N}\right)}$.*

Again, this is the first construction of k -min-wise hash with optimal seed length and sub-constant error. One remark is that k -min-wise hash requires $\Omega(k \log N)$ bits even for a constant error. This is because the fair probability could be as small as $1/\binom{N}{k}$ in Definition (1.2). Also, as a direct application, our constructions give the optimal space complexity for many applications including similarity estimation and rarity estimation [[CDF⁺01](#), [DM02](#)] whenever $k = \log^{O(1)} N$ and $\delta = 2^{-O\left(\frac{\log N}{\log \log N}\right)}$.

1.2 Technique Overview

First of all, the connection shown in [[SSZZ00](#)] is not enough to directly use known PRGs for combinatorial rectangles to construct min-wise hash families with $O(\log N)$ bits. This is because one remarkable feature of min-wise hash is that the error is *multiplicative* with respect to the fair probability $1/|X|$, which could be as small as $1/N$. On the other hand, standard pseudorandom generators only consider additive errors, and constructing $O(\log NM)$ seed length PRGs fooling combinatorial rectangles with error $1/N$ is still a big open problem. More broadly, the long line of research on classical PRGs for read-once branching programs (ROBPs) ([[Nis92](#), [INW94](#), [NZ96](#), [BRRY10](#), [BV10](#), [KNP11](#), [FK18](#), [MRT19](#)] to name a few) does not give a multiplicative error with $O(\log N)$ bits of seed.

To overcome this barrier, our main technical contribution is to extend the Nisan-Zuckerman PRG framework [[NZ96](#)] to achieve a small multiplicative error. For convenience, we use $\max h(S) := \max_{x \in S} h(x)$ and $\min h(S) := \min_{x \in S} h(x)$ in the rest of this work. To illustrate our ideas, let us consider how to fool $\Pr_{h \sim U}[h(y) < \min h(X \setminus y)]$ for a sub-constant error with $O(\log N)$ bits.

It would be more convenient to enumerate $\theta := h(y)$ and decompose

$$\Pr_{h \sim U}[h(y) < \min h(X \setminus y)] = \sum_{\theta \in [M]} \Pr_{h \sim U}[h(y) = \theta \wedge \min h(X \setminus y) > \theta], \quad (1.4)$$

instead of analyzing $\Pr_{h \sim U}[h(y) < \min h(X \setminus y)]$ itself (because $\Pr_{h \sim U}[h(y) < h(x_1)]$ and $\Pr_{h \sim U}[h(y) < h(x_2)]$ are correlated). Since the first event $\Pr_U[h(y) = \theta] = 1/M$ in (1.4), our first goal is to fool $\Pr_{h \sim U}[h(y) = \theta \wedge \min h(X \setminus y) > \theta]$ in (1.4) with a multiplicative error like $\delta \cdot \Pr_{h \sim U}[h(y) = \theta] = \delta/M$ for $O(\log N)$ bits (assuming $M = N^{O(1)}$). We remark that this is a polynomially small additive error. While $\mathbf{1}(h(y) = \theta \wedge \min h(X \setminus y) > \theta)$ ($\mathbf{1}$ denotes the indicator function) is a combinatorial rectangle and a simple RBP of width 2, no known PRGs for combinatorial rectangles or ROBPs can fool it with additive error δ/M given $O(\log NM)$ bits of seed.

Since most PRGs for combinatorial rectangles and ROBPs are based on Nisan's PRG [Nis92] (or its extension to the INW PRG [INW94]), a first attempt would be to modify these PRGs. However, the random event $(h(y) = \theta)$ already takes $\log_2 M$ bits. Since y could be any element, it is unclear how to revise these PRGs to replenish so many random bits just for one step (of $h(y)$).

Another candidate is the Nisan-Zuckerman PRG [NZ96]. Recall the basic construction of the Nisan-Zuckerman PRG: It prepares a random source w of length $C \log N$ and $\ell = \log^c N$ seeds of length $\frac{\log N}{\ell}$ (for two constants $C > 1$ and $c < 1$); then it applies an extractor $\text{Ext} : \{0, 1\}^{C \log N} \times \{0, 1\}^{(\log N)/\ell} \rightarrow \{0, 1\}^{\frac{C}{3} \log N}$ (see Definition 2.5) to obtain ℓ outputs $\text{Ext}(w, s_1), \text{Ext}(w, s_2), \dots, \text{Ext}(w, s_\ell)$. The analysis relies on the fact that a ROBP (or a small-space algorithm) cannot record too much information of w , therefore each $\text{Ext}(w, s_i)$ is close to an independent and uniform random string.

While this PRG only outputs $\frac{C}{3} \log N \cdot \ell = O(\log^{1+c} N)$ bits, one could stretch it to a vector in $[M]^N$ via balls-into-bins: first hashing these N variables into ℓ buckets and then using $\text{Ext}(w, s_i)$ to generate a $C/3$ -wise independent function for every bucket.

However, due to the limited length of s_i , the error of each $\text{Ext}(w, s_i)$ is $2^{-O(\frac{\log N}{\ell})} = N^{-o(1)}$, which is too large compared to $\Pr[h(y) = \theta] = 1/M$.

To address this issue, our starting point is that the Nisan-Zuckerman PRG can fool ROBPs with any input order. More attractively, we can choose the input order in our favor. We provide two constructions that explore this advantage in two different ways. Both can fool $\Pr_{h \sim U}[h(y) = \theta \wedge \min h(X \setminus y) > \theta]$ with an error δ/M for $\delta = 2^{-O(\frac{\log N}{\log \log N})}$.

Approach 1. We consider a special type of extractors, which provides a strong guarantee when the source is uniform. Observe that it never hurts to put $h(y)$ as the first input of the ROBP under the Nisan-Zuckerman PRG. Suppose y is in bucket $j \in [\ell]$ such that the value $h(y)$ is generated by $\text{Ext}(w, s_j)$. Our observation here is that as the first input, the random source w is uniform at the beginning, hence one could expect stronger properties for $\text{Ext}(w, s_j)$ than for other $\text{Ext}(w, s_i)$ where $i \neq j$. In particular, we build an extractor such that $\text{Ext}(U_n, s)$ is uniform for a uniform source and any fixed seed. The construction is based on the sequence of works in pseudorandomness that designed *linear seeded* randomness extractors [NZ96, Tre01, GUV09].

Back to the construction of min-wise hash, this guarantees $\text{Ext}(w, s_j)$ is uniform (without any error) such that it has a fair probability of $\Pr[h(y) = \theta]$. Hence the Nisan-Zuckerman PRG has a multiplicative error with respect to $\Pr[h(y) = \theta] = 1/M$ when it applies the extractor described above.

However, plugging this into (1.4) only gives an error like $1/\log^{O(1)} N$ because $\ell \leq \log N$ and it only provides constant-wise independence in each bucket. To reduce the error to be as small as possible, we apply the PRG for combinatorial rectangles to generate $\ell = 2^{\frac{\log N}{\log \log N}}$ seeds and use $\text{Ext}(w, s_j)$ to provide both t -wise independence and pseudorandomness against combinatorial rectangles. We describe this construction in Section 3.

Approach 2. Next we consider how to build a hash family \mathcal{H} fooling (1.4) like a conditional event

$$\Pr_{h \sim \mathcal{H}}[h(y) = \theta] \cdot \Pr_{h \sim \mathcal{H}}[\min h(X \setminus y) > \theta \mid h(y) = \theta] = \Pr_{h \sim U}[h(y) = \theta] \cdot \left(\Pr_{h \sim U}[\min h(X \setminus y) > \theta] \pm \delta \right) \quad (1.5)$$

with a multiplicative error δ . On the one hand, $O(1)$ -wise independent family would guarantee $\Pr_{h \sim \mathcal{H}}[h(y) = \theta] = \Pr_{h \sim U}[h(y) = \theta]$. On the other hand, $\mathbf{1}(\min h(X \setminus y) > \theta)$ is a combinatorial rectangle where several PRGs [ASWZ96, Lu02, GMR⁺12, GY20] can fool it with a small additive error using $\tilde{O}(\log N)$ bits of seed. This suggests us to consider the direct sum of a t -wise independent

family and a PRG for combinatorial rectangles (see definition in [Section 2](#)). However, it is unclear how to argue that this sum fools the conditional event $\Pr_{h \sim \mathcal{H}}[\min h(X \setminus y) > \theta \mid h(y) = \theta]$.

Our key observation here is that the sum of a t -wise independent family and the Nisan-Zuckerman PRG can actually fool it! Roughly, this is because as before, one can put the output $\text{Ext}(w, s_j)$ generating $h(y)$ at the beginning and fix it. Then we can use t -wise independence to argue about $\Pr[h(y) = \theta]$ for this bucket and we fix a specific t -wise independent function satisfying $h(y) = \theta$. These two steps only reduce the min-entropy of w by a constant fraction, therefore the Nisan-Zuckerman framework still works. We describe how to build k -min-wise hash based on this approach in [Section 4](#).

1.3 Discussion

In this work, we study explicit constructions of small size min-wise hash functions. Although Saks, Srinivasan, Zhou and Zuckerman [[SSZZ00](#)] have reduced the construction of min-wise hash to PRGs for combinatorial rectangles, previous works do not provide any explicit family of sub-constant (multiplicative) errors with $O(\log N)$ bits.

Our main technical contribution is to construct k -min-wise hash of $O(k \log N)$ bits and almost-polynomial $2^{-O\left(\frac{\log N}{\log \log N}\right)}$ -error via the Nisan-Zuckerman framework for any $k = \log^{O(1)} N$. Our results extend the Nisan-Zuckerman framework in several aspects. For example, our construction shows that one could guarantee one output of those extractions is uniform (without any error). We also show that the direct sum of the Nisan-Zuckerman PRG with other PRGs could fool conditional events and provide multiplicative errors with respect to small probability events.

We list several open questions here.

1. For k -min-wise hash with a (relative) large k like $\log N$, can we have constructions with $O(k \log N)$ bits and polynomially small (multiplicative) error? As mentioned earlier, k -min-wise hash needs at least $\Omega(k \log N)$ bits, which is $\Omega(\log^2 N)$ when $k = \Omega(\log N)$. At the same time, there are many PRGs for combinatorial rectangles with polynomially small (additive) errors and $O(\log^2 N)$ seed length.
2. It is interesting to investigate PRGs fooling conditional events like [\(1.5\)](#). For example, can we show the direct sum of a t -wise independent function and the Nisan PRG has a small multiplicative error for [\(1.5\)](#)?
3. The fact that the Nisan-Zuckerman PRG can fool any input order has been used to fool formulas and general branching programs [[IMZ19](#)]. Can we find more applications of this powerful framework?

1.4 Related Works

Min-wise hash was introduced and investigated by Broder, Charikar, Frieze and Mitzenmacher [[BCFM00](#)] where the first definition set the probability to be exact $1/|X|$. This is equivalent to requiring that each function in the family is a permutation when $M = N$. Such a family is also called min-wise permutation. However, they showed a lower bound $\Omega(N)$ on the number of bits for the exact probability, and suggested to consider min-wise hash with multiplicative (relative) error for applications like similarity estimation and duplicate detection.

Later on, Indyk showed the first construction that $O(\log(1/\delta))$ -wise independent families are min-wise hashing of error δ . A matching lower bound on t -wise independent family was shown by Pătraşcu and Thorup [[PT16](#)] later.

For polynomially small errors, Saks, Srinivasan, Zhou and Zuckerman [SSZZ00] provided a construction of $O(\log^{3/2} N)$ bits based on the PRGs for combinatorial rectangles; this was improved to $O(\log N \log \log N)$ by Gopalan and Yehudayoff [GY20]. This work [GY20] is still the state-of-the-art of both PRGs for combinatorial rectangles and min-wise hash given *polynomially small errors*.

While one could run k parallel min-wise hash to sample k elements with replacement, k -min-wise looks for a sampling without replacement. This turns out to be more accurate in practice [CDF⁺01, DM02]. Feigenblat, Porat and Shifan [FPS11] showed that $O(\log(1/\delta) + k \log \log(1/\delta))$ -wise independent families are k -min-wise hashing of error δ . Based on PRGs for combinatorial rectangles, Gopalan and Yehudayoff [GY20] constructed k -min-wise hash of $O(k \log N \cdot \log(k \log N))$ bits for polynomially small errors.

Min-wise hash families and combinatorial rectangles are subclasses of read-once branching programs. So the classical PRG by Nisan [Nis92] of $O(\log^2 N)$ bits implies a min-wise hash family of the same seed length. A long line of research has studied the effects and limitations of Nisan's PRG (to name a few [INW94, BRRY10, BV10, De11, KNP11]). However, there has been little quantitative progress on the improvement of Nisan's PRG. One exception is the construction of PRGs for combinatorial rectangles, where subsequent works [ASWZ96, Lu02] have reduced the seed length to $O(\log N)$ and achieved smaller errors.

The Nisan-Zuckerman PRG [NZ96] provides another method to derandomize ROBPs of seed length $O(\log N)$. While its output length is just $\log^{O(1)} N$, it can fool ROBPs of any input order. Impagliazzo, Meka and Zuckerman [IMZ19] have used this property to fool general formulas and branching programs.

Another powerful paradigm to design PRGs for ROBPs is via milder restrictions. Several beautiful applications are the PRGs for combinatorial rectangles and read-once CNFs [GMR⁺12, GKM15, GY20], the PRGs for ROBPs with an arbitrary input order [FK18], and the PRGs for ROBPs of width 3 [MRT19].

1.5 Paper Organization

In [Section 2](#), we describe basic notations, definitions, and useful theorems from previous works. In [Section 3](#), we show an explicit construction of min-wise hash based on the first approach described in [Section 1.2](#), which proves [Theorem 1.2](#). In [Section 4](#), we show another construction of k -min-wise hash based on the second approach described in [Section 1.2](#), which proves [Theorem 1.3](#). In [Section 5](#), we show the extractor whose output is uniform when the input source is uniform.

2 Preliminaries

Notations. For three real variables a , b and δ , $a = b \pm \delta$ means the error between a and b is $|a - b| \leq \delta$.

Let $[n]$ denote $\{1, 2, \dots, n\}$ and $\binom{n}{k}$ denote the binomial coefficient. For a subset X , we use $\binom{X}{k}$ to denote the family of all subsets with size k . For a vector or a string, we use $|\cdot|$ to denote its dimension or length.

In this work, for a function $h : [N] \rightarrow [M]$, we view it as a vector in $[M]^N$ and vice versa. For a subset $S \subseteq [N]$, let $h(S)$ denote the sub-vector in S . Then we use $\max h(S)$ to denote $\max_{x \in S} h(x)$, $\min h(S)$ to denote $\min_{x \in S} h(x)$ and $h(S) = \theta$ to denote the event $(h(x) = \theta, \forall x \in S)$.

For two functions $f, g : [N] \rightarrow [M]$, we use $f + g$ to denote their *direct sum* on every entry x in $[M]$, i.e., $(f + g)(x) = (f(x) + g(x) - 1) \bmod M + 1$. Similarly, for two vectors f and $g \in [M]^N$, $f + g$ denotes the corresponding vector.

Combinatorial rectangles and read-once branching programs. For an event A , let $\mathbf{1}(A)$ denote its indicator function. Given the alphabet $[M]$ and N subsets $S_1, \dots, S_N \subseteq [M]$, its combinatorial rectangle is the function $f : [M]^N \rightarrow \{0, 1\}$ defined as the product of N independent events $x_1 \in S_1, \dots, x_N \in S_N$: $f(x) := \prod_{i=1}^N \mathbf{1}(x_i \in S_i)$.

Equivalently, it is a function $f : [M]^N \rightarrow \{0, 1\}$ defined as $f(v) = \prod_{i=1}^N f_i(v_i)$ by N arbitrary functions $f_1, \dots, f_N : [M] \rightarrow \{0, 1\}$.

Combinatorial rectangles are a special type of read-once branching programs.

Definition 2.1 (Read-once branching program). *An width- w length- n read-once branching program on alphabet Γ is a layered directed graph M with $n + 1$ layers and w vertices per layer with the following properties.*

- *The first layer has a single start node and the vertices in the last layer are labeled by 0 or 1.*
- *Each vertex v in layer i ($0 \leq i < n$) has $|\Gamma|$ edges to layer $i + 1$, each labeled with an element in Γ .*

A graph M as above naturally defines a function $M : \Gamma^n \rightarrow \{0, 1\}$, where on input $(x_1, \dots, x_n) \in \Gamma^n$, one traverses the edges of the graph according to the labels and outputs the label of the final vertex reached.

Pseudorandomness. For a fixed domain like $[n]$ or $[M]^N$, we use U to denote the uniform distribution on this domain. Moreover, we use U_n to denote the uniform distribution in $\{0, 1\}^n$.

For two distributions D and D' in the domain, we use $D \approx_\varepsilon D'$ to indicate that their statistical distance is at most ε and call this fact D is ε -close to D' .

Definition 2.2 (Pseudorandom generator). *Given a fixed domain D and a family of functions from D to $\{0, 1\}$, a pseudorandom generator (PRG) $G : \{0, 1\}^\ell \rightarrow D$ ε -fools this family \mathcal{F} if*

$$\forall f \in \mathcal{F}, \quad \mathbb{E}_{s \sim U_\ell} [f(G(s))] = \mathbb{E}_{x \sim U} [f(x)] \pm \varepsilon.$$

We call ℓ the seed length of G and ε its error.

One basic component to construct pseudorandom generators is t -wise independent family(function).

Definition 2.3. *We say a distribution D is t -wise independent in $[M]^N$ if for any k distinct positions i_1, \dots, i_k in $[M]$, the marginal distribution $D(x_{i_1}, \dots, x_{i_k})$ is uniform on $[M]^k$.*

Explicit constructions of t -wise independent family of seed length $O(t \log NM)$ are well known. We state two useful bounds on t -wise independent random variables [Ind01, FPS11]. In the rest of this work, we use $\mathbb{E}_{X:t\text{-wise}}[f(X)]$ denote the expectation of $f(X)$ when X is t -wise independent.

Lemma 2.4. *Let $\sigma : [N] \rightarrow [M]$ a function sampled from t -wise-independence, and let $B \subseteq [N]$ be a subset with $b := |B|$. Then, for $\Pr_\sigma[\min \sigma(B) > \theta]$, the following two estimates hold:*

1. $\Pr_\sigma[\min \sigma(B) > \theta] = (1 - \theta/M)^b \pm (b \cdot \frac{\theta}{M})^t / t!$;
2. $\Pr_\sigma[\min \sigma(B) > \theta] \leq \left(\frac{C_t \cdot t}{b \cdot \theta/M} \right)^{t/2}$, where C_t is a universal constant.

Randomness Extractor. Our construction will be based on randomness extractors. The min-entropy of a random source X is defined as $H_\infty(X) = \log \max_x 1/\Pr[X = x]$.

Definition 2.5. $\text{Ext} : \{0, 1\}^n \times \{0, 1\}^d \rightarrow \{0, 1\}^m$ is a (k, ε) -randomness extractor if for any source X of min-entropy k , $\text{Ext}(X, U_d) \approx_\varepsilon U_m$.

Our PRG needs a randomness extractors with an extra property: $\text{Ext}(U_n, s) = U_m$, whose proof is deferred to [Section 5](#).

Lemma 2.6. Given any n and $k < n$, for any error ε , there exists a randomness extractor $\text{Ext} : \{0, 1\}^n \times \{0, 1\}^d \rightarrow \{0, 1\}^m$ with $m = k/2$ and $d = O(\log(n/\varepsilon))$. Moreover, Ext satisfies an extra property: $\text{Ext}(U_n, s) = U_m$ for any fixed seed s .

Pseudorandomness for combinatorial rectangles. PRGs for combinatorial rectangles have been extensively studied in the past few decades [[ASWZ96](#), [LLSZ97](#), [Lu02](#), [GMR⁺12](#), [GY20](#)]. Our constructions of min-wise hash are based on these PRGs and their techniques. We state the state-of-the-art here by Gopalan and Yehudayoff, i.e., Theorem 1.9 in [[GY20](#)].

Theorem 2.7. For any error ε , dimension N , and alphabet $[M]$, there exists a PRG of seed length $O\left(\log\left(\frac{M \log N}{\varepsilon}\right) \cdot \log \log(M/\varepsilon)\right)$ that fools combinatorial rectangles in $[M]^N$ within additive error ε .

Based on the reduction by Saks, Srinivasan, Zhou and Zuckerman [[SSZZ00](#)], this provides a construction of min-wise hash of $O(\log NM \log \log NM)$ bits and polynomially small errors. For completeness, we provide this reduction in [Appendix A](#).

A direct corollary of [Theorem 2.7](#) provides a PRG of seed length $O(\log NM)$ and almost polynomially small error $2^{-O\left(\frac{\log NM}{\log \log NM}\right)}$. This is based on reductions between PRGs by Lu [[Lu02](#)]. Specifically, Lu constructed a PRG for combinatorial rectangles via a sequence of reductions. In particular, Lemma 3 in [[Lu02](#)] uses $O(\log(NM/\varepsilon))$ random bits to reduce the original problem to a problem in $[1/\varepsilon^{O(1)}]^{1/\varepsilon^{O(1)}}$ within error ε . Since the PRG in [Theorem 2.7](#) fools combinatorial rectangles in $[1/\varepsilon^{O(1)}]^{1/\varepsilon^{O(1)}}$ within error ε and $O(\log(1/\varepsilon) \log \log(1/\varepsilon))$ bits. After setting $\varepsilon = 2^{-C \cdot \frac{\log NM}{\log \log NM}}$, this gives a PRG of seed length $O(\log NM)$.

Corollary 2.8. For any constant C and error $\varepsilon = 2^{-C \cdot \frac{\log NM}{\log \log NM}}$, there exists an explicit PRG of seed length $O_C(\log NM)$ that fools combinatorial rectangles within additive error ε .

3 Min-wise Hash of Polynomial Size

The goal of this section is to provide a construction of explicit min-wise hash family with polynomial size and *nearly* polynomial small error.

Recall the definition that $\mathcal{H} = \{h_i : [N] \rightarrow [M]\}$ is a min-wise hash family of error δ , if for any $X \subseteq [N]$ and any $y \in X$, $\Pr_{h \sim \mathcal{H}}[h(y) < \min_{x \in X \setminus \{y\}} h(x)] = \frac{1+\delta}{|X|}$. Since the multiplicative error δ would be $\Omega(1/N)$, we assume the size of the alphabet $M = (N/\delta)^{O(1)} = N^{O(1)}$ for convenience.

Theorem 3.1. Given any N and any constant C , there exists an explicit min-wise hash family whose seed length is $O_C(\log N)$ and multiplicative error is $\delta = 2^{-C \cdot \frac{\log N}{\log \log N}}$.

Besides the Nisan-Zuckerman PRG, our construction uses several extra ingredients. The first one is to use the extractor in [Lemma 2.6](#) with $\text{Ext}(U_n, s) = U_m$ and an asymptotic optimal error. The second idea is to generate random seeds in the Nisan-Zuckerman PRG by the PRG of combinatorial rectangles in [Theorem 2.7](#), which is motivated by a domain reduction of Lu [\[Lu02\]](#).

Now we describe the construction of our hash family $\mathcal{H} = \{h_i : [N] \rightarrow [M]\}$.

Min-wise Hash Family: dimension N , error $2^{-C \cdot \frac{\log N}{\log \log N}}$, and alphabet $N^{O(1)}$

1. Set $\ell := 2^t$ for $t := \frac{\log N}{\log \log N}$ and pick large constants C_e , C_s and C_g such that $C_e > C_s > C_g > C$.
2. Sample a C_g -wise independent function $g : [N] \rightarrow [\ell]$ as the allocation of $[N]$ into ℓ buckets.
3. Apply PRG_1 in [Theorem 2.7](#) fooling combinatorial rectangles of seed length $O(\log N)$ to generate ℓ pseudorandom seeds s_1, \dots, s_ℓ in $\{0, 1\}^{C_e \cdot t}$ with an additive error $2^{-(C_s + C_e) \cdot t - 2}$.
4. Sample a random source $w \sim \{0, 1\}^{C_e \cdot \log N}$ and apply the extractor in [Lemma 2.6](#), $\text{Ext} : \{0, 1\}^{C_e \cdot \log N} \times \{0, 1\}^{C_e \cdot t} \rightarrow \{0, 1\}^{0.3C_e \cdot \log N}$ of min-entropy $0.6C_e \cdot \log N$ and error $\text{ExtErr} = 2^{-C_s \cdot t}$, to w and s_1, \dots, s_ℓ . For every $i \in [\ell]$, let $z_i := \text{Ext}(w, s_i)$.
5. Define a hash family $\mathcal{G} = \{G_1, \dots, G_{N^{0.3C_e}} : [N] \rightarrow [M]\}$ of size $N^{0.3C_e}$ to be the direct sum of a $(0.1C_s + 1)$ -wise independent function in $[M]^N$ and PRG_2 of error $2^{-C_s \cdot t}$ for combinatorial rectangles in $[M]^N$ from [Corollary 2.8](#).
6. Use z_i to pick a function in \mathcal{G} and denote it by $\sigma_i := G_{z_i}$.
7. Finally, let hash function $h(x) = \sigma_{g(x)}(x)$ for every $x \in [N]$.

We finish the proof of [Theorem 3.1](#) in this section. Firstly, we have the following properties from the allocation function g . For ease of exposition, let $j := g(y)$ and $B_i := \{x \in X \mid y : g(x) = i\}$ in the rest of this section.

Lemma 3.2. *Let C_g be a sufficiently large constant compared to C . Then C_g -wise independent function $g : [N] \rightarrow [\ell]$ guarantees that (recall $\ell = 2^{\frac{\log N}{\log \log N}}$):*

1. When $|X| \leq \ell^{0.9}$, with probability $1 - 1/\ell^{3C}$, $|B_i| \leq C_g$ for all $i \in [\ell]$.
2. When $|X| \in (\ell^{0.9}, \ell^{1.1})$, with probability $1 - 1/\ell^{3C}$, all buckets have $|B_i| \leq 2\ell^{0.1}$.
3. When $|X| \geq \ell^{1.1}$, with probability $1 - |X|^{-3C}$, all buckets satisfy $|B_i| = (1 \pm 0.1) \cdot |X|/\ell$.

The key point is that since $\ell = 2^t = 2^{\frac{\log N}{\log \log N}}$, the failure probability of each case is small enough compared to the error $\delta/|X| = 2^{-C \cdot \frac{\log N}{\log \log N}}/|X|$. Thus, we can assume that all properties in [Lemma 3.2](#) hold.

To calculate $\Pr_{h \sim \mathcal{H}}[h(y) < \min h(X \setminus y)]$, we express this as

$$\begin{aligned} \Pr_{h \sim \mathcal{H}}[h(y) < \min h(X \setminus y)] &= \sum_{\theta \in [M]} \Pr_{h \sim \mathcal{H}}[h(y) = \theta \wedge \min h(X \setminus y) > \theta] \\ &= \sum_{\theta \in [M]} \Pr_{w \sim U, (s_1, \dots, s_\ell) \sim \text{PRG}_1}[(\sigma_j(y) = \theta \wedge \min \sigma_j(B_j) > \theta) \wedge (\min \sigma_i(B_i) > \theta, \forall i \neq j)], \end{aligned} \quad (3.1)$$

where function $\sigma_i := G_{z_i}$ for $z_i := \text{Ext}(w, s_i)$. Our analysis will bound each term of (3.1).

Our second step is to bound the multiplicative error when the seeds s_1, \dots, s_ℓ of extractors are sampled independently and uniformly from $\{0, 1\}^{C_e \cdot t}$ like the Nisan-Zuckerman PRG.

Lemma 3.3. *Let \mathcal{H}' be the hash function family after replacing s_1, \dots, s_ℓ by independent random samples in $\{0, 1\}^{C_e \cdot t}$, instead of applying PRG_1 . The (multiplicative) error of \mathcal{H}' is at most $2^{-2C \cdot t}$.*

$$\begin{aligned} \Pr_{h' \sim \mathcal{H}'}[h'(y) < \min h'(X \setminus y)] &= \sum_{\theta \in [M]} \Pr_{h' \sim \mathcal{H}'}[h'(y) = \theta \wedge \min h'(X \setminus y) > \theta] \\ &= (1 \pm 2^{-2C \cdot t}) \cdot \Pr_{\sigma \sim U}[\sigma(y) < \min \sigma(X \setminus y)]. \end{aligned} \quad (3.2)$$

Next we consider the error when the hash family \mathcal{H} uses correlated seeds s_1, \dots, s_ℓ generated by PRG_1 for combinatorial rectangles. We bound the error between h and h' as follows.

Claim 3.4. *For any fixed g , let $j := g(y)$ and $B_j := \{x \in X \setminus y : g(x) = g(y)\}$. Then we define \mathcal{H}_g to be the hash family with this fixed allocation g and correlated seeds s_1, \dots, s_ℓ generated by PRG_1 and \mathcal{H}'_g to be the hash family with this fixed allocation g and independent seeds like Lemma 3.3.*

For any $\theta \in [M]$, the error between \mathcal{H}_g and \mathcal{H}'_g is at most

$$\begin{aligned} &\left| \Pr_{h \sim \mathcal{H}_g}[h(y) = \theta \wedge \min h(X \setminus y) > \theta] - \Pr_{h' \sim \mathcal{H}'_g}[h'(y) = \theta \wedge \min h'(X \setminus y) > \theta] \right| \\ &\leq \Pr_{\sigma \sim \mathcal{G}}[\sigma(y) = \theta \wedge \min \sigma(B_j) > \theta] \cdot 2^{-C_s \cdot t}. \end{aligned} \quad (3.3)$$

The term in (3.3) is the error introduced by PRG_1 , which is multiplicative with respect to $\Pr[\sigma(y) = \theta]$. So the last piece is to show the total error of (3.3) over θ , $\sum_{\theta \in [M]} \Pr_{\sigma \sim \mathcal{G}}[\sigma(y) = \theta \wedge \min \sigma(B_j) > \theta] \cdot 2^{-C_s \cdot t}$, is bounded by $o(\delta)/|X|$. The observation here is that $\sum_{\theta \in [M]} \Pr_{\sigma \sim \mathcal{G}}[\sigma(y) = \theta \wedge \min \sigma(B_j) > \theta]$ is the exact probability of event $(\sigma(y) < \min \sigma(B_j))$ under $(0.1C_s + 1)$ -wise independence since \mathcal{G} is $(0.1C_s + 1)$ -wise independent. Thus by the classical result of Indyk [Ind01], this part is bounded by $O(1)/|B_j|$.

We finish the proof of Theorem 3.1 in Section 3.1. Then we show the proofs of Lemma 3.2, Lemma 3.3 and Claim 3.4 in Section 3.2, Section 3.3 and Section 3.4 separately.

3.1 Proof of Theorem 3.1

We continue the calculation of (3.1):

$$\sum_{\theta \in [M]} \Pr_{w \sim U, (s_1, \dots, s_\ell) \sim \text{PRG}_1}[(\sigma_j(y) = \theta \wedge \min \sigma_j(B_j) > \theta) \wedge (\min \sigma_i(B_i) > \theta, \forall i \neq j)].$$

We first fix the allocation g . Then by Claim 3.4, the above is equal to

$$\sum_{\theta \in [M]} \Pr_{h' \sim \mathcal{H}'_g}[h'(y) = \theta \wedge \min h'(X \setminus y) > \theta] \pm \sum_{\theta \in [M]} \Pr_{\sigma \sim \mathcal{G}}[\sigma(y) = \theta \wedge \min \sigma(B_j) > \theta] \cdot 2^{-C_s \cdot t}.$$

Lemma 3.3 shows that the sum of the first term over allocations g is $(1 \pm 2^{-2C \cdot t}) \cdot \Pr_{\sigma \sim U}[\sigma(y) < \min \sigma(X \setminus y)]$. Then observe that given g , the second term becomes

$$2^{-C_s \cdot t} \cdot \Pr_{\sigma: (0.1C_s + 1)\text{-wise}}[\sigma(y) < \min \sigma(B_j)] = 2^{-C_s \cdot t} \cdot \frac{O(1)}{|B_j| + 1}, \quad (3.4)$$

by Indyk's result that $O(\log(1/\varepsilon))$ -wise independence is a min-wise hash family with error ε (choosing ε as a constant here). For completeness, we provide a formal statement here.

Theorem 3.5 (Theorem 1.1 in [Ind01]). *There exists a constant c such that for any $\varepsilon > 0$, $c \cdot \log(1/\varepsilon)$ -wise independent function from $[N]$ to $[M]$ is a min-wise hash family with error ε when $M = \Omega(N/\varepsilon)$.*

Returning to the second term, if $|X| < 2^{0.5C_s \cdot t}$, then $2^{-C_s \cdot t} < 2^{-0.5C_s \cdot t}/|X|$. Otherwise, $|X| \geq 2^{0.5C_s \cdot t} > \ell^{1.1}$. By **Lemma 3.2**, with probability $1 - |X|^{-3C}$, we have $|B_j| = (1 \pm 0.1) \cdot |X|/\ell$ in this case. And (3.4) becomes $\leq 2^{-C_s \cdot t} \cdot \frac{O(1)}{|X|/\ell} = 2^{-(C_s-1) \cdot t} \cdot \frac{O(1)}{|X|}$.

Finally, summing over all allocations g , we have

$$\begin{aligned} \Pr_{h \sim \mathcal{H}}[h(y) < \min h(X \setminus y)] &= \Pr_{\sigma \sim U}[\sigma(y) < \min \sigma(X \setminus y)] \cdot (1 \pm 2^{-2C \cdot t}) \pm \frac{O(1) \cdot 2^{-(C_s-1) \cdot t}}{|X|} \pm 1/|X|^{3C} \\ &= \Pr_{\sigma \sim U}[\sigma(y) < \min \sigma(X \setminus y)] \cdot (1 \pm 2^{-C \cdot t}), \end{aligned}$$

as C_s is a large constant compared to C .

3.2 Proof of **Lemma 3.2**

For convenience, set $r := |X| - 1$ in this proof. We prove these three cases separately.

1. For $|X| \leq \ell^{0.9}$, given $C_g \gg C$, we have

$$\Pr_{g: C_g\text{-wise}}[|B_i| \geq C_g] \leq \binom{r}{C_g} \cdot (1/\ell)^{C_g} \leq (r/\ell)^{C_g} \leq 1/\ell^{0.1C_g} \leq 1/\ell^{3C+1}, \quad \forall i \in [\ell].$$

By a union bound, with probability $1 - 1/\ell^{3C}$, $|B_i| \leq C_g$ for all buckets.

2. For $|X| \in (\ell^{0.9}, \ell^{1.1})$, let us fix a bin $i \in [\ell]$ and define $Z_x := \mathbf{1}(g(x) = i)$. Thus $|B_i| = \sum_{x \in X \setminus y} Z_x$ and $\mathbb{E}_{g: C_g\text{-wise}}[|B_i| - \mathbb{E}[|B_i|]]^{C_g} \leq O(C_g \cdot r/\ell)^{C_g/2}$, and

$$\Pr_{g: C_g\text{-wise}}[|B_i| - \mathbb{E}[|B_i|] \geq \ell^{0.1}] \leq \frac{O(C_g \cdot r/\ell)^{C_g/2}}{\ell^{0.1C_g}} \leq \frac{O(C_g \cdot \ell^{0.1})^{C_g/2}}{\ell^{0.1C_g}} = \frac{O_{C_g}(1)}{\ell^{0.05C_g}} \leq 1/\ell^{3C+1}.$$

After a union bound, with probability $1 - 1/\ell^{3C}$, $|B_i| \leq \ell^{0.1} + \mathbb{E}[|B_i|] \leq 2\ell^{0.1}$ for all i .

3. For $|X| \geq \ell^{1.1}$, similar to the above analysis, we fix $i \in [\ell]$ and define $Z_x := \mathbf{1}(g(x) = i)$. However, the deviation depends more on r :

$$\Pr_{g: C_g\text{-wise}}[|B_i| - \mathbb{E}[|B_i|] \geq 0.1 \cdot r/\ell] \leq \frac{O(C_g \cdot r/\ell)^{C_g/2}}{(0.1 \cdot r/\ell)^{C_g}} \leq \frac{O_{C_g}(1)}{(r/\ell)^{C_g/2}} \leq \frac{O_{C_g}(1)}{(r^{0.05})^{C_g/2}} \leq 1/r^{3C+1}.$$

Using a union bound, with probability $1 - r^{-3C}$, $|B_i| = (1 \pm 0.1) \cdot r/\ell$ for every $i \in [\ell]$.

3.3 Proof of Lemma 3.3

This proof relies on the extra property of our extractors in Lemma 2.6 and the fact that permuting the input bits will not affect the Nisan-Zuckerman PRG.

This proof has two parts. In the first part, given $X \subseteq [N]$, $y \in X$, $\theta \in [M]$ and the allocation mapping g , let $j := g(y)$. We will prove

$$\begin{aligned} \Pr_{h' \sim \mathcal{H}'} [h'(y) = \theta \wedge \min h'(X \setminus y) > \theta] &= \frac{1}{M} \Pr_{\sigma: 0.1C_s\text{-wise}} [\min \sigma(B_j) > \theta] \\ &\cdot \prod_{i \neq j} \left((1 - \theta/M)^{|B_i|} \pm 2 \cdot 2^{-C_s \cdot t} \right) \pm \ell \cdot N^{-0.4C_e}. \end{aligned} \quad (3.5)$$

In the second part, we bound the summation of (3.5) over all $\theta \in [M]$ and allocations g as $(1 \pm 2^{-2C \cdot t}) \cdot \Pr_{\sigma \sim U} [\sigma(y) < \min \sigma(X \setminus y)]$. Note that Lemma 3.2 tells we can ignore bad allocations.

3.3.1 The First Part

We show (3.5) via the Nisan-Zuckerman framework. One subtle thing is that the extractor error $\text{ExtErr} = 2^{-C_s \cdot t}$ in (3.5) is multiplicative with respect to $\Pr_{\sigma: (0.1C_s + 1)\text{-wise}} [\sigma(y) = \theta] = 1/M$, different from applying the PRG fooling combinatorial rectangles directly. This is shown by permuting the order of s_1, \dots, s_ℓ such that $z_j := \text{Ext}(w, s_j)$ is the first input of the read-once branching program.

Given the allocation g , we consider a width-2 length- ℓ read-once branching program P whose input alphabet is $\{0, 1\}^{0.3C_e \cdot \log N}$ corresponding to z_1, \dots, z_ℓ . Because of the definition $B_i := \{x \in X \setminus y : g(x) = i\}$ and $h'(i) := \sigma_{g(i)}(i)$ for $\sigma_1 := G_{z_1}, \dots, \sigma_\ell = G_{z_\ell}$, we fix g and rewrite $(h'(y) = \theta \wedge \min h'(X \setminus y) > \theta)$ as the conjunction of ℓ events depending on z_1, \dots, z_ℓ :

$$\underbrace{(\sigma_j(y) = \theta \wedge \min \sigma_j(B_j) > \theta)}_{A_1} \wedge \dots \wedge \underbrace{(\min \sigma_\ell(B_\ell) > \theta)}_{A_{\ell-j+1}} \wedge \dots \wedge \underbrace{(\min \sigma_{j-1}(B_{j-1}) > \theta)}_{A_\ell}.$$

First of all, choosing this order will guarantee that $\sigma_j(y) = \theta$ is in the first event A_1 . Secondly, this is a ROBP of width-2 with input z_1, \dots, z_ℓ .

Since $\sigma_i = G_{z_i}$ for $z_i = \text{Ext}(w, s_i)$,

$$\begin{aligned} &\Pr_{w \sim U, (s_1, \dots, s_\ell) \sim U: z_i = \text{Ext}(w, s_i)} [(\sigma_j(y) = \theta \wedge \min \sigma_j(B_j) > \theta) \wedge (\min \sigma_i(B_i) > \theta, \forall i \neq j)] \\ &= \Pr_{w \sim U, (s_1, \dots, s_\ell) \sim U: z_i = \text{Ext}(w, s_i)} [A_1 \wedge A_2 \wedge \dots \wedge A_\ell] \end{aligned} \quad (3.6)$$

has correlated functions $\sigma_1, \dots, \sigma_\ell$ generated from the Nisan-Zuckerman PRG with an extractor error ExtErr . However, because the *first* input $z_j = \text{Ext}(w, s_j)$ is uniform by Lemma 2.6, the first function $\sigma_j = G_{z_j}$ guarantees that the first event happens as same as a uniform sampling in \mathcal{G} :

$$\Pr_{w, s_j} [A_1] = \Pr_{w, s_j} [\sigma_j(y) = \theta \wedge \min \sigma_j(B_j) > \theta] = \Pr_{\sigma \sim \mathcal{G}} [\sigma(y) = \theta \wedge \min \sigma(B_j) > \theta].$$

Then we consider the rest events. Similar to the analysis of the Nisan-Zuckerman PRG, there are two cases for each $i = 2, 3, \dots, \ell$:

1. When $\Pr_{w, s_j \dots} [A_1 \wedge \dots \wedge A_{i-1}] \geq 2^{-0.4C_e \cdot \log N}$, consider the distribution of w conditioned upon $A_1 \wedge \dots \wedge A_{i-1}$. Note that the conditioning only increases the probability of each value of w by a factor of at most $2^{0.4C_e \cdot \log N}$. Hence this conditional distribution has min-entropy at least $0.6C_e \cdot \log N$. Then by the property of the extractor, we have

$$\Pr_{w, s_j \dots} [A_i \mid A_1 \wedge \dots \wedge A_{i-1}] = \Pr_{z_{j+i-1} \sim U} [A_i] \pm \text{ExtErr}.$$

Hence

$$\begin{aligned}\Pr_{w,s_j,\dots}[A_1 \wedge \dots \wedge A_i] &= \Pr_{w,s_j,\dots}[A_1 \wedge \dots \wedge A_{i-1}] \cdot \Pr_{w,s_j,\dots}[A_i \mid A_1 \wedge \dots \wedge A_{i-1}] \\ &= \Pr_{w,s_j,\dots}[A_1 \wedge \dots \wedge A_{i-1}] \cdot \left(\Pr_{z_{j+i-1} \sim U}[A_i] \pm \text{ExtErr} \right).\end{aligned}$$

2. Otherwise, $\Pr_{w,s_j,\dots}[A_1 \wedge \dots \wedge A_{i-1}] < 2^{-0.4C_e \cdot \log N}$ indicates $\Pr_{w,s_j,\dots}[A_1 \wedge \dots \wedge A_i] \leq 2^{-0.4C_e \cdot \log N}$.

Combining two cases together, we can write the the acceptance of the first i events as

$$\Pr_{w,s_j,\dots}[A_1 \wedge \dots \wedge A_i] = \Pr_{w,s_j,\dots}[A_1 \wedge \dots \wedge A_{i-1}] \cdot \left(\Pr_{z_{j+i-1} \sim U}[A_i] \pm \text{ExtErr} \right) \pm 2^{-0.4C_e \cdot \log N}.$$

Then by induction on i , the probability in (3.6) is expressed as

$$\begin{aligned}& \Pr_{z_j \sim U}[A_1] \cdot \left(\Pr_{z_{j+1} \sim U}[A_2] \pm \text{ExtErr} \right) \cdots \left(\Pr_{z_{j-1} \sim U}[A_\ell] \pm \text{ExtErr} \right) \pm \ell \cdot 2^{-0.4C_e \cdot \log N} \\ &= \Pr_{\sigma \sim \mathcal{G}}[\sigma(y) = \theta \wedge \min \sigma(B_j) > \theta] \cdot \prod_{i \neq j} \left(\Pr_{\sigma \sim \mathcal{G}}[\min \sigma(B_i) > \theta] \pm \text{ExtErr} \right) \pm \ell \cdot N^{-0.4C_e}.\end{aligned}$$

Since \mathcal{G} is $(0.1C_s + 1)$ -wise independent, for A_1 , it follows that

$$\Pr_{\sigma \sim \mathcal{G}}[\sigma(y) = \theta \wedge \min \sigma(B_j) > \theta] = \frac{1}{M} \Pr_{\sigma: 0.1C_s\text{-wise}}[\min \sigma(B_j) > \theta].$$

\mathcal{G} is also the PRG for combinatorial rectangles with error $2^{-C_s \cdot t}$, so for the remaining events, we simplify their probabilities as

$$\Pr_{\sigma \sim \mathcal{G}}[\min \sigma(B_i) > \theta] \pm \text{ExtErr} = \Pr_{\sigma \sim U}[\min \sigma(B_i) > \theta] \pm 2^{-C_s \cdot t} \pm \text{ExtErr} = (1 - \theta/M)^{|B_i|} \pm 2 \cdot 2^{-C_s \cdot t}.$$

This shows (3.5):

$$\begin{aligned}\Pr_{h' \sim \mathcal{H}'}[h'(y) = \theta \wedge \min h'(X \setminus y) > \theta] &= \frac{1}{M} \Pr_{\sigma: 0.1C_s\text{-wise}}[\min \sigma(B_j) > \theta] \\ &\cdot \prod_{i \neq j} \left((1 - \theta/M)^{|B_i|} \pm 2 \cdot 2^{-C_s \cdot t} \right) \pm \ell \cdot N^{-0.4C_e}.\end{aligned}$$

3.3.2 The Second Part

Here we show the summation of (3.5) over all θ 's and allocations g equals $(1 \pm 2^{-2C \cdot t}) \cdot \Pr_{\sigma \sim U}[\sigma(y) < \min \sigma(X \setminus y)]$. This indicates $\Pr_{h' \sim \mathcal{H}'}[h'(y) < \min h'(X \setminus y)] = (1 \pm 2^{-2C \cdot t}) \cdot \Pr_{\sigma \sim U}[\sigma(y) < \min \sigma(X \setminus y)]$ by the first part of this proof and finishes the proof of Lemma 3.3 — \mathcal{H}' has a multiplicative error $2^{-2C \cdot t}$.

For convenience, set $\varepsilon := 2 \cdot 2^{-C_s \cdot t}$ in this proof. Our rough plan is to apply the first approximation of Lemma 2.4 for small θ to show that

$$\Pr_{\sigma: 0.1C_s\text{-wise}}[\min \sigma(B_j) > \theta] \cdot \prod_{i \neq j} \left((1 - \theta/M)^{|B_i|} \pm \varepsilon \right) \approx (1 - \theta/M)^{|X|-1}.$$

For large θ such that $(1 - \theta/M)^{|X|-1}$ is tiny, we choose a suitable subset $T \subseteq [\ell] \setminus j$, and use the following fact to bound the tail probability:

$$\Pr_{\sigma: 0.1C_s\text{-wise}}[\min \sigma(B_j) > \theta] \cdot \prod_{i \neq j} \left((1 - \theta/M)^{|B_i|} \pm \varepsilon \right) \leq \prod_{i \in T} \left((1 - \theta/M)^{|B_i|} + \varepsilon \right).$$

The actual calculation depends on the size of X . According to [Lemma 3.2](#), we will split the calculations into three cases: (1) $|X| \leq \ell^{0.9}$; (2) $|X| \in (\ell^{0.9}, \ell^{1.1})$; (3) $|X| \geq \ell^{1.1}$.

Moreover, since we have assumed $M = (N/\delta)^{O(1)}$ such that $\Pr_{\sigma \sim U}[\sigma(y) < \min \sigma(X \setminus y)] \approx 1/|X|$, we treat multiplicative errors as additive errors multiplied by a factor of $|X|$ in this analysis.

The first case of $|X| \leq \ell^{0.9}$. We assume that each bucket has at most C_g elements in $X \setminus y$ according to the first property of [Lemma 3.2](#). The corresponding failure probability will change the final multiplicative error by at most $|X|/\ell^{3C} \leq 1/\ell^{3C-0.9} < 2^{-2.5C \cdot t}$.

As $C_s \gg C_g$, we assume $0.1C_s > C_g \geq |B_j|$. Thus $\Pr_{\sigma: 0.1C_s\text{-wise}}[\min \sigma(B_j) > \theta] = (1 - \theta/M)^{|B_j|}$ and

$$\Pr_{\sigma: 0.1C_s\text{-wise}}[\min \sigma(B_j) > \theta] \cdot \prod_{i \neq j} \left((1 - \theta/M)^{|B_i|} \pm \varepsilon \right) = \prod_{i \in [\ell]} \left((1 - \theta/M)^{|B_i|} \pm \varepsilon \right).$$

When $(1 - \theta/M)^{C_g} > 2^{-0.5C_s \cdot t}$, we simplify the above additive error $\varepsilon = 2 \cdot 2^{-C_s \cdot t}$ as

$$(1 - \theta/M)^{|B_i|} \pm \varepsilon = (1 - \theta/M)^{|B_i|} \cdot (1 \pm 2 \cdot 2^{-0.5C_s \cdot t}), \quad \forall i \in [\ell].$$

Hence

$$\sum_{(1-\theta/M)^{C_g} > 2^{-0.5C_s \cdot t}} \prod_{i \in [\ell]} \left((1 - \theta/M)^{|B_i|} \pm \varepsilon \right) = (1 \pm 4 \cdot 2^{-(0.5C_s-1) \cdot t}) \cdot \sum_{(1-\theta/M)^{C_g} > 2^{-0.5C_s \cdot t}} (1 - \theta/M)^{|X|-1}.$$

Otherwise, θ is large enough such that $(1 - \theta/M)^{C_g} \leq 2^{-0.5C_s \cdot t}$. Note that each term in the summation does not exceed 1, and the number of such θ 's is at most $M \cdot 2^{-0.5C_s \cdot t/C_g}$. Thus we simply bound the sum over such θ 's as

$$\frac{1}{M} (1 - \theta/M)^{|X|-1} = \frac{1}{M} \sum_{(1-\theta/M)^{C_g} \leq 2^{-0.5C_s \cdot t}} \prod_{i \in [\ell]} \left((1 - \theta/M)^{|B_i|} \pm \varepsilon \right) \leq 2^{-0.5C_s \cdot t/C_g}.$$

Now we have

$$\begin{aligned} & \Pr_{h' \sim \mathcal{H}'}[h'(y) = \theta \wedge \min h'(X \setminus y) > \theta] \\ &= \frac{1}{M} \left(\sum_{(1-\theta/M)^{C_g} > 2^{-0.5C_s \cdot t}} \prod_{i \in [\ell]} \left((1 - \theta/M)^{|B_i|} \pm \varepsilon \right) + \sum_{(1-\theta/M)^{C_g} \leq 2^{-0.5C_s \cdot t}} \prod_{i \in [\ell]} \left((1 - \theta/M)^{|B_i|} \pm \varepsilon \right) \right) \pm 1/\ell^{3C} \\ &= \frac{1 \pm 4 \cdot 2^{-(0.5C_s-1) \cdot t}}{M} \cdot \sum_{(1-\theta/M)^{C_g} > 2^{-0.5C_s \cdot t}} (1 - \theta/M)^{|X|-1} \pm 2 \cdot 2^{-0.5C_s \cdot t/C_g} \pm 1/\ell^{3C}. \end{aligned}$$

Compared to the fair probability $\sum_{\theta \in [M]} \frac{1}{M} \cdot (1 - \theta/M)^{|X|-1}$, the multiplicative error (w.r.t. $1/|X|$) of (3.5) is at most

$$4 \cdot 2^{-(0.5C_s-1) \cdot t} + \frac{1}{\ell^{3C-0.9}} + |X| \cdot 2^{-0.5C_s \cdot t/C_g} \leq 2^{-(0.5C_s-1) \cdot t+2} + 2^{-2.5C \cdot t} + 2^{-(0.5C_s/C_g-1) \cdot t} < 2^{-2C \cdot t}.$$

In the last step, We choose $C_s \gg C_g$ such that $0.5C_s/C_g > 3C$.

The second case of $|X| \in (\ell^{0.9}, \ell^{1.1})$. Similarly, we assume $\max_{i \in [\ell]} |B_i|$ is at most $2\ell^{0.1}$ by [Lemma 3.2](#). The failure probability only affects the multiplicative error by at most $|X|/\ell^{3C} \leq 1/\ell^{3C-1.1} < 2^{-2.5C \cdot t}$.

When $\theta \leq 0.5M \cdot \ell^{-0.2}$, there comes $(1 - \theta/M)^{|B_i|} \geq 1 - |B_i| \cdot \theta/M \geq 1 - \ell^{-0.1} \geq 0.5$, $\forall i \in [\ell]$. We apply the first statement in [Lemma 2.4](#) to estimate $\Pr_{\sigma:0.1C_s\text{-wise}}[\min \sigma(B_j) > \theta]$:

$$\begin{aligned} \Pr_{\sigma:0.1C_s\text{-wise}}[\min \sigma(B_j) > \theta] &= (1 - \theta/M)^{|B_j|} \pm \left(|B_j| \cdot \frac{\theta}{M} \right)^{0.1C_s} \\ &= (1 - \theta/M)^{|B_j|} \cdot \left(1 \pm \frac{2}{\ell^{0.01C_s}} \right). \end{aligned}$$

For the remaining buckets, we have

$$(1 - \theta/M)^{|B_i|} \pm \varepsilon = (1 - \theta/M)^{|B_i|} \cdot (1 \pm 2\varepsilon), \quad \forall i \neq j.$$

Therefore, for small θ with $\theta/M \leq 0.5\ell^{-0.2}$, it holds that

$$\begin{aligned} &\frac{1}{M} \sum_{\theta \leq 0.5M \cdot \ell^{-0.2}} \Pr_{\sigma:0.1C_s\text{-wise}}[\min \sigma(B_j) > \theta] \cdot \prod_{i \neq j} \left((1 - \theta/M)^{|B_i|} \pm \varepsilon \right) \\ &= \frac{1}{M} \sum_{\theta \leq 0.5M \cdot \ell^{-0.2}} (1 - \theta/M)^{|X|-1} \cdot \left(1 \pm \left(\frac{4}{\ell^{0.01C_s}} + 4(\ell-1)\varepsilon \right) \right) \\ &= \frac{1}{M} \sum_{\theta \leq 0.5M \cdot \ell^{-0.2}} (1 - \theta/M)^{|X|-1} \cdot (1 \pm 2^{-0.001C_s \cdot t}). \end{aligned}$$

Otherwise, when $\theta > 0.5M \cdot \ell^{-0.2}$, we show the tail summations are small in both cases of $\sigma \sim U$ and $h' \sim \mathcal{H}'$, leaving a negligible additive error.

Note that both $\frac{1}{M} \sum_{\theta > 0.5M \cdot \ell^{-0.2}} (1 - \theta/M)^{|X|-1}$ and $\frac{1}{M} \sum_{\theta > 0.5M \cdot \ell^{-0.2}} \Pr_{\sigma:0.1C_s\text{-wise}}[\min \sigma(B_j) > \theta] \cdot \prod_{i \neq j} \left((1 - \theta/M)^{|B_i|} \pm \varepsilon \right)$ are upper bounded by $\frac{1}{M} \sum_{\theta > 0.5M \cdot \ell^{-0.2}} \prod_{i \neq j} \left((1 - \theta/M)^{|B_i|} + \varepsilon \right)$. Hence, we focus on the latter one, $\frac{1}{M} \sum_{\theta > 0.5M \cdot \ell^{-0.2}} \prod_{i \neq j} \left((1 - \theta/M)^{|B_i|} + \varepsilon \right)$.

Consider the sizes of all buckets, say $|B_1|, \dots, |B_\ell|$, and define b to be the $S := C' \cdot \log \log N$ largest number among $|B_1|, \dots, |B_\ell|$ excluding $|B_j|$. Without loss of generality, assume $j = \ell$ and $2\ell^{0.1} \geq |B_1| \geq |B_2| \geq \dots \geq |B_{\ell-1}|$, then $b = |B_S|$.

We split the sum into two cases depending on whether $(1 - \theta/M)^b > \varepsilon \cdot \ell$.

1. For $(1 - \theta/M)^b > \varepsilon \cdot \ell$, we further simplify it as

$$\begin{aligned} &\frac{1}{M} \sum_{\theta > 0.5M \cdot \ell^{-0.2} \wedge (1 - \theta/M)^b > \varepsilon \cdot \ell} \prod_{i \neq j} \left((1 - \theta/M)^{|B_i|} + \varepsilon \right) \\ &\leq \frac{1}{M} \sum_{\theta > 0.5M \cdot \ell^{-0.2} \wedge (1 - \theta/M)^b > \varepsilon \cdot \ell} \prod_{i=S+1}^{\ell-1} (1 - \theta/M)^{|B_i|} \cdot (1 + 1/\ell) \\ &\leq \frac{1}{M} \sum_{\theta > 0.5M \cdot \ell^{-0.2} \wedge (1 - \theta/M)^b > \varepsilon \cdot \ell} e \cdot (1 - \theta/M)^{|X|-1-(S+1) \cdot 2\ell^{0.1}} \\ &\leq \frac{e}{M} \sum_{\theta > 0.5M \cdot \ell^{-0.2} \wedge (1 - \theta/M)^b > \varepsilon \cdot \ell} (1 - \theta/M)^{0.9|X|} \\ &\leq \exp(1 - 0.9|X| \cdot 0.5\ell^{-0.2}) \leq \exp(-\ell^{0.6}). \end{aligned}$$

2. For $(1 - \theta/M)^b \leq \varepsilon \cdot \ell$, we bound it as

$$\begin{aligned}
& \frac{1}{M} \sum_{\theta > 0.5M \cdot \ell^{-0.2} \wedge (1-\theta/M)^b \leq \varepsilon \cdot \ell} \prod_{i \neq j} \left((1 - \theta/M)^{|B_i|} + \varepsilon \right) \\
& \leq \frac{1}{M} \sum_{\theta > 0.5M \cdot \ell^{-0.2} \wedge (1-\theta/M)^b \leq \varepsilon \cdot \ell} \prod_{i=1}^S (\varepsilon \cdot \ell + \varepsilon) \\
& \leq \frac{1}{M} \sum_{\theta > 0.5M \cdot \ell^{-0.2} \wedge (1-\theta/M)^b \leq \varepsilon \cdot \ell} \prod_{i=1}^S 2^{2-(C_s-1) \cdot t} \\
& \leq 2^{2S-(C_s-1) \cdot t \cdot S} = N^{-O(1)},
\end{aligned}$$

where we plug the definition of $S = C' \cdot \log \log N$ in the last step.

So, the total multiplicative error is bounded by

$$\frac{1}{\ell^{3C-1.1}} + 2^{-0.001C_s \cdot t} + |X| \cdot \left(\exp(-\ell^{0.6}) + N^{-O(1)} \right) < 2^{-2C \cdot t}.$$

The third case of $|X| \geq \ell^{1.1}$. The proof for this case is identical to the second case. Here, the max-load is bounded according to $1.1 \cdot |X|/\ell$ by [Lemma 3.2](#). The threshold of applying the tail bound would be between $\frac{|X|}{\ell} \cdot \frac{\theta}{M} \leq \ell^{-0.1}$ and $\frac{|X|}{\ell} \cdot \frac{\theta}{M} > \ell^{-0.1}$, while the rest calculation is very similar and we leave it in [Appendix B](#).

3.4 Proof of [Claim 3.4](#)

After fixing the allocation g , since we have proven $\Pr_{h' \sim \mathcal{H}'_g}[h'(y) = \theta \wedge \min h'(X \setminus y) > \theta]$ equals

$$\frac{1}{M} \Pr_{\sigma: 0.1C_s\text{-wise}} [\sigma(y) = \theta \wedge \min \sigma(B_j) > \theta] \cdot \prod_{i \neq j} \left((1 - \theta/M)^{|B_i|} \pm 2 \cdot 2^{-C_s \cdot t} \right) \pm \ell \cdot N^{-0.4C_e} \quad (3.7)$$

in [Section 3.3.1](#) (as equation (3.5)), this proof will show the error between $\Pr_{h \sim \mathcal{H}_g}[h(y) = \theta \wedge \min h(X \setminus y) > \theta]$ and (3.7) is at most $\Pr_{\sigma \sim \mathcal{G}}[\sigma(y) = \theta \wedge \min \sigma(B_j) > \theta] \cdot 2^{-C_s \cdot t}$.

We need the following property of s_1, \dots, s_ℓ when they are generated by PRG_1 .

Fact 3.6. *Those random seeds s_1, \dots, s_ℓ satisfy that for any fixed $j \in [\ell]$, any string $\alpha \in \{0, 1\}^{C_e \cdot t}$, and any ℓ functions $f_1, \dots, f_\ell : \{0, 1\}^{C_e \cdot t} \rightarrow \{0, 1\}$, it has*

$$\mathbb{E}_{(s_1, \dots, s_\ell) \sim \text{PRG}_1} \left[\prod_{i \neq j} f_i(s_i) \mid s_j = \alpha \right] = \prod_{i \neq j} \mathbb{E}_{s_i \sim U} [f_i(s_i)] \pm 2^{-C_s \cdot t}.$$

Proof. Note that both $\mathbf{1}(s_j = \alpha)$ and $\mathbf{1}(s_j = \alpha) \cdot \prod_{i \neq j} f_i(s_i)$ are combinatorial rectangles in $(\{0, 1\}^{C_e \cdot t})^\ell$. According to the property of PRG_1 , we have

$$\Pr_{s_j \sim \text{PRG}_1} [s_j = \alpha] = 2^{-C_e \cdot t} \pm 2^{-(C_s + C_e) \cdot t - 2}, \text{ and}$$

$$\mathbb{E}_{(s_1, \dots, s_\ell) \sim \text{PRG}_1} \left[\mathbf{1}(s_j = \alpha) \cdot \prod_{i \neq j} f_i(s_i) \right] = \mathbb{E}_{(s_1, \dots, s_\ell) \sim U} \left[\mathbf{1}(s_j = \alpha) \cdot \prod_{i \neq j} f_i(s_i) \right] \pm 2^{-(C_s + C_e) \cdot t - 2}.$$

So we can express the conditional expectation as

$$\begin{aligned} \mathbb{E}_{(s_1, \dots, s_\ell) \sim \text{PRG}_1} \left[\prod_{i \neq j} f_i(s_i) \mid s_j = \alpha \right] &= \frac{\mathbb{E}_{(s_1, \dots, s_\ell) \sim \text{PRG}_1} [\mathbf{1}(s_j = \alpha) \cdot \prod_{i \neq j} f_i(s_i)]}{\Pr_{s_j \sim \text{PRG}_1} [s_j = \alpha]} \\ &= \frac{\prod_{i \neq j} \mathbb{E}_{s_i \sim U} [f_i(s_i)] \pm 2^{-C_s \cdot t - 2}}{1 \pm 2^{-C_s \cdot t - 2}}. \end{aligned}$$

Note that $\prod_{i \neq j} \mathbb{E}_{s_i \sim U} [f_i(s_i)] \leq 1$, which tells that the additive error is at most

$$\frac{\prod_{i \neq j} \mathbb{E}_{s_i \sim U} [f_i(s_i)] + 2^{-C_s \cdot t - 2}}{1 - 2^{-C_s \cdot t - 2}} - \prod_{i \neq j} \mathbb{E}_{s_i \sim U} [f_i(s_i)] \leq \left(\frac{1}{1 - 2^{-C_s \cdot t - 2}} - 1 \right) + \frac{2^{-C_s \cdot t - 2}}{1 - 2^{-C_s \cdot t - 2}} \leq 2^{-C_s \cdot t},$$

as desired. \square

Now we are ready to finish the proof of [Claim 3.4](#) in this section. We rewrite the probability $\Pr_{h \sim \mathcal{H}_g} [h(y) = \theta \wedge \min h(X \setminus y) > \theta]$ as (recall $\sigma_i := G_{z_i}$ for $z_i := \text{Ext}(w, s_i)$)

$$\begin{aligned} &\Pr_{w \sim U, (s_1, \dots, s_\ell) \sim \text{PRG}_1} [(\sigma_j(y) = \theta \wedge \min \sigma_j(B_j) > \theta) \wedge (\min \sigma_i(B_i) > \theta, \forall i \neq j)] \\ &= \sum_{\alpha} \mathbb{E}_{w \sim U} \left[\Pr_{(s_1, \dots, s_\ell) \sim \text{PRG}_1} [(s_j = \alpha \wedge \sigma_j(y) = \theta \wedge \min \sigma_j(B_j) > \theta) \wedge (\min \sigma_i(B_i) > \theta, \forall i \neq j)] \right] \\ &= \sum_{\alpha} \mathbb{E}_{w \sim U} \left[\Pr_{s_j \sim \text{PRG}_1} [s_j = \alpha \wedge \sigma_j(y) = \theta \wedge \min \sigma_j(B_j) > \theta] \cdot \Pr_{(s_1, \dots, s_\ell) \sim \text{PRG}_1} [\min \sigma_i(B_i) > \theta, \forall i \neq j \mid s_j = \alpha] \right], \end{aligned} \tag{3.8}$$

where the last line applies the fact that only those $s_j = \alpha$ which make $(\sigma_j(y) = \theta \wedge \min \sigma_j(B_j) > \theta)$ true have contributions to the expectation.

For a fixed w , the first event $(\sigma_j(y) = \theta \wedge \min \sigma_j(B_j) > \theta)$ is determined since $s_j = \alpha$ is fixed. Then the rest events $\mathbf{1}(\min \sigma_i(B_i) > \theta, \forall i \neq j)$ for $\sigma_i = G_{z_i}$ and $z_i = \text{Ext}(w, s_i)$ constitute a combinatorial rectangle of s_1, \dots, s_ℓ in $(\{0, 1\}^{C_e \cdot t})^\ell$. Then by [Fact 3.6](#),

$$\Pr_{(s_1, \dots, s_\ell) \sim \text{PRG}_1} [\min \sigma_i(B_i) > \theta, \forall i \neq j \mid s_j = \alpha] = \prod_{i \neq j} \Pr_{s_i \sim U} [\min \sigma_i(B_i) > \theta] \pm 2^{-C_s \cdot t}.$$

So we apply this to simplify (3.8) as

$$\begin{aligned} &\sum_{\alpha} \mathbb{E}_{w \sim U} \left[\Pr_{s_j \sim \text{PRG}_1} [s_j = \alpha \wedge \sigma_j(y) = \theta \wedge \min \sigma_j(B_j) > \theta] \cdot \left(\prod_{i \neq j} \Pr_{s_i \sim U} [\min \sigma_i(B_i) > \theta] \pm 2^{-C_s \cdot t} \right) \right] \\ &= \sum_{\alpha} \Pr_{w \sim U, s_j \sim \text{PRG}_1} [s_j = \alpha \wedge \sigma_j(y) = \theta \wedge \min \sigma_j(B_j) > \theta] \\ &\quad \cdot \left(\Pr_{w \sim U, (s_i)_{i \neq j} \sim U} [\min \sigma_i(B_i) > \theta, \forall i \neq j \mid \sigma_j(y) = \theta \wedge \min \sigma_j(B_j) > \theta] \pm 2^{-C_s \cdot t} \right). \end{aligned} \tag{3.9}$$

We use properties of our extractor to simplify (3.9). Since $\text{Ext}(w, \alpha) = U$ when w is uniform and α is fixed, σ_j in the first event $(\sigma_j(y) = \theta \wedge \min \sigma_j(B_j) > \theta)$ conditioned on $s_j = \alpha$ is uniformly

sampled from \mathcal{G} . Thus this event holds with probability

$$\begin{aligned}
& \Pr_{w \sim U, s_j \sim \text{PRG}_1} [s_j = \alpha \wedge \sigma_j(y) = \theta \wedge \min \sigma_j(B_j) > \theta] \\
&= \Pr_{s_j \sim \text{PRG}_1} [s_j = \alpha] \cdot \Pr_{w \sim U, s_j \sim \text{PRG}_1} [\sigma_j(y) = \theta \wedge \min \sigma_j(B_j) > \theta \mid s_j = \alpha] \\
&= \Pr_{s_j \sim \text{PRG}_1} [s_j = \alpha] \cdot \Pr_{\sigma \sim \mathcal{G}} [\sigma(y) = \theta \wedge \min \sigma(B_j) > \theta].
\end{aligned}$$

Next we consider the term $\Pr_{w \sim U, (s_i)_{i \neq j} \sim U} [\min \sigma_i(B_i) > \theta, \forall i \neq j \mid \sigma_j(y) = \theta \wedge \min \sigma_j(B_j) > \theta]$ in (3.9). Following the same analysis in Section 3.3.1, it equals

$$\prod_{i \neq j} \left((1 - \theta/M)^{|B_i|} \pm 2 \cdot 2^{-C_s \cdot t} \right) \pm \ell \cdot N^{-0.4C_e}.$$

Combining all equations to simplify (3.9), we finish the proof:

$$\begin{aligned}
& \Pr_{h \sim \mathcal{H}_g} [h(y) = \theta \wedge \min h(X \setminus y) > \theta] \\
&= \sum_{\alpha} \Pr_{s_j \sim \text{PRG}_1} [s_j = \alpha] \cdot \Pr_{\sigma \sim \mathcal{G}} [\sigma(y) = \theta \wedge \min \sigma(B_j) > \theta] \cdot \left(\prod_{i \neq j} \left((1 - \theta/M)^{|B_i|} \pm 2 \cdot 2^{-C_s \cdot t} \right) \pm 2^{-C_s \cdot t} \right) \\
&= \frac{1}{M} \Pr_{\sigma: 0.1C_s\text{-wise}} [\min \sigma(B_j) > \theta] \cdot \prod_{i \neq j} \left((1 - \theta/M)^{|B_i|} \pm 2 \cdot 2^{-C_s \cdot t} \right) \pm \Pr_{\sigma \sim \mathcal{G}} [\sigma(y) = \theta \wedge \min \sigma(B_j) > \theta] \cdot 2^{-C_s \cdot t},
\end{aligned}$$

where the first term in the last line matches (3.7).

4 k -min-wise Hash

We use the second approach outlined in Section 1.2 to construct k -min-wise hash in this section. Recall the definition that $\mathcal{H} = \{h_i : [N] \rightarrow [M]\}$ is a k -min-wise hash family of error δ , if for any $X \subseteq [N]$ and $Y \subseteq X$ of size at most k , $\Pr_{h \sim \mathcal{H}} [\max h(Y) < \min h(X \setminus Y)] = \frac{1 \pm \delta}{\binom{|X|}{|Y|}}$. Without loss of generality, we assume $|Y| = k$ in this section.

Theorem 4.1. *Given any N , $k = \log^{O(1)} N$, and any constant C , there exists an explicit k -min-wise hash family such that its seed length is $O_C(k \log N)$ and its multiplicative error is $\delta = 2^{-C \cdot \frac{\log N}{\log \log N}}$.*

While our construction is in a similar framework of the construction in Section 3, there are several differences. The first one is to direct sum the output with an $O(k)$ -wise independent function h_0 in the last step. The second difference is in the analysis, Y could be in different buckets such that the first approach of guaranteeing its $\text{Ext}(w, s_j) = U$ for $j := g(y)$ does not work anymore. Instead of it, we will consider the conditional event $\Pr[\min h(X \setminus Y) > \theta \mid \max h(Y) = \theta]$ in this proof.

k -min-wise Hash Family: dimension N , $k = \log^{O(1)} N$, error $2^{-C \cdot \frac{\log N}{\log \log N}}$, and alphabet $N^{O(1)}$

1. Set $\ell := 2^t$ for $t := \frac{\log N}{\log \log N}$ and pick large constants C_e , C_s and C_g such that $C_e > C_s > C_g > C$.
2. Sample a $C_g \cdot k$ -wise independent function $g : [N] \rightarrow [\ell]$ as the allocation of $[N]$ into ℓ buckets.
3. Apply PRG_1 in [Theorem 2.7](#) fooling combinatorial rectangles of seed length $O(k \log N)$ to generate ℓ seeds s_1, \dots, s_ℓ in $\{0, 1\}^{C_e t}$ of error $2^{-(C_s + C_e) \cdot t \cdot k - 2}$.
4. Sample a source $w \sim \{0, 1\}^{10k \cdot C_e \log N}$ and apply an extractor $\text{Ext} : \{0, 1\}^{10k \cdot C_e \log N} \times \{0, 1\}^{C_e t} \rightarrow \{0, 1\}^{C_e \cdot \log N}$ from [Lemma 2.6](#) of min-entropy $6kC_e \log N$ and error $\text{ExtErr} = 2^{-C_s \cdot t}$ to w and s_1, \dots, s_ℓ : let $z_i := \text{Ext}(w, s_i), \forall i \in [\ell]$.
5. Let $\sigma_i := \text{PRG}_2(z_i)$ where PRG_2 fools combinatorial rectangles in $[M]^N$ with error $2^{-C_s \cdot t}$ from [Corollary 2.8](#).
6. Define $\varphi(x) := \sigma_{g(x)}(x)$ for every x .
7. Choose $h_0 : [N] \rightarrow [M]$ from a $(C_e + 1) \cdot k$ -wise independent hash family.
8. Output the direct sum $h := h_0 + \varphi$.

One remark is that Step 3 restrains $k = \log^{O(1)} N$ in order to guarantee the seed length of PRG_1 is $O(k \log N)$ bits. In the rest of this section, we prove [Theorem 4.1](#) and finish its analysis.

Similar to [Lemma 3.2](#), we have the following lemma about the allocation of X under g . Let $B_i := \{x \in X \setminus Y : g(x) = i\}$ be the elements in $X \setminus Y$ mapped to bucket i and $J := \{j_1, \dots, j_{k'}\}$ be the buckets in $[\ell]$ that contains elements in Y (under g), i.e., $J := \{g(y) : y \in Y\}$. And let $B_J := \bigcup_{i=1}^{k'} B_{j_i}$.

Lemma 4.2. *Let C_g be a sufficiently large constant compared to C . Then g guarantees that:*

1. When $|X| \leq \ell^{0.9}$, with probability $1 - \frac{1}{\ell^{3C \cdot |X|^k}}$, $|B_i| \leq C_g + 10 \cdot \frac{k \log |X|}{\log N / \log \log N}$ for all $i \in [\ell]$ and $|B_J| \leq C_g \cdot k$.
2. When $|X| \in (\ell^{0.9}, \ell^{1.1})$, with probability $1 - 1/\ell^{3C \cdot k}$, the max-load $\max_{i \in [\ell]} |B_i| \leq 2\ell^{0.1}$.
3. When $|X| \geq \ell^{1.1}$, with probability $1 - |X|^{-3C \cdot k}$, all buckets satisfy $|B_i| = (1 \pm 0.1) \cdot |X|/\ell$.

Similar to the analysis in [Theorem 3.1](#), the failure probability in [Lemma 4.2](#) is relatively small compared to $\delta / \binom{|X|}{|Y|} \approx \delta / |X|^k$. So we fix g and assume all properties in [Lemma 4.2](#) hold in this section. The proof of this lemma resembles [Lemma 3.2](#). We defer its proof to [Appendix C](#).

However, we can not assume the allocation of Y because we can not compare its failure probability with $1/|X|^k$. For example, the probability that a bucket has $\log \log N$ elements in Y is at least $1/N$ since the number of buckets $\ell = 2^{\frac{\log N}{\log \log N}}$. Since $1/|X|$ could be as small as $1/N$, this implies that even for Y as small as $\log \log N$, we could not guarantee Y is uniformly distributed over ℓ buckets.

We rewrite $\Pr_{h \sim \mathcal{H}}[\max h(Y) < \min h(X \setminus Y)]$ by enumerating $\theta = \max h(Y)$:

$$\Pr_{h \sim \mathcal{H}}[\max h(Y) < \min h(X \setminus Y)] = \sum_{\theta \in [M]} \Pr_{h \sim \mathcal{H}}[\max h(Y) = \theta \wedge \min h(X \setminus Y) > \theta] \quad (4.1)$$

Similar to [Claim 3.4](#) in the proof of [Theorem 3.1](#), we simplify (4.1) to events with independent seeds.

Claim 4.3. Recall $B_J := \bigcup_{i=1}^{k'} B_{j_i}$ contains buckets in $[\ell]$ with elements in Y (under g).

$\Pr_{h \sim \mathcal{H}}[\max h(Y) = \theta \wedge \min h(X \setminus Y) > \theta]$ in (4.1) could be decomposed as

$$\Pr_{\sigma: (C_e+1) \cdot k\text{-wise}}[\max \sigma(Y) = \theta \wedge \min \sigma(B_J) > \theta] \cdot \left(\prod_{i \notin J} \left(\Pr_{\sigma \sim U}[\min \sigma(B_i) > \theta] \pm 2 \cdot 2^{-C_s \cdot t} \right) \pm 2 \cdot 2^{-C_s \cdot t \cdot k} \right). \quad (4.2)$$

In (4.2), the product $\prod_{i \notin J} (\Pr_{\sigma \sim U}[\min \sigma(B_i) > \theta] \pm 2 \cdot 2^{-C_s \cdot t})$ replaces dependent seeds s_i for $i \in [\ell] \setminus J$ by independent seeds. The term $2 \cdot 2^{-C_s \cdot t}$ comes from the error of the extractor and the error of PRG_2 . Moreover, the last error term $2 \cdot 2^{-C_s \cdot t \cdot k}$ in (4.2) is similar to the error term (3.3) in [Claim 3.4](#), which is introduced by PRG_1 .

One more remark is that this shows the sum of t -wise independence and the Nisan-Zuckerman PRG could fool conditional events like (1.5). The key of (4.2) is to fool event $\Pr[\max \sigma(Y) = \theta]$ with a multiplicative error $2 \cdot 2^{-C_s \cdot t \cdot k}$.

We split (4.2) into two parts

$$\Pr_{\sigma: (C_e+1) \cdot k\text{-wise}}[\max \sigma(Y) = \theta \wedge \min \sigma(B_J) > \theta] \cdot \prod_{i \notin J} \left(\Pr_{\sigma \sim U}[\min \sigma(B_i) > \theta] \pm 2 \cdot 2^{-C_s \cdot t} \right) \quad (4.3)$$

$$\pm \Pr_{\sigma: (C_e+1) \cdot k\text{-wise}}[\max \sigma(Y) = \theta \wedge \min \sigma(B_J) > \theta] \cdot 2^{-C_s \cdot t \cdot k+1}. \quad (4.4)$$

Similar to the proof strategy of [Theorem 3.1](#), we bound (4.3) and (4.4) separately.

Claim 4.4. The summation of (4.4),

$$\sum_{\theta \in [M]} \Pr_{\sigma: (C_e+1) \cdot k\text{-wise}}[\max \sigma(Y) = \theta \wedge \min \sigma(B_J) > \theta] \cdot 2^{-C_s \cdot t \cdot k+1} \leq 2^{-2C \cdot t} / |X|^k.$$

Then we bound (4.3) by the following claim, which shows its summation is an approximation of k -min-wise hash with a small multiplicative error.

Claim 4.5. The summation of (4.3) over θ ,

$$\sum_{\theta \in [M]} \Pr_{\sigma: (C_e+1) \cdot k\text{-wise}}[\max \sigma(Y) = \theta \wedge \min \sigma(B_J) > \theta] \cdot \prod_{i \notin J} \left(\Pr_{\sigma \sim U}[\min \sigma(B_i) > \theta] \pm 2 \cdot 2^{-C_s \cdot t} \right),$$

equals $(1 \pm 2^{-2C \cdot t}) \cdot \Pr_{\sigma \sim U}[\max \sigma(Y) < \min \sigma(X \setminus Y)]$.

For completeness, we show the proof of [Claim 4.3](#) in [Section 4.1](#). Then we defer the proofs of [Claim 4.4](#) and [Claim 4.5](#) to [Section 4.2](#) and [Section 4.3](#). We are ready to finish the proof of [Theorem 4.1](#) here.

Proof of Theorem 4.1. We rewrite $\Pr_{h \sim \mathcal{H}}[\max h(Y) < \min h(X \setminus Y)]$ as (4.1). Then we apply [Claim 4.3](#) to each term $\Pr_{h \sim \mathcal{H}}[\max h(Y) = \theta \wedge \min h(X \setminus Y) > \theta]$.

Next we decompose the bound (4.2) in [Claim 4.3](#) into two terms: (4.3) and (4.4). Finally, [Claim 4.4](#) shows (4.4) is a small multiplicative error and [Claim 4.5](#) shows (4.3) is an approximation with multiplicative error $2^{-2C \cdot t}$. \square

4.1 Proof of Claim 4.3

We enumerate the non-empty subset $Y' \subseteq Y$ with $h(Y') = \theta$ to rewrite $\max h(y) = \theta$ as $(h(Y') = \theta \wedge \max h(Y \setminus Y') < \theta)$. So our goal is to bound

$$\begin{aligned} & \Pr_{h \sim \mathcal{H}} [\max h(Y) = \theta \wedge \min h(X \setminus Y) > \theta] \\ &= \sum_{\emptyset \neq Y' \subseteq Y} \Pr_{h \sim \mathcal{H}} [h(Y') = \theta \wedge \max h(Y \setminus Y') < \theta \wedge \min h(X \setminus Y) > \theta]. \end{aligned} \quad (4.5)$$

Let us consider each probability for a fixed Y' .

For convenience, we define $I_i(h_0, z_i)$ to denote the indicator of that hash $h(x) = h_0(x) + \sigma_i(x)$ for $\sigma_i = \text{PRG}_2(z_i)$ satisfies all conditions in (4.5) for $x \in X$ mapped to bucket i , i.e., $h(x) = \theta$ for $x \in Y'$ with $g(x) = i$, $h(x) < \theta$ for $x \in Y \setminus Y'$ with $g(x) = i$, and $h(x) > \theta$ for $x \in X \setminus Y$ with $g(x) = i$. Then we can rewrite $\Pr_{h \sim \mathcal{H}} [h(Y') = \theta \wedge \max h(Y \setminus Y') < \theta \wedge \min h(X \setminus Y) > \theta]$ as

$$\mathbb{E}_{h_0, w, (s_1, \dots, s_\ell) \sim \text{PRG}_1: z_i = \text{Ext}(w, s_i)} \left[\prod_{i \in [\ell]} I_i(h_0, z_i) \right]. \quad (4.6)$$

We use the analysis of the Nisan-Zuckerman PRG implicitly. In the first step, we apply h_0 to calculating $\mathbb{E} \left[\prod_{j \in J} I_j \right]$. In this calculation, we fix $z_J := (z_{j_1}, \dots, z_{j_{k'}})$ and their corresponding functions $\sigma_J := (\sigma_{j_1}, \dots, \sigma_{j_{k'}})$. Recall that h_0 is $(C_e + 1) \cdot k$ -wise independent and $|Y| \leq k$ are mapped to $J := \{j_1, \dots, j_{k'}\}$ under g . Therefore

$$\begin{aligned} & \mathbb{E}_{h_0, w, s \sim \text{PRG}_1} \left[\prod_{j \in J} I_j \right] = \mathbb{E}_{w, s \sim \text{PRG}_1} \left[\mathbb{E}_{h_0} \left[\prod_{j \in J} I_j \mid z_J, \sigma_J \right] \right] \\ &= \Pr_{\sigma: (C_e+1)k\text{-wise}} [\sigma(Y') = \theta \wedge \max \sigma(Y \setminus Y') < \theta \wedge \min \sigma(B_J) > \theta]. \end{aligned} \quad (4.7)$$

Now we fix h_0 and z_J such that $\prod_{j \in J} I_j = 1$ (otherwise it contributes 0 to (4.1)) and analyze the conditional expectation:

$$\mathbb{E}_{h_0, w, s \sim \text{PRG}_1} \left[\prod_{i \notin J} I_i \mid \prod_{j \in J} I_j = 1 \right]. \quad (4.8)$$

Similar to the proof in Theorem 3.1, we use the property of PRG_1 to replace s_1, \dots, s_ℓ by independent seeds. Let $\alpha_J = (\alpha_{j_1}, \dots, \alpha_{j_{k'}}) \in \{0, 1\}^{C_e t \times J}$ denote the enumeration of $s_J = (s_{j_1}, \dots, s_{j_{k'}})$. We expand the conditional expectation as

$$\begin{aligned} & \mathbb{E}_{h_0, w, s \sim \text{PRG}_1} \left[\prod_{i \notin J} I_i \mid \prod_{j \in J} I_j = 1 \right] = \sum_{\alpha_J} \mathbb{E}_{h_0, w, s \sim \text{PRG}_1} \left[\mathbf{1}(s_J = \alpha_J) \cdot \prod_{i \notin J} I_i \mid \prod_{j \in J} I_j = 1 \right] \\ &= \sum_{\alpha_J} \mathbb{E}_{h_0, w} \left[\Pr_{s \sim \text{PRG}_1} \left[s_J = \alpha_J \mid \prod_{j \in J} I_j = 1 \right] \cdot \mathbb{E}_{s \sim \text{PRG}_1} \left[\prod_{i \notin J} I_i \mid s_J = \alpha_J, \prod_{j \in J} I_j = 1 \right] \right]. \end{aligned}$$

For each fixed w and h_0 , $\prod_{i \notin J} I_i$ is a combinatorial rectangle, whose inputs $(s_i)_{i \notin J}$ are in $\{0, 1\}^{C_e t}$. Hence we apply PRG_1 to $\mathbb{E}_{s \sim \text{PRG}_1} [\prod_{i \notin J} I_i \mid s_J = \alpha_J]$. Observed that $\prod_{j \in J} I_j = 1$ is determined given w , h_0 , and $s_J = \alpha_J$, which could be neglected here.

Because $|s_J| = k' \cdot C_e \cdot t \leq C_e \cdot t \cdot k$, by the same argument of [Fact 3.6](#), it follows that

$$\mathbb{E}_{s \sim \text{PRG}_1} \left[\prod_{i \notin J} I_i \mid s_J = \alpha_J \right] = \prod_{i \notin J} \mathbb{E}_{s_i \sim U} [I_i] \pm 2^{-C_s \cdot t \cdot k}.$$

This simplifies [\(4.8\)](#) to

$$\sum_{\alpha_J} \Pr_{h_0, w, s \sim \text{PRG}_1} \left[s_J = \alpha_J \mid \prod_{j \in J} I_j = 1 \right] \cdot \left(\mathbb{E}_{h_0, w, (s_i)_{i \notin J} \sim U} \left[\prod_{i \notin J} I_i \mid \prod_{j \in J} I_j = 1 \right] \pm 2^{-C_s \cdot t \cdot k} \right). \quad (4.9)$$

We will bound $\mathbb{E}_{h_0, w, (s_i)_{i \notin J} \sim U} \left[\prod_{i \notin J} I_i \mid \prod_{j \in J} I_j = 1 \right]$ for any fixed $s_J = \alpha_J$ conditioned with $\prod_{j \in J} I_j = 1$. We apply the Nisan-Zuckerman analysis because seeds s_i are independent here. Since $|z_J| = k' \cdot C_e \cdot \log N \leq k \cdot C_e \cdot \log N \leq 0.1 \cdot |w|$ given $|w| = 10k \cdot C_e \cdot \log N$, the min-entropy of w is at least $0.8 \cdot |w|$ with probability $1 - 2^{-0.1 \cdot |w|}$ after fixing z_J . So we assume the min-entropy of w is at least $0.8|w|$ conditioned on that h_0 and z_J will satisfy $\prod_{j \in J} I_j = 1$.

Then for $i \notin J$, $z_i = \text{Ext}(w, s_i)$ with $s_i \sim U$ is **ExtErr**-close to the uniform distribution, which implies σ_i is **ExtErr**-close to the uniform distribution in PRG_2 . We repeat this argument for every $i \notin J$ and obtain

$$\mathbb{E}_{h_0, w, (s_i)_{i \notin J} \sim U} \left[\prod_{i \notin J} I_i \mid \prod_{j \in J} I_j = 1 \right] = \prod_{i \notin J} \left(\mathbb{E}_{\sigma_i \sim \text{PRG}_2} [I_i] \pm \text{ExtErr} \right) \pm (\ell - k') \cdot 2^{-0.2 \cdot |w|}. \quad (4.10)$$

Moreover, $\mathbb{E}_{\sigma_i \sim \text{PRG}_2} [I_i]$ is equal to $\Pr_{\sigma \sim U} [\min \sigma(B_i) > \theta] \pm 2 \cdot 2^{-C_s \cdot t}$ by the definition of $\sigma_i = \text{PRG}_2(z_i)$ for a PRG with error $2^{-C_s \cdot t}$.

The additive term $(\ell - k') \cdot 2^{-0.2 \cdot |w|}$ is the union bound for the min-entropy of w is less than $0.6 \cdot |w|$ after conditioned previous indicators are 1. Since $t := \frac{\log N}{\log \log N}$ and $|w| = 10k \cdot C_e \cdot \log N$, we combine it with the error in [\(4.9\)](#) as $2 \cdot 2^{-C_s \cdot t \cdot k}$. Thus [\(4.9\)](#) becomes

$$\begin{aligned} & \sum_{\alpha_J} \Pr_{h_0, w, s \sim \text{PRG}_1} \left[s_J = \alpha_J \mid \prod_{j \in J} I_j = 1 \right] \\ & \cdot \left(\prod_{i \notin J} \left(\Pr_{\sigma \sim U} [\min \sigma(B_i) > \theta] \pm \text{ExtErr} \pm 2^{-C_s \cdot t} \right) \pm 2^{-C_s \cdot t \cdot k} \pm 2^{-0.1 \cdot |w|} \pm (\ell - k') \cdot 2^{-0.2 \cdot |w|} \right) \\ & = \prod_{i \notin J} \left(\Pr_{\sigma \sim U} [\min \sigma(B_i) > \theta] \pm 2 \cdot 2^{-C_s \cdot t} \right) \pm 2 \cdot 2^{-C_s \cdot t \cdot k}. \end{aligned} \quad (4.11)$$

We complete this proof by plugging [\(4.7\)](#) for $\mathbb{E} \left[\prod_{j \in J} I_j \right]$ and [\(4.11\)](#) for $\mathbb{E} \left[\prod_{i \notin J} I_i \mid \prod_{j \in J} I_j \right]$.

$$\begin{aligned} & \mathbb{E} \left[\prod_{i \in [\ell]} I_i \right] = \mathbb{E} \left[\prod_{j \in J} I_j \right] \cdot \mathbb{E} \left[\prod_{i \notin J} I_i \mid \prod_{j \in J} I_j = 1 \right] \\ & = \Pr_{\sigma: (C_e+1)k\text{-wise}} [\sigma(Y') = \theta \wedge \max \sigma(Y \setminus Y') < \theta \wedge \min \sigma(B_J) > \theta] \\ & \cdot \left(\prod_{i \notin J} \left(\Pr_{\sigma \sim U} [\min \sigma(B_i) > \theta] \pm 2 \cdot 2^{-C_s \cdot t} \right) \pm 2 \cdot 2^{-C_s \cdot t \cdot k} \right). \end{aligned}$$

Moreover, by (4.5),

$$\begin{aligned} & \sum_{Y'} \Pr_{\sigma: (C_e+1)k\text{-wise}} [\sigma(Y') = \theta \wedge \max \sigma(Y \setminus Y') < \theta \wedge \min \sigma(B_J) > \theta] \\ &= \Pr_{\sigma: (C_e+1)k\text{-wise}} [\max \sigma(Y) = \theta \wedge \min \sigma(B_J) > \theta], \end{aligned}$$

which finishes this proof.

4.2 Proof of Claim 4.4

If $|X| < 2^{0.5C_s \cdot t+1}$, then the last factor $2^{-C_s \cdot t \cdot k+1}$ implies that this summation is at most $2^{-0.5C_s \cdot t \cdot k}/|X|^k$.

Otherwise $|X| \geq 2^{0.5C_s \cdot t+1}$ such that Lemma 4.2 implies that all B_i has $|B_i| = (1 \pm 0.1) \cdot |X \setminus Y|/\ell$. So $|B_J| = (1 \pm 0.1) \cdot k' \cdot |X \setminus Y|/\ell$.

If we neglect the $2^{-C_s \cdot t \cdot k+1}$ factor and consider $\sum_{\theta \in [M]} \Pr_{\sigma: (C_e+1) \cdot k\text{-wise}} [\max \sigma(Y) = \theta \wedge \min \sigma(B_J) > \theta]$, this is the exact probability of $B_J \cup Y$ satisfying the k -min-wise hash condition under $(C_e + 1) \cdot k$ -wise independence. Feigenblat, Porat and Shifan [FPS11] have shown this bounded by $2k!/|B_J|^k$ when C_e is large enough. We state their results for completeness.

Theorem 4.6 (Theorem 1.1 in [FPS11]). *There exists a constant c such that for any $\varepsilon > 0$, any $c \cdot (k \log \log(1/\varepsilon) + \log(1/\varepsilon))$ -wise independent function from $[N]$ to $[M]$ is a k -min-wise hash family of error ε when $M = \Omega(N/\varepsilon)$.*

So we simplify the summation as

$$\frac{O(1)}{\binom{|B_J \cup Y|}{|Y|}} \cdot 2^{-C_s \cdot t \cdot k+1} \leq \frac{2k!}{|B_J|^k} \cdot 2^{-C_s \cdot t \cdot k+1} \leq 2k! \cdot 2^{-C_s \cdot t \cdot k+1} \cdot \left(\frac{\ell}{0.9 \cdot k' \cdot |X \setminus Y|} \right)^k \leq 2^{-0.5C_s \cdot t \cdot k}/|X|^k,$$

given $\ell = 2^t$.

4.3 Proof of Claim 4.5

First of all, it would be more convenient to rewrite the summation as

$$\begin{aligned} & \sum_{\theta \in [M]} \Pr_{\sigma: (C_e+1)k\text{-wise}} [\max \sigma(Y) = \theta \wedge \min \sigma(B_J) > \theta] \cdot \prod_{i \notin J} \left(\Pr_{\sigma \sim U} [\min \sigma(B_i) > \theta] \pm 2 \cdot 2^{-C_s \cdot t} \right) \\ &= \sum_{\theta \in [M]} \Pr_{\sigma \sim U} [\max \sigma(Y) = \theta] \cdot \Pr_{\sigma: C_e \cdot k\text{-wise}} [\min \sigma(B_J) > \theta] \cdot \prod_{i \notin J} \left(\Pr_{\sigma \sim U} [\min \sigma(B_i) > \theta] \pm 2 \cdot 2^{-C_s \cdot t} \right) \\ &= \sum_{\theta \in [M]} \frac{\theta^k - (\theta - 1)^k}{M^k} \cdot \Pr_{\sigma: C_e \cdot k\text{-wise}} [\min \sigma(B_J) > \theta] \cdot \prod_{i \notin J} \left(\Pr_{\sigma \sim U} [\min \sigma(B_i) > \theta] \pm 2 \cdot 2^{-C_s \cdot t} \right). \end{aligned}$$

This proof considers the three cases in Lemma 4.2 according to the size of $|X|$. Recall that we have assumed $k = \log^{O(1)} N$. Set $\varepsilon := 2 \cdot 2^{-C_s \cdot t}$ for simplicity.

The first case of $|X| \leq \ell^{0.9}$. Lemma 4.2 implies $|B_i| \leq C_g + 10 \cdot \frac{k \log |X|}{\log N / \log \log N}$, $\forall i \in [\ell]$ and $|B_J| \leq C_g \cdot k$. Thus the middle term $\Pr_{\sigma: C_e k\text{-wise}}[\min \sigma(B_J) > \theta]$ has no error. Let us focus on the product

$$\prod_{i \notin J} \left(\Pr_{\sigma \sim U}[\min \sigma(B_i) > \theta] \pm 2 \cdot 2^{-C_s t} \right) = \prod_{i \notin J} \left((1 - \theta/M)^{|B_i|} \pm \varepsilon \right).$$

Without loss of generality, we assume $J = \{\ell - k' + 1, \dots, \ell\}$ and $|B_1| \geq |B_2| \geq \dots \geq |B_{\ell-k'}|$. When θ is small, say $(1 - \theta/M)^{|B_1|} \geq 2^{-0.5C_s t}$, this product has a small multiplicative error:

$$\begin{aligned} \prod_{i \notin J} \left((1 - \theta/M)^{|B_i|} \pm \varepsilon \right) &= \prod_{i \notin J} (1 - \theta/M)^{|B_i|} \cdot (1 \pm 2^{-0.5C_s t}) \\ &= (1 \pm \ell \cdot 2^{-0.5C_s t}) \cdot \prod_{i \notin J} (1 - \theta/M)^{|B_i|} = (1 \pm 2^{-(0.5C_s - 1) \cdot t}) \cdot \prod_{i \notin J} (1 - \theta/M)^{|B_i|}. \end{aligned}$$

Otherwise $(1 - \theta/M)^{|B_1|} < 2^{-0.5C_s t}$ is already small enough such that we only need to give a tail bound. Let $S := 0.5k \cdot \log \log N$ and $B_{\notin J} := \bigcup_{i=1}^{\ell-k'} B_i$ for convince.

1. The easy case is $|B_{\notin J}| \geq |B_1| \cdot k \log \log N$, which tells that $(1 - \theta/M)^{|B_{\notin J}|} \leq (2^{-0.5C_s t})^{k \log \log N} = N^{-0.5C_s \cdot k}$ is negligible. We show $\prod_{i \notin J} ((1 - \theta/M)^{|B_i|} \pm \varepsilon)$ is also negligible in this case.

- If $(1 - \theta/M)^{|B_S|} \leq 2^{-t-1}$, $\prod_{i=1}^S ((1 - \theta/M)^{|B_i|} \pm \varepsilon) \leq N^{-0.5k}$ is sufficiently small.
- If not, then $\prod_{i=S+1}^{\ell-k'} ((1 - \theta/M)^{|B_i|} \pm \varepsilon) \leq 2 \cdot \prod_{i=S+1}^{\ell-k'} (1 - \theta/M)^{|B_i|} = N^{-\Omega(k)}$ is sufficiently small.

2. When $|B_{\notin J}| \leq |B_1| \cdot k \log \log N = \log^{O(1)} N$, this further implies $|B_i| \leq C_g + 10 \cdot \frac{k(\log \log N)^2}{\log N}$.

- If $10 \cdot \frac{k(\log \log N)^2}{\log N} \geq C_g$, then $k \geq 0.1C_g \cdot \frac{\log N}{(\log \log N)^2}$ such that the number of non-empty buckets is at least $\frac{k}{20 \cdot \frac{k(\log \log N)^2}{\log N}} = \frac{\log N}{20(\log \log N)^2}$. So we could prove $\prod_{i=S+1}^{\ell-k'} ((1 - \theta/M)^{|B_i|} \pm \varepsilon)$ is negligible by considering B_S again like the above case.
- If $10 \cdot \frac{k(\log \log N)^2}{\log N} < C_g$, then each bucket has at most $2C_g$ elements. Also we have $k = O\left(\frac{\log N}{(\log \log N)^2}\right)$ and $|X| = O(k \log \log N) = O\left(\frac{\log N}{\log \log N}\right)$ such that $1/|X|^k = 2^{-O\left(\frac{\log N}{\log \log N}\right)}$.

From the condition $(1 - \theta/M)^{|B_1|} < 2^{-0.5C_s t}$, we have $\theta/M \geq 1 - 2^{-(C_s t)/(4C_g)}$. The number of such θ 's is at most $M \cdot 2^{-(C_s t)/(4C_g)}$. Using the fact $\frac{\theta^k - (\theta-1)^k}{M^k} \leq k/M$, this implies the additive error is $k \cdot 2^{-(C_s t)/(4C_g)} < 2^{-3C \cdot t}$.

The second case of $|X| \in (\ell^{0.9}, \ell^{1.1})$. The failure probability in Lemma 4.2 has a negligible impact on the final multiplicative error. Thus we assume that the max-load $\max_{i \in [\ell]} |B_i|$ is bounded by $2\ell^{0.1}$.

When $\theta/M \leq 0.5\ell^{-0.2}$, $(1 - \theta/M)^{|B_J|} \geq 1 - |B_J| \cdot \theta/M \geq 1 - k \cdot \ell^{-0.1} \geq 0.5$. By the first statement of Lemma 2.4, we have

$$\Pr_{\sigma: C_e k\text{-wise}}[\min \sigma(B_J) > \theta] = (1 - \theta/M)^{|B_J|} \pm (|B_J| \cdot \theta/M)^{C_e \cdot k} = (1 - \theta/M)^{|B_J|} \cdot \left(1 \pm 2 \left(\frac{k}{\ell^{0.1}} \right)^{C_e \cdot k} \right),$$

and for $i \notin J$:

$$(1 - \theta/M)^{|B_i|} \pm \varepsilon = (1 - \theta/M)^{|B_i|} \cdot (1 \pm 2\varepsilon).$$

Since $k = \log^{O(1)} N$ and $\ell = 2^t$, which tells that $k \ll \ell$, the multiplicative error B_J is bounded by $2 \left(\frac{k}{\ell^{0.1}}\right)^{C_e \cdot k} \leq 2\ell^{-0.05C_e \cdot k}$. So

$$\begin{aligned} & \Pr_{\sigma: C_e k\text{-wise}} [\min \sigma(B_J) > \theta] \cdot \prod_{i \notin J} \left((1 - \theta/M)^{|B_i|} \pm \varepsilon \right) \\ &= (1 - \theta/M)^{|X \setminus Y|} \cdot \left(1 \pm \left(\frac{4}{\ell^{0.05C_e \cdot k}} + 4(\ell - k') \cdot \varepsilon \right) \right) \\ &= (1 - \theta/M)^{|X \setminus Y|} \cdot (1 \pm 2^{-0.5C_s \cdot t}). \end{aligned}$$

When $\theta/M > 0.5\ell^{-0.2}$, we only need to estimate $\sum_{\theta > 0.5M \cdot \ell^{-0.2}} \frac{\theta^k - (\theta - 1)^k}{M^k} \cdot \prod_{i \notin J} ((1 - \theta/M)^{|B_i|} + \varepsilon)$. Consider the $S := k \cdot C' \cdot \log \log N$ largest number among $\{|B_i|\}_{i \notin J}$, denoted by b .

1. If $(1 - \theta/M)^b \geq \varepsilon \cdot \ell$, then

$$\begin{aligned} & \sum_{\theta > 0.5M \cdot \ell^{-0.2} \wedge (1 - \theta/M)^b \geq \varepsilon \cdot \ell} \frac{\theta^k - (\theta - 1)^k}{M^k} \cdot \prod_{i \notin J} ((1 - \theta/M)^{|B_i|} + \varepsilon) \\ & \leq \sum_{\theta > 0.5M \cdot \ell^{-0.2} \wedge (1 - \theta/M)^b \geq \varepsilon \cdot \ell} \frac{\theta^k - (\theta - 1)^k}{M^k} \cdot \prod_{i \notin J: |B_i| > b} ((1 - \theta/M)^{|B_i|} + \varepsilon) \\ & \leq \sum_{\theta > 0.5M \cdot \ell^{-0.2} \wedge (1 - \theta/M)^b \geq \varepsilon \cdot \ell} \frac{\theta^k - (\theta - 1)^k}{M^k} \cdot e \cdot (1 - \theta/M)^{|X \setminus Y| - (S + k') \cdot 2\ell^{0.1}} \\ & \leq \exp(1 - 0.9|X| \cdot 0.5\ell^{-0.2}) \leq \exp(-\ell^{0.6}). \end{aligned}$$

2. If $(1 - \theta/M)^b < \varepsilon \cdot \ell$, then

$$\begin{aligned} & \sum_{\theta > 0.5M \cdot \ell^{-0.2} \wedge (1 - \theta/M)^b < \varepsilon \cdot \ell} \frac{\theta^k - (\theta - 1)^k}{M^k} \cdot \prod_{i \notin J} ((1 - \theta/M)^{|B_i|} + \varepsilon) \\ & \leq \sum_{\theta > 0.5M \cdot \ell^{-0.2} \wedge (1 - \theta/M)^b < \varepsilon \cdot \ell} \frac{\theta^k - (\theta - 1)^k}{M^k} \cdot (\varepsilon \cdot \ell + \varepsilon)^S \\ & \leq 2^{2S - C' \cdot (C_s - 1) \cdot t \cdot S} = 1/N^{O(k)}. \end{aligned}$$

The third case of $|X| \geq \ell^{1.1}$. Again, the proof of this case is the same as the second case, and is thus omitted.

5 Extractors

We restate [Lemma 2.6](#) here and finish its proof in this section. Different from previous works, this randomness extractor has an extra property: $\text{Ext}(U_n, s) = U_m$ for any fixed seed s .

Lemma 5.1. *Given any n and $k < n$, for any error ε , there exists a randomness extractor $\text{Ext} : \{0, 1\}^n \times \{0, 1\}^d \rightarrow \{0, 1\}^m$ with $m = k/2$ and $d = O(\log(n/\varepsilon))$. Moreover, Ext satisfies an extra property: $\text{Ext}(U_n, s) = U_m$ for any fixed seed s .*

We show how to design explicit extractors for [Lemma 5.1](#) in the rest of this section. Our construction will be based on the linear form of the lossless condenser in [\[GUV09, CI17\]](#).

Definition 5.2. $h : \mathbf{F}_2^n \times \mathbf{F}_2^d \rightarrow \mathbf{F}_2^m$ is a (k, ε) -lossless condenser if for any random source X of min-entropy at least k , the distribution $(Z, h(X, Z))$ is ε -close to some distribution with min-entropy at least $k + d$, where Z is an independent seed uniformly distributed in \mathbf{F}_2^d .

We strengthen the linear seeded condensers in [\[GUV09, CI17\]](#) such that they are surjective for every seed. We state the main properties of lossless condensers as follows, which reformulates Corollary 38 of [\[CI17\]](#) with an extra surjective guarantee.

Lemma 5.3. Let $\alpha > 0$ be any constant. For any n , any $k < n$, and $\varepsilon > 0$, there exists an explicit (k, ε) -lossless condenser $h : \mathbf{F}_2^n \times \mathbf{F}_2^d \rightarrow \mathbf{F}_2^m$ with $d \leq (1 + 1/\alpha) \cdot \log(nk/\varepsilon) + O(1)$ and $m \leq d + (1 + \alpha)k$. Moreover, this lossless condenser satisfies the following two properties:

1. For every seed $s \in \mathbf{F}_2^d$, $h(x, s)$ is a linear function on x .
2. For every seed $s \in \mathbf{F}_2^d$, $h(x, s)$ is surjective such that $h(U_n, s) = U_m$.

Proof. Corollary 38 in [\[CI17\]](#) provides a linear lossless condenser with the above parameters although it is not necessarily surjective. In particular, its construction provides a condenser for every n and every $k < n$ as long as there exists an irreducible univariate polynomial of degree n in some finite extension field of \mathbf{F}_2 . Irreducible polynomials of degree n always exist because the number of irreducible polynomials of degree n in the finite field \mathbf{F}_q is $\frac{1}{n} \sum_{d|n} \mu(n/d) q^d$ greater than 0 (by the Gauss formula). Moreover, there are efficient algorithms [\[Sho90\]](#) to find such an irreducible polynomial.

Now we modify its construction to satisfy the second property.

Specifically, for every seed s , if $h(x, s)$ is not surjective, we expand it into a surject function on \mathbf{F}_2^m . Namely, if $h(x, s) = M_s \cdot x$ for a matrix $M_s \in \mathbf{F}_2^{m \times n}$ whose rank is not m , then we replace every linearly dependent row in M_s by an independent vector. Let M'_s be the new matrix and $h'(x, s) = M'_s \cdot x$ be the new condenser after guaranteeing the new matrix is of rank m .

Hence $h'(\cdot, s)$ is surjective and $h'(\cdot, s)$ is still linear. Moreover, the min-entropy of $h'(X, s)$ is not less than the min-entropy of $h(X, s)$ for any fixed s . Thus $H_\infty(Z, h'(X, Z)) \geq H_\infty(Z, h(X, Z))$. \square

Similar to [Lemma 5.3](#), we modify the classical leftover hash lemma to construct a linear extractor E of optimal error such that $E(U_n, s) = U_m$.

Claim 5.4. For any n , any $k < n$, and $m < k$, there exists a $(k, 2 \cdot 2^{\frac{m-k}{2}})$ -strong extractor $E : \{0, 1\}^n \times \{0, 1\}^{n-1} \rightarrow \{0, 1\}^m$ such that

1. $E(x, s)$ is a linear function of x for any seed s .
2. $E(U_n, s)$ is surjective such that $E(U_n, s) = U_m$ for any seed s .

Proof. We consider the extension field \mathbf{F}_{2^n} of size 2^n and view each element $\alpha \in \mathbf{F}_{2^n}$ as a vector in $\{0, 1\}^n$. For every seed $s \in \{0, 1\}^{n-1}$, we pick a distinct non-zero element $y_s \in \mathbf{F}_{2^n}$ and define $E(x, s) := (x \cdot y_s)_m$ to output the first m bits of the product $x \cdot y_s \in \mathbf{F}_{2^n}$. This guarantees that $E(U_n, s) = U_m$ for any seed s (since $y_s \neq 0$).

Then we bound its error by $2 \cdot 2^{\frac{m-k}{2}}$. The standard leftover hash lemma shows that $(y, (xy)_m) \approx_{2^{\frac{m-k}{2}}} (U_n, U_m)$ when $x \sim X$ of min-entropy k and $y \sim \mathbf{F}_{2^n}$. However, the support size of y is 2^{n-1} instead of 2^n in our construction. But this will only increase the error by a factor of 2 (via the Markov inequality). \square

The two extra properties in [Claim 5.4](#) are identical to the properties in [Lemma 5.3](#). These properties help us to design an extractor such that $E(U_n, s) = U_m$. The second property guarantees that the output is uniformly distributed over \mathbf{F}_2^m for any seed s whenever we apply [Lemma 5.3](#) or [Claim 5.4](#) to a uniform source. The first property further shows that $U_n \mid E(U_n, s)$ is still a uniform random source in the dual space of $h(\cdot, s)$ of dimension $n - m$. This allows our construction to continue this type of randomness extraction and condensation.

While our construction of [Lemma 5.1](#) follows the same outline of Theorem 5.12 in [\[GUV09\]](#), one *subtle difference* is that after every application of [Lemma 5.3](#) or [Claim 5.4](#), we replace the random source X by its projection onto the dual space of the linear map.

Proof of Lemma 5.1. The difference between our construction and Theorem 5.12 in [\[GUV09\]](#) are (1) we replace every operation of leftover hash lemma by [Claim 5.4](#) and replace every operation of lossless condensers by [Lemma 5.3](#); (2) after each operation, we project the random source to the dual space of that operation. Specifically, let $X_0 \in \mathbf{F}_2^n$ be the initial random source and $X_i \in \mathbf{F}_2^{n_i}$ be the random source after applying i times [Claim 5.4](#) and [Lemma 5.3](#). For example, suppose the $(i + 1)$ -th operation applies the lossless condenser $h_i : \mathbf{F}_2^{n_i} \times \mathbf{F}_2^{d_i} \rightarrow \mathbf{F}_2^{m_i}$ with seed s_i in [Lemma 5.3](#), say $h_i(X_i, s_i) := A \cdot X_i$ for some full rank matrix $A \in \mathbf{F}_2^{m_i \times n_i}$. Then we set the next random source to be $X_{i+1} := A^\perp \cdot X_i$ where $A^\perp \in \mathbf{F}_2^{(n_i - m_i) \times n_i}$ is the dual of A . This works because [Lemma 5.3](#) and [Claim 5.4](#) hold for any n and any $k < n$.

To prove $E(U_n, s) = U_m$, we use the two extra properties in [Lemma 5.3](#) and [Claim 5.4](#). Our modification guarantees that every X_i is uniform (by inductions).

The analysis of the error follows the same argument of [\[GUV09\]](#). A small difference is to bound the min-entropy X_{i+1} given $h_i(X_i, s_i)$. Since $h_i(\cdot, s_i)$ is linear, X_{i+1} is the exact distribution of X_i conditioned on $h_i(X_i, s_i)$ such that $H_\infty(X_{i+1}) = H_\infty(X_i \mid h_i(X_i, s_i))$. \square

Acknowledgements

We appreciate anonymous reviewers for their helpful comments. We are also grateful to Parikshit Gopalan for pointing out some typos in an earlier version of this paper.

References

- [AGM12] Kook Jin Ahn, Sudipto Guha, and Andrew McGregor. Graph sketches: sparsification, spanners, and subgraphs. In *Proceedings of the 31st ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, PODS '12, page 5–14, New York, NY, USA, 2012. Association for Computing Machinery. [2](#)
- [ASWZ96] R. Armoni, M. Saks, A. Wigderson, and Shiyu Zhou. Discrepancy sets and pseudorandom generators for combinatorial rectangles. In *Proceedings of the 37th Annual Symposium on Foundations of Computer Science*, FOCS '96, page 412, USA, 1996. IEEE Computer Society. [2](#), [4](#), [6](#), [8](#)
- [BCFM00] Andrei Z Broder, Moses Charikar, Alan M Frieze, and Michael Mitzenmacher. Min-wise independent permutations. *Journal of Computer and System Sciences*, 60(3):630–659, 2000. [1](#), [5](#)
- [BRRY10] Mark Braverman, Anup Rao, Ran Raz, and Amir Yehudayoff. Pseudorandom generators for regular branching programs. In *Proceedings of the IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 40–47, 2010. [3](#), [6](#)

- [BV10] Joshua Brody and Elad Verbin. The coin problem and pseudorandomness for branching programs. In *Proceedings of the 51st IEEE symposium on Foundations of Computer Science*, pages 30–39, 2010. 3, 6
- [CDF⁺01] Edith Cohen, Mayur Datar, Shinji Fujiwara, Aristides Gionis, Piotr Indyk, Rajeev Motwani, Jeffrey D. Ullman, and Cheng Yang. Finding interesting associations without support pruning. *IEEE Trans. on Knowl. and Data Eng.*, 13(1):64–78, January 2001. 1, 2, 3, 6
- [CF14] Graham Cormode and Donatella Firmani. A unifying framework for ℓ_0 -sampling algorithms. *Distrib Parallel Databases*, 32:315–335, January 2014. 1
- [CI17] Mahdi Cheraghchi and Piotr Indyk. Nearly optimal deterministic algorithm for sparse walsh-hadamard transform. *ACM Trans. Algorithms*, 13(3), March 2017. 26
- [Coh16] Edith Cohen. *Min-Hash Sketches*, pages 1282–1287. Springer New York, New York, NY, 2016. 1
- [De11] Anindya De. Pseudorandomness for permutation and regular branching programs. In *Proceedings of the IEEE 26th Annual Conference on Computational Complexity*, pages 221–231, 2011. 6
- [DM02] Mayur Datar and S. Muthukrishnan. Estimating rarity and similarity over data stream windows. In *Algorithms — ESA 2002*, pages 323–335, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg. 1, 2, 3, 6
- [FK18] Michael A. Forbes and Zander Kelley. Pseudorandom generators for read-once branching programs, in any order. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018*, pages 946–955. IEEE Computer Society, 2018. 3, 6
- [FPS11] Guy Feigenblat, Ely Porat, and Ariel Shiftan. Exponential time improvement for min-wise based algorithms. *Inf. Comput.*, 209(4):737–747, April 2011. 1, 2, 3, 6, 7, 23
- [GKM15] Parikshit Gopalan, Daniek Kane, and Raghu Meka. Pseudorandomness via the discrete fourier transform. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015*, pages 903–922, 2015. 6
- [GMR⁺12] Parikshit Gopalan, Raghu Meka, Omer Reingold, Luca Trevisan, and Salil P. Vadhan. Better pseudorandom generators from milder pseudorandom restrictions. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012*, pages 120–129, 2012. 2, 4, 6, 8
- [GUV09] Venkatesan Guruswami, Christopher Umans, and Salil Vadhan. Unbalanced expanders and randomness extractors from parvaresh–vardy codes. *J. ACM*, 56(4), July 2009. 4, 26, 27
- [GY20] Parikshit Gopalan and Amir Yehudayoff. Concentration for limited independence via inequalities for the elementary symmetric polynomials. *Theory of Computing*, 16(17):1–29, 2020. 1, 2, 3, 4, 6, 8, 30, 31
- [Hen06] Monika Henzinger. Finding near-duplicate web pages: a large-scale evaluation of algorithms. In *Proceedings of the 29th Annual International ACM SIGIR Conference on*

- Research and Development in Information Retrieval*, SIGIR '06, page 284–291, New York, NY, USA, 2006. Association for Computing Machinery. [1](#)
- [HGKI02] Taher H. Haveliwala, Aristides Gionis, Dan Klein, and Piotr Indyk. Evaluating strategies for similarity search on the web. In *Proceedings of the 11th International Conference on World Wide Web*, WWW '02, page 432–442, New York, NY, USA, 2002. Association for Computing Machinery. [1](#)
- [IMZ19] Russell Impagliazzo, Raghu Meka, and David Zuckerman. Pseudorandomness from shrinkage. *J. ACM*, 66(2), April 2019. [5](#), [6](#)
- [Ind01] Piotr Indyk. A small approximately min-wise independent family of hash functions. *J. Algorithms*, 38(1):84–90, January 2001. [1](#), [2](#), [3](#), [7](#), [10](#), [11](#)
- [INW94] Russell Impagliazzo, Noam Nisan, and Avi Wigderson. Pseudorandomness for network algorithms. In *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing*, pages 356–364, 1994. [3](#), [4](#), [6](#)
- [KNP11] Michal Koucky, Prajakta Nimbhorkar, and Pavel Pudlak. Pseudorandomness for group products. In *Proceedings of the 2011 IEEE 43rd Annual Symposium on Theory of Computing*, pages 263–272, 2011. [3](#), [6](#)
- [KNP⁺17] Michael Kapralov, Jelani Nelson, Jakub Pachocki, Zhengyu Wang, David P. Woodruff, and Mobin Yahyazadeh. Optimal lower bounds for universal relation, and for samplers and finding duplicates in streams. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017*, pages 475–486. IEEE Computer Society, 2017. [2](#)
- [LLSZ97] Nathan Linial, Michael Luby, Michael Saks, and David Zuckerman. Efficient construction of a small hitting set for combinatorial rectangles in high dimension. *Combinatorica*, 17(3):215–234, December 1997. [2](#), [8](#)
- [Lu02] Chi-Jen Lu. Improved pseudorandom generators for combinatorial rectangles. *Combinatorica*, 22(3):417–433, 2002. [2](#), [4](#), [6](#), [8](#), [9](#)
- [McG14] Andrew McGregor. Graph stream algorithms: a survey. *SIGMOD Rec.*, 43(1):9–20, May 2014. [1](#)
- [MRT19] Raghu Meka, Omer Reingold, and Avishay Tal. Pseudorandom generators for width-3 branching programs. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019*, pages 626–637. ACM, 2019. [3](#), [6](#)
- [Nis92] Noam Nisan. Pseudorandom generators for space-bounded computation. *Combinatorica*, 12(4):449–461, 1992. [3](#), [4](#), [6](#)
- [NZ96] Noam Nisan and David Zuckerman. Randomness is linear in space. *J. Comput. Syst. Sci.*, 52(1):43–52, 1996. [2](#), [3](#), [4](#), [6](#)
- [PT16] Mihai Pătraşcu and Mikkel Thorup. On the k -independence required by linear probing and minwise independence. *ACM Trans. Algorithms*, 12(1):8:1–8:27, 2016. [1](#), [5](#)
- [Sho90] Victor Shoup. New algorithms for finding irreducible polynomials over finite fields. *Mathematics of Computation*, 54(189):435–447, 1990. [26](#)

- [SSZZ00] Michael Saks, Aravind Srinivasan, Shiyu Zhou, and David Zuckerman. Low discrepancy sets yield approximate min-wise independent permutation families. *Inf. Process. Lett.*, 73(1–2):29–32, January 2000. [1](#), [2](#), [3](#), [5](#), [6](#), [8](#), [30](#)
- [Tre01] Luca Trevisan. Extractors and pseudorandom generators. *J. ACM*, 48(4):860–879, July 2001. [4](#)

A Connection between Min-wise Hash and PRG for Combinatorial Rectangles

Here we describe the connection between min-wish hash families and pseudo-random generators for combinatorial rectangles [SSZZ00, GY20].

As mentioned earlier $M = \Omega(N)$, we assume $\Pr_{h \sim U}[h(y) < \min h(X \setminus y)] \geq 1/(2|X|)$ and $\Pr_{h \sim U}[\max h(Y) < \min h(X \setminus Y)] \geq 1/(2\binom{|X|}{k})$ for $Y \subseteq X$ of size k in this work.

Lemma A.1. *Let $G : \{0, 1\}^s \rightarrow [M]^N$ be a PRG that fools combinatorial rectangles within additive error δ . Then G provides a min-wise hash family of size 2^s and error $2NM \cdot \delta$ and a k -min-wise hash family of size 2^s and error $\frac{N^k}{k!} \cdot 4M\delta$.*

Proof. Let x_1, \dots, x_n be the elements in $X \setminus y$. We rewrite $\Pr_{h \sim \mathcal{H}}[h(y) < \min h(X \setminus y)]$ as

$$\sum_{\theta \in [M]} \Pr_{h \sim \mathcal{H}} [h(y) = \theta \wedge h(x_1) > \theta \wedge \dots \wedge h(x_n) > \theta].$$

Since $\mathbf{1}(h(y) = \theta \wedge h(x_1) > \theta \wedge \dots \wedge h(x_n) > \theta)$ is a combinatorial rectangle in $[M]^N$, the additive error of each term is at most δ . Hence the total additive error is $M \cdot \delta$. Then, by the assumption $\Pr_{h \sim U}[h(y) < \min h(X \setminus y)] \geq 1/(2N)$, we have

$$\begin{aligned} & \sum_{\theta \in [M]} \Pr_{h \sim \mathcal{H}} [h(y) = \theta \wedge h(x_1) > \theta \wedge \dots \wedge h(x_n) > \theta] \\ &= \sum_{\theta \in [M]} \Pr_{h \sim U} [h(y) = \theta \wedge h(x_1) > \theta \wedge \dots \wedge h(x_n) > \theta] \pm M \cdot \delta \\ &= \Pr_{h \sim U} [h(y) < \min h(X \setminus y)] \cdot (1 \pm 2NM\delta). \end{aligned}$$

Similarly, for the k -min-wise hash, we express $\Pr[\max h(Y) = \theta \wedge \min h(X \setminus Y) > \theta]$ as

$$\Pr[\max h(Y) \leq \theta \wedge \min h(X \setminus Y) > \theta] - \Pr[\max h(Y) \leq \theta - 1 \wedge \min h(X \setminus Y) > \theta].$$

Then we rewrite $\Pr_{h \sim \mathcal{H}}[\max h(Y) < \min h(X \setminus Y)]$ as

$$\begin{aligned} & \sum_{\theta \in [M]} \Pr_{h \sim \mathcal{H}} [\max h(Y) = \theta \wedge \min h(X \setminus Y) > \theta] \\ &= \sum_{\theta \in [M]} \left(\Pr_{h \sim \mathcal{H}} [\max h(Y) \leq \theta \wedge \min h(X \setminus Y) > \theta] - \Pr_{h \sim \mathcal{H}} [\max h(Y) \leq \theta - 1 \wedge \min h(X \setminus Y) > \theta] \right) \\ &= \sum_{\theta \in [M]} \left(\Pr_{h \sim U} [\max h(Y) \leq \theta \wedge \min h(X \setminus Y) > \theta] - \Pr_{h \sim U} [\max h(Y) \leq \theta - 1 \wedge \min h(X \setminus Y) > \theta] \right) \pm 2M\delta \\ &= \Pr_{h \sim U} [\max h(Y) < \min h(X \setminus Y)] \pm 2M\delta. \end{aligned}$$

Since we assume $\Pr_{h \sim U}[\max h(Y) < \min h(X \setminus Y)] \geq k!/(2N^k)$, this k -min-wise hash family has an error $\frac{N^k}{k!} \cdot 4M\delta$. \square

Plugging [Theorem 2.7](#) to [Lemma A.1](#), Gopalan and Yehudayoff [\[GY20\]](#) had the following results of min-wise and k -min-wise hash with small errors.

Theorem A.2. *Given any N and multiplicative error δ , there is an explicit min-wise hash family of seed length $O(\log(NM/\delta) \cdot \log \log(NM/\delta))$.*

More generally, given any k , there is an explicit k -min-wise hash family of seed length $O((k \log N + \log(1/\delta)) \cdot \log(k \log N + \log(1/\delta)))$.

B Omitted Calculation in [Section 3.3.2](#)

Here we complete the calculation omitted for the third case of $|X| > \ell^{1.1}$ in [Section 3.3.2](#). Recall that $\varepsilon = 2 \cdot 2^{-C_s \cdot t}$, and \mathcal{H}' is the hash function family after replacing s_1, \dots, s_ℓ by independent random samples in $\{0, 1\}^{C_e \cdot t}$.

We want to prove that for any $X \subseteq [N]$ with size larger than $\ell^{1.1}$ and any $y \in X$, it holds that $\Pr_{h' \sim \mathcal{H}'}[h'(y) < \min h'(X \setminus y)] = (1 \pm 2^{-2C \cdot t}) \cdot \Pr_{\sigma \sim U}[\sigma(y) < \min \sigma(X \setminus y)]$.

Again, by [Lemma 3.2](#), the failure probability only affects the multiplicative error by at most $|X|/|X|^{3C} \leq 1/\ell^{1.1 \cdot (3C-1)} < 2^{-2.5C \cdot t}$.

Hence, we can suppose that in this case all buckets satisfy $|B_i| = (1 \pm 0.1) \cdot |X|/\ell$.

When $\frac{|X|}{\ell} \cdot \frac{\theta}{M} \leq \ell^{-0.1}$, there comes $(1 - \theta/M)^{|B_i|} \geq 1 - |B_i| \cdot \theta/M \geq 1 - 1.1/\ell^{0.1} \geq 0.5$, $\forall i \in [\ell]$. We use the first statement in [Lemma 2.4](#) to estimate $\Pr_{\sigma: 0.1C_s\text{-wise}}[\min \sigma(B_j) > \theta]$:

$$\begin{aligned} \Pr_{\sigma: 0.1C_s\text{-wise}}[\min \sigma(B_j) > \theta] &= (1 - \theta/M)^{|B_j|} \pm \left(|B_j| \cdot \frac{\theta}{M} \right)^{0.1C_s} \\ &= (1 - \theta/M)^{|B_j|} \left(1 \pm 2 \left(\frac{1.1}{\ell^{0.1}} \right)^{0.1C_s} \right) \\ &= (1 - \theta/M)^{|B_j|} (1 \pm O(1/\ell^{0.01C_s})). \end{aligned}$$

For the remaining buckets, we have

$$(1 - \theta/M)^{|B_i|} \pm \varepsilon = (1 - \theta/M)^{|B_i|} (1 \pm 2\varepsilon), \quad \forall i \neq j.$$

Thus, for relatively small θ , we obtain that

$$\begin{aligned} &\frac{1}{M} \sum_{\frac{|X|}{\ell} \cdot \frac{\theta}{M} \leq \ell^{-0.1}} \Pr_{\sigma: 0.1C_s\text{-wise}}[\min \sigma(B_j) > \theta] \cdot \prod_{i \neq j} ((1 - \theta/M)^{|B_i|} \pm \varepsilon) \\ &= \frac{1}{M} \sum_{\frac{|X|}{\ell} \cdot \frac{\theta}{M} \leq \ell^{-0.1}} (1 - \theta/M)^{|X|-1} \cdot (1 \pm (O(1/\ell^{0.01C_s}) + 4(\ell-1)\varepsilon)) \\ &= \frac{1}{M} \sum_{\frac{|X|}{\ell} \cdot \frac{\theta}{M} \leq \ell^{-0.1}} (1 - \theta/M)^{|X|-1} \cdot (1 \pm 2^{-0.001C_s \cdot t}). \end{aligned}$$

When $\frac{|X|}{\ell} \cdot \frac{\theta}{M} > \ell^{-0.1}$, we need to bound $\frac{1}{M} \sum_{\frac{|X|}{\ell} \cdot \frac{\theta}{M} > \ell^{-0.1}} \prod_{i \neq j} ((1 - \theta/M)^{|B_i|} \pm \varepsilon)$. Suppose that $j = \ell$, and $1.1 \cdot |X|/\ell \geq |B_1| \geq |B_2| \geq \dots \geq |B_{\ell-1}|$, and set $S = C' \cdot \log \log N \leq \ell - 1$. We split the sum into two cases depending on whether $(1 - \theta/M)^{|B_S|} > \varepsilon \cdot \ell$.

1. For $(1 - \theta/M)^{|B_S|} > \varepsilon \cdot \ell$, since $|X| \cdot \theta/M = \ell \cdot \frac{|X|}{\ell} \cdot \frac{\theta}{M} \geq \ell^{0.9}$, we further simplify it as

$$\begin{aligned}
& \frac{1}{M} \sum_{\substack{\frac{|X|}{\ell} \cdot \frac{\theta}{M} > \ell^{-0.1} \wedge (1-\theta/M)^{|B_S|} > \varepsilon \cdot \ell}} \prod_{i \neq j} ((1 - \theta/M)^{|B_i|} + \varepsilon) \\
& \leq \frac{1}{M} \sum_{\substack{\frac{|X|}{\ell} \cdot \frac{\theta}{M} > \ell^{-0.1} \wedge (1-\theta/M)^{|B_S|} > \varepsilon \cdot \ell}} \prod_{i=S+1}^{\ell-1} (1 - \theta/M)^{|B_i|} (1 + 1/\ell) \\
& \leq \frac{1}{M} \sum_{\substack{\frac{|X|}{\ell} \cdot \frac{\theta}{M} > \ell^{-0.1} \wedge (1-\theta/M)^{|B_S|} > \varepsilon \cdot \ell}} e \cdot (1 - \theta/M)^{|X| - 1.1(S+1) \cdot |X|/\ell} \\
& \leq \exp(1 - (|X| - 1.1(S+1) \cdot |X|/\ell) \cdot \theta/M) \leq \exp(-\ell^{0.8}).
\end{aligned}$$

2. For $(1 - \theta/M)^{|B_S|} \leq \varepsilon \cdot \ell$, we bound it as

$$\begin{aligned}
& \frac{1}{M} \sum_{\substack{\frac{|X|}{\ell} \cdot \frac{\theta}{M} > \ell^{-0.1} \wedge (1-\theta/M)^{|B_S|} \leq \varepsilon \cdot \ell}} \prod_{i \neq j} ((1 - \theta/M)^{|B_i|} + \varepsilon) \\
& \leq \frac{1}{M} \sum_{\substack{\frac{|X|}{\ell} \cdot \frac{\theta}{M} > \ell^{-0.1} \wedge (1-\theta/M)^{|B_S|} \leq \varepsilon \cdot \ell}} \prod_{i=1}^S (\varepsilon \cdot \ell + \varepsilon) \\
& \leq \frac{1}{M} \sum_{\substack{\frac{|X|}{\ell} \cdot \frac{\theta}{M} > \ell^{-0.1} \wedge (1-\theta/M)^{|B_S|} \leq \varepsilon \cdot \ell}} \prod_{i=1}^S 2^{2-(C_s-1) \cdot t} \\
& \leq 2^{2C' \cdot \log \log N - C' \cdot (C_s-1) \cdot \log N} = N^{-O(1)}.
\end{aligned}$$

Thus, we bound the total multiplicative error as

$$\frac{1}{|X|^{3C-1}} + 2^{-0.001C_s \cdot t} + |X| \cdot \left(\exp(-\ell^{0.8}) + N^{-O(1)} \right) < 2^{-2C \cdot t}.$$

C Proof of Lemma 4.2

Here we complete the proof of Lemma 4.2. We reproduce the lemma below for easy reference.

Lemma C.1. *For the allocation function g , let $B_i := \{x \in X \setminus Y : g(x) = i\}$. Particularly, define $B_J := \bigcup_{i=1}^{k'} B_{j_i}$. Then g guarantees that:*

1. When $|X| \leq \ell^{0.9}$, with probability $1 - \frac{1}{\ell^{3C} \cdot |X|^k}$, $|B_i| \leq C_g + 10 \cdot \frac{k \log |X|}{\log N / \log \log N}$ for all $i \in [\ell]$ and $|B_J| \leq C_g \cdot k$.
2. When $|X| \in (\ell^{0.9}, \ell^{1.1})$, with probability $1 - 1/\ell^{3C \cdot k}$, the max-load $\max_{i \in [\ell]} |B_i| \leq 2\ell^{0.1}$.
3. When $|X| \geq \ell^{1.1}$, with probability $1 - |X|^{-3C \cdot k}$, all buckets satisfy $|B_i| = (1 \pm 0.1) \cdot |X|/\ell$.

Proof. For convenience, set $r := |X \setminus Y|$ in this proof. We prove these three cases separately.

When $|X| \leq \ell^{0.9}$, let $v := C_g + 10 \cdot \frac{k \log |X|}{\log N / \log \log N}$. Note that $v < C_g \cdot k$, and there comes

$$\Pr_{g: C_g k\text{-wise}} [|B_i| \geq v] \leq \binom{r}{v} \cdot (1/\ell)^v \leq (r/\ell)^v \leq 1/\ell^{0.1v} \leq \frac{1}{\ell^{4C} \cdot |X|^k}, \quad \forall i \in [\ell].$$

Moreover, for B_J , note that $k \ll \ell$. Thus it follows that

$$\begin{aligned} \Pr_{g:C_g k\text{-wise}}[|B_J| \geq C_g \cdot k] &\leq \binom{r}{C_g \cdot k} \cdot \left(\frac{k'}{\ell}\right)^{C_g \cdot k} \leq \left(\frac{r \cdot k}{\ell}\right)^{C_g \cdot k} \\ &\leq \left(\frac{k}{\ell}\right)^{0.1 C_g \cdot k} \leq \left(\frac{1}{\ell}\right)^{0.5 C_g \cdot k} \leq \ell^{-5C \cdot k} \leq \frac{1}{\ell^{4C} \cdot |X|^k}. \end{aligned}$$

By a union bound, with probability $1 - \frac{1}{\ell^{3C} \cdot |X|^k}$, $|B_i| \leq v$ for all buckets as well as $|B_J| \leq C_g \cdot k$.

When $|X| > \ell^{0.9}$, the proof follows exactly the same logic as [Lemma 3.2](#). Here we only show the analysis for $|X| \in (\ell^{0.9}, \ell^{1.1})$. Fix $i \in [\ell]$ and define $Z_x := \mathbf{1}(g(x) = i)$. Then $\mathbb{E}_{g:C_g k\text{-wise}}[|B_i| - \mathbb{E}[|B_i|]]^{C_g \cdot k} \leq O(C_g \cdot k \cdot r/\ell)^{C_g \cdot k/2}$. Because $k \ll \ell$, we have

$$\Pr_{g:C_g k\text{-wise}}[|B_i| - \mathbb{E}[|B_i|] \geq \ell^{0.1}] \leq \frac{O(C_g \cdot k \cdot r/\ell)^{C_g \cdot k/2}}{\ell^{0.1 C_g \cdot k}} \leq \frac{O_{C_g}(k^{C_g \cdot k/2})}{\ell^{0.05 C_g \cdot k}} \leq \frac{1}{\ell^{0.01 C_g \cdot k}} \leq \ell^{-(3C+1) \cdot k}.$$

We obtain that with probability $1 - 1/\ell^{3C \cdot k}$, $\max_{i \in [\ell]} |B_i| \leq 2\ell^{0.1}$ after a union bound. \square