The Interdisciplinary Center, Herzliya

Efi Arazi School of Computer Science

M.Sc. Program - Research Track

Generically Increasing Bounded Independence

by

Arbel Deutsch Peled

M.Sc. dissertation, submitted in partial fulfillment of the requirements
for the M.Sc. degree, research track, School of Computer Science
The Interdisciplinary Center, Herzliya

December 2016

This work was carried out under the supervision of Alon Rosen. It is based on joint work with Andrej Bogdanov and Alon Rosen.

Abstract

Almost k-wise independent hash functions are function families whose outputs on any k distinct queries are close to uniform in L_1 distance; this is also called *bounded independence*. A close relative, called almost k-wise unbiased functions, is a family of functions whose outputs on any k distinct queries are close to uniform in L_∞ distance.

In this work we investigated methods for increasing the bounded independence of a family of hash functions. Namely, given an almost k-wise independent (unbiased) function family, we aim to produce an almost k'-wise independent (unbiased) one with $k' > k$. Our transformations are generic in the sense that they only require black-box access to the underlying hash function families, and in most cases only require these to be almost k-wise independent without any further restrictions. To the best of our knowledge, no such method was published to date.

In order to achieve our goals we employed the following method: repeatedly sample from the original function family and define a new function that is some combination of the samples. We identified two types of predicates with which to combine the sampled functions. One type allows one to decrease the bias of the output; the second type allows us to increase the bounded independence parameter k. We finally combine the two types of predicates in an iterative construction which has the required properties.

Contents

1 Introduction

Hash functions are a fundamental element of modern computer science. They have been the subject of extensive research dating back to the 1950s and have seen practical use in diverse settings (see section 1.2). They come in many flavors, which can be roughly divided into two types: information-theoretic functions and cryptographic ones.

In 1979, Carter and Wegman [CW79] defined the concept of *Universal-Hashing*, which started a vast array of publications on this topic. Roughly speaking, Universal Hash Families guarantee that the function outputs are pairwise-independent. This notion was later extended by the same authors to k-wise independence. Families of functions whose outputs are k-wise independent are called k-Universal, or simply k-wise independent.

In simple words, this definition means that the distribution induced by the hash function family over the outputs of up to any k distinct inputs is completely uniform. This notion can be relaxed slightly without losing too much of its power if we only require that the distribution over output tuples, is statistically close to uniform in L_1 norm. That is to say, if we consider distributions over $\{0,1\}^k$ as vectors in $\mathbb{R}^{2^k}$ then the L_1-norm of the difference between the output distribution and the uniform one is small. Families of hash functions that satisfy this constraint are said to be almost k-wise independent.

This notion provides a very strong guarantee on the k-tuple of outputs. Basically, it says that the distinguishing advantage of any, even unbounded, adversary between the k outputs of such a hash family and k completely random elements from the function family's range is small.

A less stringent requirement is that the function family have small bias. We say that a function has small bias if the expectation of any linear test over the components of the output distribution is close 0.5. Families of functions (or distributions) which exhibit this property are said to be almost-unbiased.

It turns out that these two notions are actually related to one another. In [Vaz86], it was shown that an exponentially small bias can be translated into a bound on L_1 distance. This was later improved upon in [Dia88]. Specifically:

Lemma 1.1 (Diaconis-Shahshahani lemma restated)**.** *If $\mathcal{D}$ is a distribution over $\{0,1\}^k$, whose bias with respect to any linear test is at most ϵ: then this $\mathcal{D}$ is also at most $2^{k/2}\epsilon$-far from the uniform distribution in L_1 norm.*

Throughout this text, we use the notion of bias, rather than that of independence, which turns out to be more natural to our analysis. These results can then be transformed via Lemma 1.1 into the language of almost k-wise independence. In some cases we will specify the implications regarding almost k-wise independence explicitly, but in others we will let these results remain implicit.

1.1 Our Results

We consider function families which have small ϵ bias w.r.t. all linear tests of size at most t, where the size of a linear test is the number of components in the output distribution that participate in the test. Such families are called (t, ϵ)-biased. In this thesis we present a generic method for transforming a (t, ϵ)-bias function family into a (t', ϵ')-biased one while only using black-box access to the original family. Our constructions do not change the input or output length of the original family.

We have several motivations for this approach. The first is that, while explicit constructions for k-wise independent hash functions exist, it is plausible that in some applications k-wise hash function are inherent to the problem, but that these functions do not admit a natural method for increasing their independence parameter. In this case using a generic transformation may be useful.

A second, and possibly more important, motivation is in creating almost k-wise unbiased functions which can be computed by small formulae. It has been shown in [RR97] that pseudo-random functions cannot exist in complexity classes that admit Natural proofs. Our smallest constructions lie very close in size to the current known bound for pseudo-random function (PRF) formulae. One may conjecture that almost k-wise independent hash functions are information-theoretic relatives of the computational PRFs. Therefore we hope to construct functions which are not only almost k-wise unbiased but may also be valid PRF candidates.

In section 4 we prove the following theorem:

Theorem 4.1 (loosely stated)**.** *Let $\mathcal{F}_0 \subseteq D \to \{0,1\}$ be a (k_0, ϵ_0)-biased function family for some constants $k_0 \geq 2$ and $\epsilon_0 < 1$. Then it is possible to efficiently construct a family $\mathcal{F} \subseteq D \to \{0,1\}$ that is (k, ϵ)-biased by deterministically combining at most a polynomial (in k and in $\log(\frac{1}{\epsilon})$) number of independent samples from $\mathcal{F}$.*

We first state and prove this theorem for hash functions that hash bit strings into single bits. In section 5 we generalize this to arbitrary input and output sizes.

This theorem follows from two combination lemmas of complementing natures. We show that XOR-ing two samples from a hash function family reduces bias, while AND-ing two samples effectively doubles the independence parameter of the function family while, unfortunately, increasing the bias. Alternating between these two combination methods allows us to increase the independence parameter while controlling the bias. This process is roughly sketched in Fig. 1. In section 3 we state and prove these two lemmas.

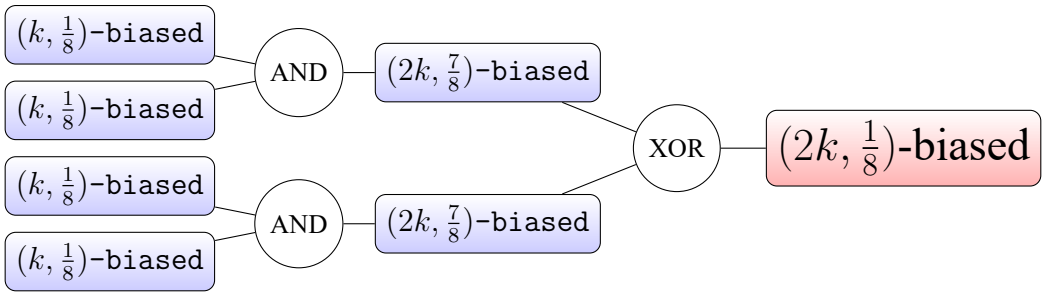

Figure 1: Sketch of a single independence increasing step

This latter process is the heart of our constructions. It is the means by which we increase the titular bounded independence. However, it cannot work on its own; rather, it is the second stage of three.

In the first stage the bias is reduced down to a "small enough" constant using repeated XORs. This is necessary in order to bound the bias in an AND step. The third and final stage is also a bias-reduction one, which is needed since the second stage only provides a constant bias. In most cases we will require sub-constant, and even exponentially small, final bias.

Applying theorem 1.1, we define a new type of hash-function family which is most easily described as a formula, i.e. a circuit in which every gate has fan-out 1. Our construction follows in the footsteps of [Val84]. In this paper the majority function of n bits is shown to be calculable by a monotone formula which uses just one type of gate. The original construction works by sampling with replacement $m > n$ bits from the input and then calculating a formula with the structure of a balanced tree that contains only majority gates.

Our construction uses the same sampling technique, but employs two types of gates: XOR and AND, as in our theorem. By proving that the sampling procedure is effectively an almost pairwise-independent hash function we will be able to directly apply our theorem and show that a suitable, explicit formula over a suitably long series of independent samples from the input is almost k-wise unbiased for some desired value of k and distance from uniformity. This is detailed in section 4.2

We then turn our attention to optimizing the performance of the construction. In this endeavour we have two conflicting goals:

1. Reduce the randomness used in the construction
2. Reduce the construction's formula size

In addition, one may hope to support larger outputs and more general inputs (i.e. not just bit strings).

The first goal is motivated primarily by practical reasons: in many applications it is important to save on randomness. We explore this goal in section 5. First, we improve upon the efficiency of the AND gate. In order to do this, we first need to generalize our construction to handle longer outputs. We treat our hash functions as families $\mathcal{F} : D \to \mathbb{Z}_q$. At this point we replace the AND gate with a different one which enables us to square the independence parameter k instead of just

doubling it. This drastically reduces the required randomness. We also show that addition modulo q generalizes the XOR bias-reduction operation.

Second, we optimize the final bias-reduction stage. We employ the well-known technique of using random walks on expanders instead of independently sampling, first used in [AKS87]. In lemma 5.11 we show that XOR-ing together samples taken from a random walk on an expander graph is almost as good at reducing bias as XOR-ing completely independent samples from the same graph. Using this idea, we are able to reduce the cost of a single XOR gate, which naïvely requires doubling the randomness, to an additive constant number of bits. These two optimizations are generic, and can be applied to almost any hash function family.

In section 6 we first derive a general transformation from (k_0, ϵ_0)-bias to (k, ϵ)-bias using the improvements of the previous section. This implies a similar transformation for k-wise independence. The parameters of our most randomness-efficient constructions are then shown to be only linearly dependent on the final required value of k, and logarithmically dependent on the size of the functions' range.

We then generalize our simple circuit construction from section 4 to allow for multiple output bits, and show how the new theorems relate to this construction. This is done by randomly sampling multiple bits from the input to create a single bit-string output. We then show a further simple optimization of this construction using error-correcting codes. Specifically, we show that by applying a suitable code on the input prior to sampling from it, we can reduce the bias of the bit-string output to a constant. This helps reduce the amount of randomness required by the initial bias-reduction stage.

In section 7 we pursue a tangent direction: improving the formula size of our original $\{0,1\}^n \to \{0,1\}$ hash function family from section 4.2. It has been shown in [RR97] that PRFs cannot exist in complexity classes that have a Natural Property. In [Nec66] a Natural proof was shown giving a lower bound of $O(\frac{n^2}{\log n})$ for the formula size of any function computing the so-called *selection* function. Furthermore, in the case of De-Morgan formulae, in a series of papers ([And87], [IN93] and [Hås98]) a Natural proof was shown giving a lower bound of $O(n^{3-o(1)})$ for the size of such a formula computing some specific function. In this discussion we refer to *general formulae* as ones comprised of AND, OR, XOR and NOT gates. De-Morgan formulae are restricted to AND, OR and NOT gates.

We show constructions of almost n-unbiased hash functions which can be implemented by general formulae of size $\tilde{O}(n^2)$ and by De-Morgan formulae of size $\tilde{O}(n^4)$. By doing so we hope to promote further exploration of this kind of formulae, which may result in either a stronger support for the possibility of constructing such PRFs or, alternatively, in finding Natural proofs of this size. Our results in this area are summarized in theorem 1.2.

Theorem 1.2. *There exists a family of $(n, 2^{-n})$-biased hash functions $\mathcal{F} : \{0,1\}^n \to \{0,1\}$ whose formula size is:*

1. $O(n^2 \log^2 n)$ *for general formulae, and*
2. $O(n^4 \log^2 n)$ *for De-Morgan formulae*

In order to improve upon the naïve construction's formula size we employ two additional techniques. Reducing the starting bias is done using a well-known technique for generating pairwise independent distributions from slightly unbiased ones. Namely, we compute the inner product between the input and a uniformly random vector over the same space. Then we replace our generic, iterative independence-amplification technique with a more standard, but less generic, lemma from [VV86]. The final stage of the construction, the bias-reduction stage, remains untouched. We comment that this seems to be the main bottleneck in any attempt to increase bounded independence.

1.2 Related Work

k-wise independent hashing is very useful in a wide array of fields. To cite a few: It has been shown by [PPR11] that 5-wise independent hash families provide optimal expected time-complexity for adding key-value pairs to hash tables using linear probing. In Cuckoo Hashing introduced by [PR04], when storing n key-value pairs, it is required to have 2 independently chosen $\log n$-wise independent hash functions in order to provide good analytical guarantees for the expected performance of the system. In [TZ04] an online algorithm is shown that estimates the second moments of a stream of data, using 4-wise independent hash functions. Another theorem proved in [Hus+12] states that 2-wise almost independent hash families are Storage Enforcing, meaning that they allow a verifier to ascertain that their data is indeed stored on a server by just saving some small hash of the data.

The standard technique for generating k-wise independent hash functions is the original one introduced in [WC81]. In that paper, in was shown that the family of hash functions defined by all polynomials of degree at most k over some finite field F_p with $p > k$, sampled uniformly at random, yields a k-wise independent hash function family. This construction is very efficient in terms of key size ($k \log p$). Our most efficient constructions have a key size larger by a constant factor from their result. Furthermore, in the case when one is interested only in small bias (as opposed to independence): the key size required by the polynomial-based construction is much larger than the one used by the constructions presented here.

Another good property of the standard construction is that it is also exactly (as opposed to almost) k-wise independent. However, it does not offer any insight into how to combine arbitrary hash function families in order to improve their parameters, which is the main concern of this thesis.

Another common method, which has seen wide application in practice, for generating k-wise independent hash families is Tabulation Hashing. This approach, which has many derivatives (e.g. [TZ04], [PT13]), stores several "small" copies of truly random hash functions over a smaller domain. These functions are stored as input-output tables, which give this method its name. When queried with some input: the function splits it into several smaller parts, queries each table using a different part of the input and then combines the results in a deterministic way. This sort of hash function is only useful for small values of k and constant input length, as it needs to randomly draw and then store tables of exponential size.

The usefulness of these functions comes from their time-efficiency: each one only requires a

constant time to evaluate, and this constant is usually very small. However, from an asymptotic point of view, they are prohibitive. Furthermore, it was shown in [Sie04] that constant evaluation time can only be achieved using exponentially-large storage. Our constructions have logarithmic evaluation time, and the amount of memory the require is polynomial in the output length and independence parameter k.

In our work we use the notion of bias both for its own sake and in order to imply almost-independence. The importance of small-bias distributions was first shown in [NN90], in which probability spaces with this property are constructed using a low amount of entropy. In the following text we make use of the same definitions for bias but do so in a different setting. When we consider k outputs of a hash function over k distinct inputs, we must assume that the hash function is defined in the same way for each of these input-output pairs. Moreover, the i-th output cannot depend on the j-th input, for any $i \neq j$, since this would imply that the construction is not a function. This restriction, which is not present in the setting of the original paper, means that our constructions would be hard-pressed to compete with the original ones in terms of key-size. In fact, they are altogether a different kind of object.

The main topic of this thesis is the introduction of methods that can be used to increase bounded independence. Recently and independently, a similar result was shown in [GV15]. In that paper, distributions over $SL(2,q)^m$ are considered. It was shown that a component-wise product of $2^{\Omega(m)}$ pairwise independent distributions over that domain forms a distribution that is $\frac{1}{|SL(2,q)|}$-close to uniform on $SL(2,q)^m$. The authors of that paper achieve this result via an iterative process which takes some constant number of almost t-independent distributions and outputs an almost $(t+1)$-independent one. The motivation in that case was completely different, and the increase in independence was the means to a very specific end, which we refrain from mentioning here for brevity.

In contrast, our methods allow for a doubly-exponential increase in the parameter t using a small constant number of samples from a (t, ϵ)-biased distribution, for a suitably small ϵ. This means that the same kind of results can be achieved using a much lower number of samples from the original distribution. Our results are also more generic since they work over any group $\mathbb{Z}_q$. They are not, however, directly applicable to the objects considered in [GV15], since in that case there is no control over which function to use when combining the different samples.

2 Preliminaries

2.1 Notation

2.1.1 General

We denote scalars in lower-case (e.g x) and vectors in **bold** (e.g. $\boldsymbol{x}$ or $\boldsymbol{X}$). Distributions are designated by a calligraphic font (e.g. $\mathcal{D}$). Random variables are denoted by upper-case letters (e.g. X or $\boldsymbol{X}$).

The uniform distribution over n bits is denoted by $\mathcal{U}_n$. The uniform distribution over a set S is

denoted by $\mathcal{U}_S$. When the domain is obvious from context, we sometimes simply write $\mathcal{U}$.

For any natural number $n \in \mathbb{N}$ we denote by $[n]$ the set $\{1, 2, 3, \ldots, n\}$.

If $\boldsymbol{x}$ is a vector, i is an integer and S is a set of integers then we denote by v_i the value of $\boldsymbol{x}$ in the i-th coordinate and by $\boldsymbol{x_S}$ the restriction of $\boldsymbol{x}$ to the coordinates in S.

2.1.2 Distributions and Hash Functions

For a distribution $\mathcal{D}$ over domain D we denote for any element $x \in D$ its probability of being drawn from $\mathcal{D}$ by: $\mathcal{D}(x) = \Pr_{X \sim \mathcal{D}}[X = x]$.

Definition 2.1. *The* support *of $\mathcal{D}$:* $\text{Supp}(\mathcal{D}) = \{x \in D \,|\, \mathcal{D}(x) > 0\}$ *is the set of elements in the domain which have non-zero probability of being drawn from $\mathcal{D}$.*

The following definitions form the basic properties we expect our hash families to have.

Definition 2.2 (Distribution induced by a Function Family)**.** *Let $\mathcal{F} \in D \times \{0,1\}^r \to R$ be some family of functions. We define the* distribution induced by $\mathcal{F}$ and inputs $\boldsymbol{x} = \{x_1, x_2, \ldots, x_k\} \in D^k$*, $\mathcal{D}_{\mathcal{F},\boldsymbol{x}} : R^k \to [0,1]$, as a probability distribution over the outputs of the hash function:*

$$\mathcal{D}_{\mathcal{F},\boldsymbol{x}}(\boldsymbol{y}) = \Pr_{\rho \xleftarrow{R} \{0,1\}^r}[\forall i : \mathcal{F}(x_i, \boldsymbol{\rho}) = y_i]$$

Definition 2.3 (L_1 distance for distributions)**.** *Let $\mathcal{D}_1$ and $\mathcal{D}_2$ be two probability distributions over the same domain D. Then the L_1 distance between the distributions is defined as:*

$$|\mathcal{D}_1 - \mathcal{D}_2| = \sum_{x \in D} |\mathcal{D}_1(x) - \mathcal{D}_2(x)| = \sum_{x \in D} \left| \Pr_{X \overset{R}{\sim} \mathcal{D}_1}[X = x] - \Pr_{X \overset{R}{\sim} \mathcal{D}_2}[X = x] \right|$$

Definition 2.4 (Almost k-Wise-Independent Hash Function Family)**.** *A family of functions $\mathcal{F} : D \times \{0,1\}^r \to R$ is said to be* (k, ϵ)-Wise-Independent *if for any set of k* distinct *inputs $\boldsymbol{x} = \{x_1, x_2, \ldots, x_k\} \in D^k$, it holds that:*

$$|\mathcal{D}_{\mathcal{F},\boldsymbol{x}} - \mathcal{U}_{R^k}| \leq \epsilon$$

In order to analyze the behaviour of a family of hash-functions on several inputs we extend the definition of a hash family in the natural manner. If $\mathcal{F}$ is a family of hash functions $\mathcal{F} : D \times \{0,1\}^r \to R$ then we define for all $F \in \mathcal{F}$ and $\boldsymbol{x} \in D^k$:

$$F(\boldsymbol{x}) = (F(x_1), F(x_2), \ldots, F(x_k))$$

The latter definition simply states that each input is handled separately as originally defined and the different outputs are outputted as a vector.

Definition 2.5. *We say a distribution $\mathcal{D}$ over $\{0,1\}^k$ is* Symmetric *iff for all $\boldsymbol{x} \in \{0,1\}^k$:*

$$\mathcal{D}(\boldsymbol{x}) = \mathcal{D}(\boldsymbol{x^c})$$

Where $\boldsymbol{x^c} = (1 - x_1, 1 - x_2, \ldots, 1 - x_k)$.

2.2 Fourier Expansion

The proofs of the main lemmas rely heavily on the Fourier expansion of Boolean functions and the notion of bias, which are defined next. The following is only be a cursory introduction to the subject. For more details, we refer the reader to [ODo14]. In section 5 we further generalize these definitions to the non-Boolean case, however the special case described here suffices for the coming sections.

Definition 2.6 (Linear Boolean Functions)**.** *For every $\boldsymbol{x} \in \{0,1\}^n$ and every subset $S \subseteq [n]$ we define the linear function $\chi_S(\boldsymbol{x})$ by:*

$$\chi_S(\boldsymbol{x}) \triangleq \bigoplus_{i \in S} x_i$$

By convention: $\chi_\phi(\boldsymbol{x}) = 0$ for all $\boldsymbol{x}$.

Note 2.1. *A linear test is defined by the subset S of indices included in the sum. This parameter S can be viewed either as a set $S \subseteq [n]$ or as an indicator vector $S \in \{0,1\}^n$. Although the definition was given, for clarity's sake, in the first form, we actually be use the second one more frequently. This allows us to view the input to the function χ_S and the definition S of the function itself as vectors over the same domain $\{0,1\}^n$. This, in turn, yields the equation $\chi_S(\boldsymbol{x}) = \langle S, \boldsymbol{x} \rangle$.*

The Fourier expansion of Boolean functions is written in a different basis from the usual one. The values $\{0,1\}$ are mapped to the values $\{1,-1\}$, respectively. One can verify that under this mapping, if x, y are Boolean variables then $x \oplus y = x \cdot y$, where $\oplus$ is the usual XOR and $\cdot$ is multiplication over the reals.

Functions defined over $D \to \{-1,1\}$ can be represented as vectors in $\{-1,1\}^{|D|}$ where each coordinate stores the function's value for its respective input. It turns out that any Boolean function $f : \{-1,1\}^n \to \{-1,1\}$ can be written as a weighted sum of the linear functions.

Fact 2.1 (Fourier Expansion of a Boolean Function)**.** *For all $f : \{-1,1\}^n \to \{-1,1\}$ there exists a function $\hat{f} : \{-1,1\}^n \to [-1,1]$ s.t. for all $\boldsymbol{x} \in \{-1,1\}^n$:*

$$f(\boldsymbol{x}) = \sum_{S \subseteq [n]} \hat{f}(S) \cdot \chi_S(\boldsymbol{x}) = \sum_{S \subseteq [n]} \hat{f}_S \cdot \prod_{i \in S} x_i$$

$\hat{f}$ is called the Fourier decomposition *of f, and its values are given by:*

$$\hat{f}(S) = \langle f, \chi_S \rangle$$

In this last equation we used the vector representation of Boolean functions,

2.3 Bias

The following notion of bias of a distribution w.r.t some linear test captures how much the test can help distinguish between that distribution and the uniform one.

Definition 2.7 (Bias with respect to a linear test)**.** *For any distribution $\mathcal{D}$ over domain $\{-1,1\}^k$ and any set $S \subseteq [k]$ s.t. $S \neq \phi$, we say $\mathcal{D}$ is* ϵ-biased w.r.t to test χ_S *if:*

$$\left| \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathcal{D}} [\chi_S(\boldsymbol{x})] \right| \leq \epsilon$$

The next notion we define here is central to our arguments. It allows us to aggregate bias bounds over many tests, which proves to be extremely useful.

Definition 2.8. *A distribution $\mathcal{D}$ over $\{-1,1\}^k$ is said to be* (t,ϵ)-biased *if for any subset of indices $S \in [k]$ s.t. $0 < |S| \leq t$: $\mathcal{D}$ is ϵ-biased w.r.t χ_S.*

We say a distribution $\mathcal{D}$ over $\{-1,1\}^k$ is ϵ-*biased* if it is (k,ϵ)-biased.

One nice property of linear tests is that their output is completely uniform when given a uniformly random input. This property is one of the incentives for this definition. The exception to this rule is the trivial linear test χ_ϕ, which is always constant. This is why the definition only includes linear tests S of size $|S| > 0$.

Definition 2.9. *A function family $\mathcal{F} : \{-1,1\}^n \times \{-1,1\}^r \to \{-1,1\}$ is said to be* (t,ϵ)-biased *if for any set of up to t inputs $\boldsymbol{x}$, $\mathcal{D}_{\mathcal{F},\boldsymbol{x}}$ is (t,ϵ)-biased.*

We note that (t,ϵ)-independence is stronger than, and indeed implies, (t,ϵ)-bias. The converse is not true.

3 Basic Composition Lemmas

In this section we analyze the effects of the functions XOR and AND on (t,ϵ)-biased distributions. We show that XOR reduces bias, while AND increases the independence parameter t. For simplicity's sake, we first assume that the input distribution is symmetric. In later sections we remove this restriction.

We begin with the analysis of the XOR function. One of the properties of this function is that if we take the XOR of two independent distributions, then the bias of the resultant distribution w.r.t. any linear test is at most the minimum of the biases of the original distributions w.r.t. the same test. This property will be quite useful to us both in producing a generic construction and when generalizing to a construction that recursively uses just one type of gate.

3.1 Effects of XOR

Claim 3.1. *Let $\mathcal{D}$ be any symmetric probability distribution over $\{-1,1\}^k$ and $\mathcal{D}'$ be any, not necessarily symmetric, probability distribution over the same domain. Then* $\mathrm{XOR}(\mathcal{D},\mathcal{D}')$ *is a symmetric distribution.*

Proof. Let $\boldsymbol{x} \in \text{Supp}\left(\text{XOR}(\mathcal{D}, \mathcal{D}')\right)$, and let A be the set of all pairs of inputs $(\boldsymbol{x_1}, \boldsymbol{x_2}) \in \{-1,1\}^k \times \{-1,1\}^k$ s.t. $\boldsymbol{x_1} \oplus \boldsymbol{x_2} = \boldsymbol{x}$. Then:

$$\begin{aligned}
\Pr_{\boldsymbol{X} \sim \text{XOR}(\mathcal{D},\mathcal{D}')}[\boldsymbol{X} = \boldsymbol{x}] &= \Pr_{\boldsymbol{X_1} \sim \mathcal{D}, \boldsymbol{X_2} \sim \mathcal{D}'}[\boldsymbol{X_1} \oplus \boldsymbol{X_2} = \boldsymbol{x}] \\
&= \Pr_{\boldsymbol{X_1} \sim \mathcal{D}, \boldsymbol{X_2} \sim \mathcal{D}'}[\boldsymbol{X_1^c} \oplus \boldsymbol{X_2} = \boldsymbol{x}] \\
&= \Pr_{\boldsymbol{X_1} \sim \mathcal{D}, \boldsymbol{X_2} \sim \mathcal{D}'}\left[\boldsymbol{X_1} \oplus \boldsymbol{X_2} \oplus -\boldsymbol{1^k} = \boldsymbol{x}\right] \\
&= \Pr_{\boldsymbol{X} \sim \text{XOR}(\mathcal{D},\mathcal{D}')}[\boldsymbol{X} = \boldsymbol{x^c}]
\end{aligned}$$

Where the second equality follows from the assumption that $\mathcal{D}$ is symmetric.

□

Lemma 3.2 (XOR Bias reduction)**.** *Let $S \subseteq [k]$ be a set of indices, $\mathcal{D}$ a distribution over $\{-1,1\}^k$ that is ϵ-biased w.r.t χ_S and $\mathcal{D}'$ a distribution over $\{-1,1\}^k$ that is ϵ'-biased w.r.t χ_S Then:* $\text{XOR}\left(\mathcal{D}, \mathcal{D}\right)$ *is $(\epsilon \cdot \epsilon')$-biased w.r.t. χ_S.*

Proof.

$$\begin{aligned}
\left|\mathop{\mathbb{E}}_{\boldsymbol{X} \sim \text{XOR}(\mathcal{D},\mathcal{D}')}[\chi_S(\boldsymbol{X})]\right| &= \left|\mathop{\mathbb{E}}_{\boldsymbol{X} \sim \text{XOR}(\mathcal{D},\mathcal{D}')}\left[\prod_{i \in S} X_i\right]\right| \\
&= \left|\mathop{\mathbb{E}}_{\boldsymbol{Y} \sim \mathcal{D}, \boldsymbol{Z} \sim \mathcal{D}'}\left[\prod_{i \in S} (Y_i \cdot Z_i)\right]\right| \\
&= \left|\mathop{\mathbb{E}}_{\boldsymbol{Y} \sim \mathcal{D}, \boldsymbol{Z} \sim \mathcal{D}'}\left[\prod_{i \in S} Y_i \cdot \prod_{i \in S} Z_i\right]\right| \\
&= \left|\mathop{\mathbb{E}}_{\boldsymbol{Y} \sim \mathcal{D}}\left[\prod_{i \in S} Y_i\right] \cdot \mathop{\mathbb{E}}_{\boldsymbol{Z} \sim \mathcal{D}'}\left[\prod_{i \in S} Z_i\right]\right| \\
&= \left|\mathop{\mathbb{E}}_{\boldsymbol{Y} \sim \mathcal{D}}\left[\prod_{i \in S} Y_i\right]\right| \cdot \left|\mathop{\mathbb{E}}_{\boldsymbol{Z} \sim \mathcal{D}}\left[\prod_{i \in S} Z_i\right]\right| \\
&< \epsilon \cdot \epsilon'
\end{aligned}$$

Where the 4-th equality follows from the independence of $\boldsymbol{Y}$ and $\boldsymbol{Z}$ and the last inequality follows from the assumption on $\mathcal{D}$.

□

This lemma has two immediate and useful corollaries:

Corollary 3.2.1. *If $\mathcal{D}$ is ϵ biased w.r.t. χ_S then:*

1. *$\mathrm{XOR}(\mathcal{D}, \mathcal{D})$ is ϵ^2-biased w.r.t. χ_S.*

2. *For any distribution $\mathcal{D}'$ defined over the same domain: $\mathrm{XOR}(\mathcal{D}, \mathcal{D}')$ is ϵ-biased w.r.t. χ_S.*

3.2 Effects of AND

In this section we prove the following proposition.

Proposition 3.3. *If $\mathcal{D}$ is a symmetric, (t, ϵ)-biased probability distribution, then: $\mathrm{XOR}(\mathcal{D}, \mathrm{AND}(\mathcal{D}, \mathcal{D}))$ is symmetric and $(2t, \frac{1}{2} + 2^{-t} + \epsilon)$-biased, where the samples from $\mathcal{D}$ are independent.*

The next lemma establishes the simplest form of our main idea, and can be treated as a proof-of-concept. It basically states that AND-ing together two samples from a (t, ϵ)-biased distribution, for $t \geq 2$ and a suitably small value ϵ, forms a $(2t, c)$-biased distribution for some not-too-large constant c.

Lemma 3.4. *Let $t \in \mathbb{N}$ and let $\mathcal{D}$ be a symmetric, (t, ϵ)-biased probability distribution over $\{-1, 1\}^k$. Then for any $S \subseteq [k]$ s.t. $0 < |S| \leq 2t$: $\mathrm{AND}(\mathcal{D}, \mathcal{D})$ is $\frac{1}{2} + \epsilon + 2^{-\min(t, |S|)}$-biased w.r.t χ_S.*

Note that this result gives only the trivial bound on the bias of tests S of size 1. We can overcome this by XOR-ing with an additional copy from the original distribution. This also has the effect of making the final distribution symmetric once again. With this in mind, we now prove the proposition.

Proof of proposition 3.3. Let $S \subseteq [k]$ s.t. $0 < |S| \leq 2t$. If $|S| \geq t$ then we get the required bias from lemma 3.4. If $|S| < t$, then we have two options:

1. If $\epsilon > \frac{1}{2}$ then the bound specified by the proposition is meaningless, since it bounds a probability by a constant larger than 1, which is a tautology.

2. If $\epsilon < \frac{1}{2}$ then the bias of w.r.t S must be at most $\frac{1}{2}$ by corollary 3.2.1 and the fact that $\mathcal{D}$ is (t, ϵ)-biased.

Finally, by claim 3.1: since $\mathcal{D}$ is symmetric: so is $XOR(\mathcal{D}, AND(\mathcal{D}, \mathcal{D}))$. □

We now turn to proving the lemma.

Proof of Lemma 3.4. The intuition behind the lemma is the following. Let $\boldsymbol{x}, \boldsymbol{y}$ be independent samples from $\mathcal{D}$. Whenever a bit $x_i = 1$: then $\mathrm{AND}(x_i, y_i) = 1$ as well. This means that this bit does not change the result of the test χ_S.

Let $\mathrm{wt}(\boldsymbol{x_S})$ denote the weight of $\boldsymbol{x_S}$, which is the number of bits in $\boldsymbol{x_S}$ which are equal to -1. If $1 \leq \mathrm{wt}(\boldsymbol{x_S}) \leq t$, then $\chi_S(\mathrm{AND}(\boldsymbol{x}, \boldsymbol{y}))$ is a linear test of size between 1 and t over $\boldsymbol{y}$, and its bias

is be bounded by ϵ. We now show this more rigorously and then proceed to bound the probability that $\text{wt}(\boldsymbol{x})$ is outside the interval $[1, t]$.

Let $P_1, P_2 \subseteq \{-1,1\}^k$ s.t. $P_2 = \{\boldsymbol{x} \in \{-1,1\}^k \mid \boldsymbol{x}^{\mathbf{c}} \in P_1\}$, $P_1 \cap P_2 = \phi$ and for all $\boldsymbol{x} \in P_1$ it holds that $\text{wt}(\boldsymbol{x_S}) \leq t$. Such sets can be constructed by greedily choosing pairs of complementing vectors from $\{-1,1\}^k$ and placing each element in the correct set, or placing them randomly when both options are valid.

Since $\mathcal{D}$ is symmetric:

$$\Pr_{\boldsymbol{X}\sim\mathcal{D}}[\boldsymbol{X} \in P_1] = \Pr_{\boldsymbol{X}\sim\mathcal{D}}[\boldsymbol{X} \in P_2]$$

Therefore:

$$\Pr_{\boldsymbol{X}\sim\mathcal{D}}[\text{wt}(\boldsymbol{X_S}) \leq t] \geq \frac{1}{2}$$

Thus:

$$\begin{aligned}
\left|\mathop{\mathbb{E}}_{\boldsymbol{X}\sim\text{AND}(\mathcal{D},\mathcal{D})}\left[\prod_{i\in S} X_i\right]\right| &= \left|\mathop{\mathbb{E}}_{\boldsymbol{Y},\boldsymbol{Z}\sim\mathcal{D}}\left[\prod_{i\in S} Y_i \wedge Z_i\right]\right| \\
&\leq \left|\Pr_{\boldsymbol{Z}\sim\mathcal{D}}[\text{wt}(\boldsymbol{X_S}) > t] \cdot \mathop{\mathbb{E}}_{\boldsymbol{Y},\boldsymbol{Z}\sim\mathcal{D}}\left[\prod_{i\in S} Y_i \wedge Z_i \middle| \boldsymbol{Z} \in P_2\right]\right| \\
&\quad + \left|\Pr_{\boldsymbol{Z}\sim\mathcal{D}}[1 \leq \text{wt}(\boldsymbol{Z}) \leq t] \cdot \mathop{\mathbb{E}}_{\boldsymbol{Y},\boldsymbol{Z}\sim\mathcal{D}}\left[\prod_{i\in S} Y_i \wedge Z_i \middle| 1 \leq \text{wt}(\boldsymbol{Z}) \leq t\right]\right| \quad (1) \\
&\quad + \left|\Pr_{\boldsymbol{Z}\sim\mathcal{D}}[\text{wt}(\boldsymbol{Z}) = 0] \cdot \mathop{\mathbb{E}}_{\boldsymbol{Y},\boldsymbol{Z}\sim\mathcal{D}}\left[\prod_{i\in S} Y_i \wedge Z_i i \middle| \text{wt}(\boldsymbol{Z}) = 0\right]\right| \\
&\leq \frac{1}{2} + \frac{1}{2}\epsilon + \Pr_{\boldsymbol{Z}\sim\mathcal{D}}[\text{wt}(\boldsymbol{Z}) = 0]
\end{aligned}$$

Where the final inequality is by $\boldsymbol{Y}$ being ϵ-biased w.r.t. tests of size between 1 and t, and by the trivial bound on the bias in the other cases.

We now need a bound on the probability that a random sample from $\mathcal{D}$, restricted to S has all of its components equal to 0. The following claim, which is subsequently proved, establishes the required bound.

Claim 3.5. *Let $\mathcal{D}$ be a (t,ϵ)-biased probability distribution over $\{-1,1\}^k$ and $S \in \{-1,1\}^k$. Then:*

$$\Pr_{\boldsymbol{X}\sim\mathcal{D}}\left[\boldsymbol{X_S} = \mathbf{1}^{|\boldsymbol{S}|}\right] \leq 2^{-\min(t,|S|)} + \frac{1}{2}\epsilon$$

Assigning the values from this claim into Eq. (1) yields the lemma.

□

Proof of Claim 3.5. If $|S| > t$, then we can restrict $\boldsymbol{X}$ to some subset $S' \subseteq S$ such that $|S'| = t$. We then need only prove that $\Pr_{\boldsymbol{X}\sim\mathcal{D}}\left[\boldsymbol{X_{S'}} = \mathbf{1}^{|\boldsymbol{S'}|}\right] \leq 2^{-t} + \frac{1}{2}\epsilon$. Instead, and without loss of generality, we will assume that $|S| \leq t$ and prove the originally stated inequality.

Let $\mathrm{OR} : \{-1,1\}^{|S|} \to \{-1,1\}$ be the binary OR function on $|S|$ bits and let $\widehat{\mathrm{OR}} : \{-1,1\}^S \to [-1,1]$ denote its Fourier decomposition. Notice that $\boldsymbol{X_S} = \mathbf{1^t}$ iff $\mathrm{OR}_{i\in S} X_i = 1$. Then:

$$\begin{aligned}
&\left| \mathop{\mathbb{E}}_{\boldsymbol{X}\sim\mathcal{D}} [\mathrm{OR}_{i\in S} X_i] - \mathop{\mathbb{E}}_{\boldsymbol{X}\sim\mathcal{U}} [\mathrm{OR}_{i\in S} X_i] \right| = \\
&\qquad = \left| \mathop{\mathbb{E}}_{\boldsymbol{X}\sim\mathcal{D}} \left[\sum_{M\subseteq S} \left(\widehat{\mathrm{OR}}(M) \cdot \prod_{i\in M} X_i \right) \right] - \mathop{\mathbb{E}}_{\boldsymbol{X}\sim\mathcal{U}} \left[\sum_{M\subseteq S} \left(\widehat{\mathrm{OR}}(M) \cdot \prod_{i\in M} X_i \right) \right] \right| \\
&\qquad = \Bigg| \widehat{\mathrm{OR}}(\phi) + \sum_{\phi\neq M\subseteq S} \left(\widehat{\mathrm{OR}}(M) \cdot \mathop{\mathbb{E}}_{\boldsymbol{X}\sim\mathcal{D}} \left[\prod_{i\in M} X_i \right] \right) \\
&\qquad\quad - \widehat{\mathrm{OR}}(\phi) + \sum_{\phi\neq M\subseteq S} \left(\widehat{\mathrm{OR}}(M) \cdot \mathop{\mathbb{E}}_{\boldsymbol{X}\sim\mathcal{U}} \left[\prod_{i\in M} X_i \right] \right) \Bigg| \\
&\qquad \leq \epsilon \left| \sum_{\phi\neq M\subseteq S} \widehat{\mathrm{OR}}(M) \right| \\
&\qquad \leq \epsilon
\end{aligned}$$

In which the penultimate inequality stems from our assumption that $\mathcal{D}$ is (t,ϵ)-biased, the fact that $|S| \leq t$ and the fact that the uniform distribution is unbiased w.r.t. all (non-trivial) linear tests. The final inequality stems from the following well-known fact about the Fourier expansion of the OR functions:

Fact 3.6. *Let $\hat{V} : \{0,1\}^{[t]} \to [-1,1]$ be the Fourier coefficients of the OR function on t bits. Then:*

$$0 \leq \sum_{\phi\neq S\subseteq[t]} \hat{M}(S) \leq 1$$

Now, notice that

$$\mathop{\mathbb{E}}_{\boldsymbol{X}\sim\mathcal{U}} \left[\bigvee_{i\in S} X_i \right] = 1 \cdot 2^{-|S|} - 1 \cdot (1 - 2^{-|S|}) = -1 + 2^{1-|S|}$$

It therefore follows that:

$$\begin{aligned}
2 \cdot \Pr_{\boldsymbol{X}\sim\mathcal{D}} \left[\bigvee_{i\in S} X_i = 1 \right] - 1 &= \Pr_{\boldsymbol{X}\sim\mathcal{D}} \left[\bigvee_{i\in S} X_i = 1 \right] - \Pr_{\boldsymbol{X}\sim\mathcal{D}} \left[\bigvee_{i\in S} X_i = -1 \right] \\
&= \mathop{\mathbb{E}}_{\boldsymbol{X}\sim\mathcal{D}} \left[\bigvee_{i\in S} X_i \right] \\
&\leq -1 + 2^{1-|S|} + \epsilon
\end{aligned}$$

Which, in turn, means that:

$$\begin{aligned}\Pr_{\boldsymbol{X}\sim\mathcal{D}}\left[\boldsymbol{X_S}=\mathbf{1}^{|\boldsymbol{S}|}\right] &= \Pr_{\boldsymbol{X}\sim\mathcal{D}}\left[\bigvee_{i\in S} X_i = 1\right] \\ &\leq \frac{1}{2}\left(2^{1-|S|}+\epsilon\right) \\ &= 2^{-|S|}+\frac{1}{2}\epsilon \\ &= 2^{-\min(t,|S|)}+\frac{1}{2}\epsilon\end{aligned}$$

□

4 Basic Construction

In this section we provide explicit constructions which serve as a basis for our more efficient versions. We first show how to apply the lemmas of the previous section to increasing the bounded independence of any hash function family which takes bit strings and maps them into single bits. We then show how these results may be applied in creating a hash function family which closely resembles the construction of [Val84].

4.1 General transformation

In this section we present our first theorem, which puts the lemmas and corollaries from the previous section into a useful framework. We make no attempts at this point to optimize the construction in any way. In particular, we only deal with hash function families which are defined over bit strings and have single output bits. More general and efficient constructions are deferred to later sections.

We also note that in this section we use the more readily-familiar $\{0,1\}$-basis since all calculations using the Fourier decomposition are contained in the previous section.

Theorem 4.1. *Let $\mathcal{F}:\{0,1\}^n\times\{0,1\}^r\to\{0,1\}$ be a family of (k_0,ϵ_0)-wise independent hash functions, with $k_0\geq 2$ and $\epsilon_0<1$. Then for all k and all $\epsilon>0$: it is possible to explicitly and efficiently construct a family $\mathcal{F}:\{0,1\}^n\times\{0,1\}^r\to\{0,1\}$ that is (k,ϵ)-wise independent with $r=\text{poly}\left(k,\log\frac{1}{\epsilon},\frac{1}{\log\frac{1}{\epsilon_0}}\right)\cdot r_0$ using only black-box access to $\mathcal{F}_0$.*

Proof. We construct the new hash function in three stages:

1. Reduce the initial bias enough to use corollary 3.3, by repeatedly XOR-ing samples from $\mathcal{F}_0$.
2. Repeat the following two sub-steps until the independence parameter reaches k:

(a) AND two samples from the current hash function family in order to increase the independence parameter.

(b) Apply XOR on the result sufficiently many times to reduce the bias back to a useful value.

3. Successively apply XOR steps in order to reduce the bias enough to imply ϵ-wise independence.

This process is described in detail in figure 2.

Input: (k_0, ϵ_0)-biased hash family $\mathcal{F}_0 : \{0,1\}^n \times \{0,1\}^{r_0} \to \{0,1\}$.

Output: (k, ϵ)-biased hash family $\mathcal{F} : \{0,1\}^n \times \{0,1\}^r \to \{0,1\}$.

Phase 1: Initial bias reduction

- Repeat until $\epsilon_i \leq \frac{1}{8}$:
  - $\mathcal{F}_i \leftarrow \text{XOR}(\mathcal{F}_{i-1}, \mathcal{F}_{i-1})$

Phase 2: increasing the independence parameter

- Repeat until $k_i \geq k$:
  - $\mathcal{F}_i \leftarrow \text{XOR}(\mathcal{F}_{i-1}, \text{AND}(\mathcal{F}_{i-1}, \mathcal{F}_{i-1}))$
  - Repeat until $\epsilon_i \leq \frac{1}{8}$:
    - $\mathcal{F}_i \leftarrow \text{XOR}(\mathcal{F}_{i-1}, \mathcal{F}_{i-1})$

Phase 3: Final bias reduction

- Repeat until $\epsilon_i \leq 2^{-k/2}\epsilon$:
  - $\mathcal{F}_i \leftarrow \text{XOR}(\mathcal{F}_{i-1}, \mathcal{F}_{i-1})$

Figure 2: A basic algorithm for increasing bounded independence

First note that if the algorithm terminates then its output is $(k, 2^{-k/2}\epsilon)$-biased. By lemma 1.1 this makes it also (k, ϵ)-wise independent, as required. Now, by proposition 3.3 and corollary 3.2.1: each of the loops in the algorithm always terminates. Therefore, the algorithm is correct. We now analyze the amount of randomness required by it.

In the first stage of the construction we reduce the bias down to $\frac{1}{8}$. By corollary 3.2.1, after s_1 iterations of XOR-ing, the new hash function family $\mathcal{F}_{s_1}$ is (k_0, ϵ_{s_1})-biased for $\epsilon_{s_1} = \epsilon_0^{2^{s_1}}$. We

therefore require $s_1 = \max(0, \log\log 8 - \log\log \frac{1}{\epsilon_0})$ such steps, each of which requires two samples from the previous step's distribution.

In the second stage we sample 3 times from our new hash function family and then combine these samples as suggested in 3.3: $\text{XOR}(f_1, \text{AND}(f_2, f_3))$. This yields a $(2k_0, \frac{7}{8})$-biased hash function. Applying 4 rounds of XOR results in a $(2k_0, \frac{1}{8})$-biased hash function which can now be used in the same process to produce a $(4k_0, \frac{1}{8})$-biased family and so forth. The number of steps required in this stage is $s_2 = \log k - \log k_0$, where each step needs $3 \cdot 2^4 = 48$ samples from the previous step's distribution.

The final stage is similar to the first and requires $s_3 = \max(0, \log\log \frac{2^{k/2}}{\epsilon} - \log\log 8)$ steps.

We are now ready to calculate the amount of randomness required by this construction. For simplicity, we assume that $\epsilon_0 > \frac{1}{8}$ and that $\epsilon < \frac{1}{8}$.

$$
\begin{aligned}
r' &= 3^{s_2} \cdot 2^{s_1 + 4s_2 + s_3} \cdot r \\
&= 3^{\log k - \log k_0} \cdot 2^{\log\log 8 - \log\log \frac{1}{\epsilon_0} + 4\log k - 4\log k_0 + \log\log\left(2^{k/2}\cdot\frac{1}{\epsilon}\right) - \log\log\frac{1}{8}} \cdot r \\
&= \left(\frac{k}{k_0}\right)^{\log_2 3} \cdot \left(\frac{k}{k_0}\right)^4 \cdot \frac{1}{\log\frac{1}{\epsilon_0}} \cdot \left[\frac{1}{2}k + \log\frac{1}{\epsilon}\right] \cdot r \\
&= o\left(\frac{k^6 r}{\log\frac{1}{\epsilon_0}} \cdot \left(k + \log\frac{1}{\epsilon}\right)\right)
\end{aligned}
$$

□

The construction presented above can be viewed as a complete, balanced tree which has the property that all nodes of equal depth contain the same type of gate. This simple structure can be made even simpler at a further cost to the efficiency of the construction.

Specifically, consider the following gate:

$$\text{Universal}(x_1, x_2, x_3, x_4) = (x_1 \wedge x_2) \oplus x_3 \oplus x_4$$

Claim 4.2. *If all the gates in the construction of theorem 4.1 are replaced with the* Universal *gate then the final output is still* (k, ϵ)*-wise-independent.*

Proof. By corollary 3.2.1: $\text{Universal}(x_1, x_2, x_3, x_4)$ has bias not larger than that of $(x_1 \wedge x_2) \oplus x_3$ or that of $(x_3 \oplus x_4)$. Since these are the only types of gates used in the original construction, the claim follows. □

4.2 A new Family of Hash Functions

In [Val84], it was shown that a polynomial-size, monotone formula can be used to calculate the majority function over n bits. This was done by sampling these bits independently at random some

$m > n$ times and calculating a formula on these m bits. At this point a probabilistic argument was used to prove that there exists a random seed such that the formula computes the majority of the inputs for *all* possible inputs.

We will use the same random-sampling technique to generate a "seed" $(2, \epsilon)$-independent hash function family. Our generic construction from theorem 4.1 can then be applied on this "seed" to achieve (k, ϵ)-independence for any $k = \text{poly}(n)$ and $\epsilon \geq \Omega(2^{-2^{\text{poly}(n)}})$.

Explicitly, let $\{0,1\}^n$ be the domain of the hash function family, and let k and ϵ be some target parameters. We define the following hash function family: $\mathcal{F} : \{0,1\}^n \times [2n+2] \to \{0,1\}$ as:

$$\mathcal{F}(\boldsymbol{x}, \rho) = \begin{cases} x_i & \rho \leq n \\ 1 - x_i & n+1 \leq \rho \leq 2n \\ 0 & \rho = 2n+1 \\ 1 & \rho = 2n+2 \end{cases}$$

Where sampling a hash function from the family is done by sampling $\rho \xleftarrow{R} [2n+2]$.

Note that any function from this family can be implemented as a formula with at most 2 inputs, one of which is the constant 1, and at most a single XOR gate. Also, the number of bits required to sample from this hash function family is $r = \log(2n+2) = O(\log n)$.

Claim 4.3. *$\mathcal{F}$ has the following two properties:*

1. *For all $k > 0$ and all $\boldsymbol{x} \in \{0,1\}^k$: $\mathcal{D}_{\mathcal{F},\boldsymbol{x}}$ is symmetric.*
2. *$\mathcal{F}$ is $(2, 1 - \frac{2}{n+1})$-biased.*

Combining claim 4.3 with theorem 4.1 yields a family of (k, ϵ)-wise independent hash functions. This construction has a structure similar to the formula for the majority function presented in [Val84].

Proof. Let $\rho \in [n+1]$. Then for all $\boldsymbol{x}$:

$$\mathcal{F}(\boldsymbol{x}, \rho) = \mathcal{F}(\boldsymbol{x}, \rho + n + 1)^c$$

This concludes the proof of item 1 above.

Claim 4.4. *If a distribution $\mathcal{D}$ over $\{-1,1\}^k$ is symmetric, then it is also $(1,0)$-biased.*

Using this claim, which is proved shortly, we immediately obtain that $\mathcal{F}$ is $(1,0)$-biased.

Now let $\boldsymbol{X} = (\boldsymbol{y}, \boldsymbol{z}) \in (\{0,1\}^n)^2$ be a set of two distinct inputs to the function. The pairwise bias of $\mathcal{F}$ is:

$$\left| \Pr_{\rho \xleftarrow{R} [2n+2]} [\mathcal{F}(\boldsymbol{X}, \rho)_1 \oplus \mathcal{F}(\boldsymbol{X}, \rho)_2 = 1] - \Pr_{\rho \xleftarrow{R} [2n+2]} [\mathcal{F}(\boldsymbol{X}, \rho)_1 \oplus \mathcal{F}(\boldsymbol{X}, \rho)_2 = 0] \right|$$

Since $\boldsymbol{y} \neq \boldsymbol{z}$: there exists at least one index $i \in [n]$ s.t. $y_i \neq z_i$. Without loss of generality, let us assume that $i = 1$. Therefore, if $\rho \in \{1, n+2\}$ then $\mathcal{F}(\boldsymbol{X}, \rho)_1 \oplus \mathcal{F}(\boldsymbol{X}, \rho)_2 = 1$. But by the function definition, if $\rho \in \{2n+1, 2n+2\}$ then $\mathcal{F}(\boldsymbol{X}, \rho)_1 \oplus \mathcal{F}(\boldsymbol{X}, \rho)_2 = 1$. Therefore, for $b \in \{0, 1\}$:

$$\Pr_{\rho \xleftarrow{R} [2n+2]} [\mathcal{F}(\boldsymbol{X}, \rho)_1 \oplus \mathcal{F}(\boldsymbol{X}, \rho)_2 = b] \leq 1 - \frac{1}{n+1}$$

Which means that:

$$\left| \Pr_{\rho \xleftarrow{R} [2n+2]} [\mathcal{F}(\boldsymbol{X}, \rho)_1 \oplus \mathcal{F}(\boldsymbol{X}, \rho)_2 = 1] - \Pr_{\rho \xleftarrow{R} [2n+2]} [\mathcal{F}(\boldsymbol{X}, \rho)_1 \oplus \mathcal{F}(\boldsymbol{X}, \rho)_2 = 0] \right| \leq 1 - \frac{2}{n+1}$$

□

Note 4.1. *Claim 4.2 together with these last results imply that replacing the gate in the construction of [Val84] with the* Universal *gate from the previous section results in an almost k-wise independent hash function, where the parameters k and ϵ are determined by the number of samples taken from the input.*

We now prove claim 4.4. In fact, we prove a stronger claim that immediately implies it.

Claim 4.5. *If a distribution $\mathcal{D}$ over $\{-1, 1\}^k$ is symmetric, then it is also 0-biased with respect to all linear tests of odd size.*

Proof. Let $S \in \{-1, 1\}^k$ s.t. $\text{wt}(S)$ is odd; let $A = \{\boldsymbol{y} \in \text{Supp}(\mathcal{D}) \mid \chi_S(\boldsymbol{y} = 1)\}$ and $\bar{A} = \text{Supp}(\mathcal{D}) \setminus A$. Then:

$$\begin{aligned}
\Pr_{\boldsymbol{Y} \sim \mathcal{D}} [\chi_S(\boldsymbol{Y}) = 1] &= \sum_{\boldsymbol{y} \in A} \Pr_{\boldsymbol{Y} \sim \mathcal{D}} [\boldsymbol{Y} = \boldsymbol{y}] \\
&= \sum_{\boldsymbol{y} \in A} \Pr_{\boldsymbol{Y} \sim \mathcal{D}} [\boldsymbol{Y} = \boldsymbol{y}^{\boldsymbol{c}}] \\
&= \sum_{\boldsymbol{y} \in \bar{A}} \Pr_{\boldsymbol{Y} \sim \mathcal{D}} [\boldsymbol{Y} = \boldsymbol{y}] \\
&= \Pr_{\boldsymbol{Y} \sim \mathcal{D}} [\chi_S(\boldsymbol{Y}) = -1]
\end{aligned}$$

Above, we used symmetry in the second equality and the fact that $|S|$ is odd in the third.

□

5 Reducing Key Size

In this section we are interested in reducing the key size of our generic constructions. In particular, we would like it to be linearly, instead of polynomially, dependent on k. To this end, we generalize our results to arbitrary input and output sizes.

We will therefore start using the convention that hash function families are of the form $\mathcal{F} : D \times \{0,1\}^r \to \mathbb{Z}_q$ for arbitrary domain D and $q \in \mathbb{N}$. Note that although the outputs of hash functions are sometimes not in any group $\mathbb{Z}_q$, they can always be embedded in such a group, so long as we do not care how our transformations affect their original domain. Therefore, this definition is without loss of generality.

It will later turn out to be the case that larger outputs allow us to rapidly increase the independence parameter k without using more samples from the distribution, which will be very useful. This is because there are many more functions over larger alphabets than over small ones. This property will enable us to have more fine-grained control over the rate at which the independence parameter increases. Details shortly follow.

In order to claim anything meaningful about functions of this sort we need to generalize the notion of bias to functions over groups $\mathbb{Z}_q$, where $\mathbb{Z}_q = \{0, 1, \ldots, q-1\}$ with addition modulo q as the group operation. We also need to generalize the Fourier decomposition of functions to work on ones that have outputs in $\mathbb{Z}_q$.

5.1 A Short Primer On Fourier Analysis of Functions Over $\mathbb{Z}_q$

We roughly follow the exposition given in [Bab89].

Let $\mathbb{Z}_q$ be the group $\{0, 1, \ldots, q-1\}$ with addition modulo q as the group operation, and let $G = \mathbb{Z}_q^k$ be the vector space of dimension k over $\mathbb{Z}_q$. Note that G with vector summation modulo q is also a group.

We denote by $\mathbb{C}$ the field of complex numbers. We use $\mathbb{T}$ to denote the group of complex numbers of unit length with multiplication over $\mathbb{C}$ as the group operation.

Definition 5.1. *The* characters *of a group G are all homomorphisms $\chi : G \to \mathbb{T}$. The set of characters of G is denoted by $\hat{G}$.*

Fact 5.1. *The set of characters of G forms a group under the following definition of group product:*

$$\chi_1 * \chi_2(x) \stackrel{\Delta}{=} \chi_1(x) \cdot \chi_2(x)$$

In which $\cdot$ is the usual multiplication over the complex field.

It turns out that $\hat{G}$ is isomorphic to G. Specifically, with every element $\boldsymbol{s} \in G = \mathbb{Z}_q^k$ we identify the following element $\chi_{\boldsymbol{s}} : G \to \mathbb{T}$:

$$\chi_{\boldsymbol{s}}(\boldsymbol{x}) \stackrel{\Delta}{=} e^{\frac{2\pi i}{k}\langle \boldsymbol{s}, \boldsymbol{x} \rangle}$$

where $i = \sqrt{-1}$. One can observe that χ_S is the exponentiation of a linear test on $\boldsymbol{x}$ that is defined by $\boldsymbol{s}$.

The set of functions $f : G \to \mathbb{C}$ forms a $|G| = \left|\mathbb{Z}_q^k\right| = q^k$-dimensional vector space $\mathbb{C}^G$.

Definition 5.2. *The* inner product *of two functions $f_1, f_2 : G \to \mathbb{C}$ is defined as:*

$$\langle f_1, f_2 \rangle = \frac{1}{|G|} \sum_{\boldsymbol{s} \in G} f_1(\boldsymbol{s}) \cdot f_2(\boldsymbol{s}) = q^{-k} \sum_{\boldsymbol{s} \in G} f_1(\boldsymbol{s}) \cdot f_2(\boldsymbol{s})$$

Fact 5.2. *The set of characters $\{\chi_{\boldsymbol{s}} \mid \boldsymbol{s} \in G\}$ of a group G forms an orthonormal basis of $\mathbb{C}^G$. That is, for all pairs $\boldsymbol{s_1}, \boldsymbol{s_2} \in G$:*

$$\langle \chi_{\boldsymbol{s_1}}, \chi_{\boldsymbol{s_2}} \rangle = \begin{cases} 1 & \boldsymbol{s_1} = \boldsymbol{s_2} \\ 0 & \boldsymbol{s_1} \neq \boldsymbol{s_2} \end{cases}$$

Any function $f : G \to \mathbb{C}$ can be decomposed in the following way:

$$f = \sum_{\boldsymbol{s} \in G} \hat{f}(\boldsymbol{s}) \cdot \chi_{\boldsymbol{s}}$$

That is, f can be written as a linear combination of its characters.

The function $\hat{f}$ is called the Fourier decomposition of f, and its values are given by:

$$\hat{f}(\boldsymbol{s}) = \langle f, \chi_{\boldsymbol{s}} \rangle$$

Note that a distribution $\mathcal{D}$ over G is itself a function $\mathcal{D} : G \to \mathbb{C}$, and can be decomposed in the same way. We next generalize the notion of bias as defined previously for the binary case.

Definition 5.3. *The* bias *of a distribution $\mathcal{D}$ with respect to a linear test $\boldsymbol{s} \in G$ is the quantity:*

$$\text{bias}_{\mathcal{D}}(\boldsymbol{s}) = \left| \mathop{\mathbb{E}}_{\boldsymbol{X} \sim \mathcal{D}} [\chi_{\boldsymbol{s}}(\boldsymbol{X})] \right|$$

Note 5.1. *The definition given here is not the only possible one. Other, stronger generalizations of bias exist (see for example in [AMN98]). However, in our context the notion defined here fits best, and in most cases replaces the special case seamlessly.*

Definition 5.4. *The* support *of a vector $\boldsymbol{s} \in \mathbb{Z}_q^k$ is defined to be the set of components of $\boldsymbol{s}$ that have non-zero value and is denoted by* $\text{Supp}(\boldsymbol{s})$.

Definition 5.5. *The* Hamming weight *of a vector $\boldsymbol{s} \in \mathbb{Z}_q^k$ is the number of non-zero components of $\boldsymbol{s}$ and is denoted by* $\text{wt}(\boldsymbol{s}) = |\text{Supp}(\boldsymbol{s})|$.

Note that the weight of a vector $\boldsymbol{s}$ defines the number of components of $\boldsymbol{x} \in G = \mathbb{Z}_q^k$ which the linear test $\chi_{\boldsymbol{s}}(\boldsymbol{x})$ depends on.

Definition 5.6. *A distribution $\mathcal{D}$ over $\mathbb{Z}_q^l$ is said to be (t, ϵ)-biased if it is ϵ-biased w.r.t. any linear test $\chi_{\boldsymbol{s}}$ with $0 < \text{wt}(\boldsymbol{s}) \leq t$.*

As in the binary case, we say a distribution over $\mathbb{Z}_q^k$ is ϵ-biased if it is (k, ϵ)-biased.

We now proceed to state the general version of the bias-to-independence bound which will be used heavily in the following sections.

Lemma 5.3. *Let $\mathcal{F} : D \to \mathbb{Z}_q$ be a (k, ϵ)-biased hash function family for some domain D Then $\mathcal{F}$ is $(k, q^{k/2}\epsilon)$-wise independent.*

We give the proof of this lemma in appendix A.

5.2 Generalized Combination Lemmas

5.2.1 Reducing Bias

We next present a general framework for enhancing the bounded independence of distributions over general alphabets $\mathbb{Z}_q^k$. We begin with generalizing lemma 3.2. For this purpose we denote by ADD the operation of addition modulo q defined both over $\mathbb{Z}_q$ and over $\mathbb{Z}_q^k$.

Lemma 5.4. *Let $\mathcal{D}_1, \mathcal{D}_2$ be probability distributions over $\mathbb{Z}_q^k$ and let $\chi_{\boldsymbol{s}}$ be a linear test defined over the same domain. Then if $\mathcal{D}_i$ is ϵ_i-biased w.r.t $\chi_{\boldsymbol{s}}$ for $i \in \{1,2\}$ then $\mathcal{D} = \text{ADD}(\mathcal{D}_1, \mathcal{D}_2)$ is $\epsilon_1 \cdot \epsilon_2$-biased w.r.t. $\chi_{\boldsymbol{s}}$.*

Proof. The proof is almost identical to the one in the binary case.

$$\begin{aligned}
\text{bias}_{\mathcal{D}}(\boldsymbol{s}) &= \left| \mathop{\mathbb{E}}_{\boldsymbol{X}\sim\mathcal{D}} [\chi_{\boldsymbol{s}}(\boldsymbol{X})] \right| \\
&= \left| \mathop{\mathbb{E}}_{\boldsymbol{X_1}\sim\mathcal{D}_1, \boldsymbol{X_2}\sim\mathcal{D}_2} [\chi_{\boldsymbol{s}}(\boldsymbol{X_1} + \boldsymbol{X_2})] \right| \\
&= \left| \mathop{\mathbb{E}}_{\boldsymbol{X_1}\sim\mathcal{D}_1, \boldsymbol{X_2}\sim\mathcal{D}_2} [\chi_{\boldsymbol{s}}(\boldsymbol{X_1}) \cdot \chi_{\boldsymbol{s}}(\boldsymbol{X_2})] \right| \\
&= \left| \mathop{\mathbb{E}}_{\boldsymbol{X_1}\sim\mathcal{D}_1} [\chi_{\boldsymbol{s}}(\boldsymbol{X_1})] \cdot \mathop{\mathbb{E}}_{\boldsymbol{X_2}\sim\mathcal{D}_2} [\chi_{\boldsymbol{s}}(\boldsymbol{X_2})] \right| \\
&= \left| \mathop{\mathbb{E}}_{\boldsymbol{X_1}\sim\mathcal{D}_1} [\chi_{\boldsymbol{s}}(\boldsymbol{X_1})] \right| \cdot \left| \mathop{\mathbb{E}}_{\boldsymbol{X_2}\sim\mathcal{D}_2} [\chi_{\boldsymbol{s}}(\boldsymbol{X_2})] \right| \\
&\leq \epsilon_1 \cdot \epsilon_2
\end{aligned}$$

Where the third equality is by linearity of $\chi_{\boldsymbol{s}}$ and the fourth by independence of $\boldsymbol{X_1}$ and $\boldsymbol{X_2}$. The final inequality is due to our assumption that $\mathcal{D}_i$ is ϵ_i-biased w.r.t. $\chi_{\boldsymbol{s}}$. □

We get from this lemma a corollary which is "morally" equivalent to corollary 3.2.1.

Corollary 5.4.1. *If $\mathcal{D}$ is ϵ biased w.r.t. $\chi_{\boldsymbol{s}}$ then:*

1. $\text{ADD}(\mathcal{D}, \mathcal{D})$ *is ϵ^2-biased w.r.t. $\chi_{\boldsymbol{s}}$.*
2. *For any distribution $\mathcal{D}'$ defined over the same domain:* $\text{ADD}(\mathcal{D}, \mathcal{D}')$ *is ϵ-biased w.r.t. $\chi_{\boldsymbol{s}}$.*

The proof of this corollary is almost identical to that of corollary 3.2.1.

5.2.2 Increasing The Independence Parameter

With the bias-reduction function defined, we proceed to increasing the independence parameter k. This is done in two stages. In the first stage we show that an asymmetric function which receives

two inputs from very different domains can be used to almost square the value of k when given inputs from suitable distributions. In the second stage we show how to generate these distributions from two samples from a (k, ϵ)-biased hash family.

Definition 5.7. *We define the function XIST with* $\mathrm{XIST} : G \times \{0,1\} \to G$ *for any group* G *as:*

$$\mathrm{XIST}(x, b) = \begin{cases} x & b = 1 \\ e & b = 0 \end{cases}$$

Where e *is the neutral element of* G*. Specifically, when* $G = \mathbb{Z}_q^k$*:* $e = \mathbf{0^k}$*.*

Note 5.2. *The name XIST (read: transist) was chosen because this function behaves very much like a transistor.*

The intuition behind this definition is that whenever $b_i = 0$: a linear test on $\mathrm{XIST}(\boldsymbol{x}, \boldsymbol{b})$ will not depend on x_i, which is very similar to the effect of the AND gate from lemma 3.4.

We now define the properties we require a distribution over $\{0,1\}^k$ to have, in order to be useful when applying XIST.

Definition 5.8. *Let* $\mathcal{B}^p$ *be any pairwise-independent probability distribution over* $\{0,1\}^k$ *for arbitrary* k *s.t. each bit* $\mathcal{B}^p$ *is a Bernoulli random variable with expectation* p*. We say a probability distribution* $\mathcal{B}$ *is an* (ϵ_1, ϵ_2)-good *approximation of* $\mathcal{B}^p$ *if it has the two following properties:*

1. *For all* $i \in [k]$*:* $\mathcal{B}_i$ *is* ϵ_1*-close in* L_1 *norm to* $\mathcal{B}_i^p$*.*
2. *For all pairs* $i \neq j \in [k]$*: the pair* $(\mathcal{B}_i, \mathcal{B}_j)$ *is* ϵ_2*-close in* L_1 *norm to* $(\mathcal{B}_i^p, \mathcal{B}_j^p)$*.*

Lemma 5.5. *Let* $\mathcal{D}$ *be a* (t, ϵ)*-biased distribution over* $\mathbb{Z}_q^k$*. Furthermore, let* $\mathcal{B}$ *be an* (ϵ_1, ϵ_2)*-good approximation of* $\mathcal{B}^p$ *defined over* $\{0,1\}^k$*, and assume* $p > \epsilon_1$*. Then* $\mathrm{XIST}(\mathcal{D}, \mathcal{B})$ *is* $\nu + \epsilon(1-\nu)$*-biased w.r.t. all linear tests* $\chi_{\boldsymbol{s}}$ *of weight* $1 \leq \mathrm{wt}(\boldsymbol{s}) \leq \frac{t}{2(p+\epsilon_1)}$*, for:*

$$\nu = \frac{\mathrm{wt}(\boldsymbol{s})\,\epsilon_2 + \epsilon_1 + p}{\mathrm{wt}(\boldsymbol{s}) \cdot (p - \epsilon_1)^2}$$

Note 5.3. *We only require* $\mathcal{B}$ *to have small* pairwise *correlations, while* $\mathcal{D}$ *has small bias in any* t*-tuple of its components. This enables us to achieve relatively high efficiency in our lemma, above.*

Proof. Let $\boldsymbol{s} \in \mathbb{Z}_q^k$ s.t. $t < \mathrm{wt}(\boldsymbol{s}) \leq \frac{1}{8}t^2$.

For all $\boldsymbol{b} \in \{0,1\}^k$ we denote by $\psi_{\boldsymbol{s},\boldsymbol{b}} = \mathrm{Supp}(\boldsymbol{s}) \cap \mathrm{Supp}(\boldsymbol{b}) \in [k]$ the mutual support of $\boldsymbol{s}$ and $\boldsymbol{b}$. Then, by definitions:

$$\begin{aligned} \chi_{\boldsymbol{s}}(\mathrm{XIST}(\boldsymbol{x}, \boldsymbol{b})) &= e^{\frac{2\pi i}{q} \sum_{j \in [k]} s_j \cdot \mathrm{XIST}(x_j, b_j)} \\ &= e^{\frac{2\pi i}{q} \sum_{j \in \mathrm{Supp}(\boldsymbol{s})} s_j \cdot \mathrm{XIST}(x_j, b_j)} \\ &= e^{\frac{2\pi i}{q} \sum_{j \in \psi_{\boldsymbol{s},\boldsymbol{b}}} s_j \cdot x_j} \end{aligned}$$

Thus, if $1 \leq |\psi_{\boldsymbol{s},\boldsymbol{b}}| \leq t$, then: $\chi_{\boldsymbol{s}}(\text{XIST}(\boldsymbol{x},\boldsymbol{b}))$ is the exponent of a linear test of size between 1 and t on $\boldsymbol{x}$, and is therefore at most ϵ-biased.

Next we begin with the same analysis technique used in the proof of lemma 3.4.

$$\begin{aligned}
\text{bias}_{\text{XIST}(\mathcal{D},\mathcal{B})}(\boldsymbol{s}) &= \left| \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{D},\boldsymbol{b}\sim\mathcal{B}} \left[\chi_{\boldsymbol{s}}(\text{XIST}(\boldsymbol{x},\boldsymbol{b}))\right] \right| \\
&\leq \left| \Pr_{\boldsymbol{x}\sim\mathcal{D},\boldsymbol{b}\sim\mathcal{B}} \left[|\psi_{\boldsymbol{s},\boldsymbol{b}}| \in [t]\right] \cdot \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{D},\boldsymbol{b}\sim\mathcal{B}} \left[\chi_{\boldsymbol{s}}(\text{XIST}(\boldsymbol{x},\boldsymbol{b}))\middle| |\psi_{\boldsymbol{s},\boldsymbol{b}}| \in [t]\right] \right| \\
&\quad + \left| \Pr_{\boldsymbol{x}\sim\mathcal{D},\boldsymbol{b}\sim\mathcal{B}} \left[|\psi_{\boldsymbol{s},\boldsymbol{b}}| \notin [t]\right] \cdot \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{D},\boldsymbol{b}\sim\mathcal{B}} \left[\chi_{\boldsymbol{s}}(\text{XIST}(\boldsymbol{x},\boldsymbol{b}))\middle| |\psi_{\boldsymbol{s},\boldsymbol{b}}| \notin [t]\right] \right| \\
&\leq \Pr_{\boldsymbol{b}\sim\mathcal{B}} \left[|\psi_{\boldsymbol{s},\boldsymbol{b}}| \notin [t]\right] + \epsilon \cdot \Pr_{\boldsymbol{b}\sim\mathcal{B}} \left[|\psi_{\boldsymbol{s},\boldsymbol{b}}| \in [t]\right]
\end{aligned}$$

Using the following lemma concludes the proof:

Lemma 5.6. *$\psi_{\boldsymbol{s},\mathcal{B}}$ admits the following property:*

$$\Pr_{\boldsymbol{b}\sim\mathcal{B}} \left[|\psi_{\boldsymbol{s},\boldsymbol{b}}| \notin [t]\right] \leq \frac{\text{wt}(\boldsymbol{s})\,\epsilon_2 + \epsilon_1 + p}{\text{wt}(\boldsymbol{s}) \cdot (p - \epsilon_1)^2}$$

□

To prove lemma 5.6, we use a Chebyshev's inequality in order to bound the probability that $|\psi_{\boldsymbol{s},\boldsymbol{b}}| \notin [t]$. We are guided by the following intuition: we require the Chebyshev bound to bound the probability that $|\psi_{\boldsymbol{s},\boldsymbol{x}}| = 0$. Stated in terms of distance from the expectation, we need a bound on the probability that ψ is $\mathbb{E}[\psi]$-far from $\mathbb{E}[\psi]$.

However, the Chebyshev bound is symmetric, and therefore will bound the probability that $0 < \psi < 2\mathbb{E}[\psi]$. This requires us to use $t \geq 2\mathbb{E}[\psi]$. In order to use Chebyshev's inequality, we need to establish bounds on both the bias and variance of $|\psi_{\boldsymbol{s},\boldsymbol{b}}|$.

Claim 5.7. *Let $\mathcal{D}$, $\mathcal{B}$ and $\boldsymbol{s}$ be as defined in lemma 5.5. Then:*

$$\mathop{\mathbb{E}}_{\boldsymbol{b}\sim\mathcal{B}} \left[|\psi_{\boldsymbol{s},\boldsymbol{b}}|\right] \in \text{wt}(\boldsymbol{s}) \cdot (p \pm \epsilon_1)$$

Proof.

$$\begin{aligned}
\mathop{\mathbb{E}}_{\boldsymbol{b}\sim\mathcal{B}}\left[|\psi_{\boldsymbol{s},\boldsymbol{b}}|\right] &= \mathop{\mathbb{E}}_{\boldsymbol{b}\sim\mathcal{B}}\left[\sum_{i\in\text{Supp}(\boldsymbol{s})} b_i\right] \\
&= \sum_{i\in\text{Supp}(\boldsymbol{s})} \mathop{\mathbb{E}}_{\boldsymbol{b}\sim\mathcal{B}}\left[b_i\right] \\
&\in \sum_{i\in\text{Supp}(\boldsymbol{s})} \left(\mathop{\mathbb{E}}_{b\sim\mathcal{B}_i^p}\left[b\right] \pm \epsilon_1\right) \\
&= \sum_{i\in\text{Supp}(\boldsymbol{s})} \left(p \pm \epsilon_1\right) \\
&= \text{wt}\left(\boldsymbol{s}\right)\cdot\left(p \pm \epsilon_1\right)
\end{aligned}$$

Where the first equality is by definition and the second by linearity of expectation. The third row inclusion is by assumption that $\mathcal{B}_i$ is ϵ_1-close to $\mathcal{P}$. The fourth row equality is by the formula for the expectation of a Bernoulli random variable.

Note that all summations in this proof are done over $\mathbb{R}$ and not over some cyclic group. □

Claim 5.8. *Let $\mathcal{D}$, $\mathcal{B}$ and $\boldsymbol{s}$ as defined in lemma 5.5. Then:*

$$\mathop{\text{Var}}_{\boldsymbol{b}\sim\mathcal{B}}\left[|\psi_{\boldsymbol{s},\boldsymbol{b}}|\right] \leq \text{wt}\left(\boldsymbol{s}\right)\cdot\left(\text{wt}\left(\boldsymbol{s}\right)\epsilon_2 + \epsilon_1 + p\right)$$

Proof. Let $\mathcal{P}_{i,j} = (\mathcal{B}_i^p, \mathcal{B}_j^p)$. Then:

$$\begin{aligned}
\mathop{\text{Var}}_{\boldsymbol{b}\sim\mathcal{B}}\left[|\psi_{\boldsymbol{s},\boldsymbol{b}}|\right] &= \mathop{\text{Var}}_{\boldsymbol{b}\sim\mathcal{B}}\left[\sum_{i\in\text{Supp}(\boldsymbol{s})} b_i\right] \\
&= \sum_{i\in\text{Supp}(\boldsymbol{s})} \mathop{\text{Var}}_{\boldsymbol{b}\sim\mathcal{B}}\left[b_i\right] + \sum_{i\neq j} \mathop{\text{Cov}}_{\boldsymbol{b}\sim\mathcal{B}}\left[b_i, b_j\right] \\
&\leq \sum_{i\in\text{Supp}(\boldsymbol{s})} \left(\mathop{\text{Var}}_{b\sim\mathcal{B}_i^p}\left[b\right] + \epsilon_1\right) + \sum_{i\neq j}\left(\mathop{\text{Cov}}_{\boldsymbol{b}\sim\mathcal{P}_{i,j}}\left[b_1, b_2\right] + \epsilon_2\right) \\
&= \sum_{i\in\text{Supp}(\boldsymbol{s})} \left(p\left(1-p\right) + \epsilon_1\right) + \sum_{i\neq j}\epsilon_2 \\
&\leq \text{wt}\left(\boldsymbol{s}\right)\cdot\left(\text{wt}\left(\boldsymbol{s}\right)\epsilon_2 + \epsilon_1 + p - p^2\right) \\
&\leq \text{wt}\left(\boldsymbol{s}\right)\cdot\left(\text{wt}\left(\boldsymbol{s}\right)\epsilon_2 + \epsilon_1 + p\right)
\end{aligned}$$

Where the first equality is by definition and the second by the well-known formula for the variance of a sum of random variables. The third row inequality is by assumptions on the distribution $\mathcal{B}$. The fourth row equality is derived from the formula for the variance of a Bernoulli random variable and from the fact that the covariance of independent variables is 0. □

We now prove the lemma.

Proof of lemma 5.6. Let $\psi = \psi_{\boldsymbol{s},\boldsymbol{b}}$. Then:

$$\begin{aligned}
\Pr_{\boldsymbol{b}\sim\mathcal{B}}\left[\left||\psi| - \mathop{\mathbb{E}}_{\boldsymbol{b}\sim\mathcal{B}}\left[|\psi|\right]\right| \geq \mathop{\mathbb{E}}_{\boldsymbol{b}\sim\mathcal{B}}\left[|\psi|\right]\right] &= \Pr_{\boldsymbol{b}\sim\mathcal{B}}\left[\left||\psi| - \mathop{\mathbb{E}}_{\boldsymbol{b}\sim\mathcal{B}}\left[|\psi|\right]\right| \geq \frac{\mathop{\mathbb{E}}_{\boldsymbol{b}\sim\mathcal{B}}\left[|\psi|\right]}{\sqrt{\mathop{\mathrm{Var}}_{\boldsymbol{b}\sim\mathcal{B}}\left[|\psi|\right]}} \cdot \sqrt{\mathop{\mathrm{Var}}_{\boldsymbol{b}\sim\mathcal{B}}\left[|\psi|\right]}\right] \\
&\leq \frac{\mathop{\mathrm{Var}}_{\boldsymbol{b}\sim\mathcal{B}}\left[|\psi|\right]}{\mathop{\mathbb{E}}_{\boldsymbol{b}\sim\mathcal{B}}\left[|\psi|\right]^2} \\
&\leq \frac{\mathrm{wt}\left(\boldsymbol{s}\right) \cdot \left(\mathrm{wt}\left(\boldsymbol{s}\right)\epsilon_2 + \epsilon_1 + p\right)}{\mathrm{wt}\left(\boldsymbol{s}\right)^2 \cdot (p-\epsilon_1)^2} \\
&= \frac{\mathrm{wt}\left(\boldsymbol{s}\right)\epsilon_2 + \epsilon_1 + p}{\mathrm{wt}\left(\boldsymbol{s}\right) \cdot (p-\epsilon_1)^2}
\end{aligned}$$

Where the first equality is obtained simply by multiplying and dividing by the standard deviation of $|\psi|$. The second row inequality is by Chebyshev's inequality. The third row inequality is due to claims 5.7 and 5.8 and the assumption that $p > \epsilon_1$.

Using claim 5.7 and the fact that ψ is integer-valued we obtain the following:

$$\Pr_{\boldsymbol{b}\sim\mathcal{B}}\left[|\psi| \in \left[2\mathrm{wt}\left(\boldsymbol{s}\right) \cdot (p+\epsilon_1)\right]\right] \leq \frac{\mathrm{wt}\left(\boldsymbol{s}\right)\epsilon_2 + \epsilon_1 + p}{\mathrm{wt}\left(\boldsymbol{s}\right) \cdot (p-\epsilon_1)^2}$$

Recalling that $\mathrm{wt}\left(\boldsymbol{s}\right) \leq \frac{t}{2(p+\epsilon_1)}$ we obtain $2\mathrm{wt}\left(\boldsymbol{s}\right) \cdot (p+\epsilon_1) \leq t$. This concludes the proof.

□

Lemma 5.5 was presented in a rather general way. At this point it will prove instructive to specify some of the lemma parameters more explicitly and derive a statement we can use in our constructions.

5.2.3 Extracting Good Bits from a Hash Output

We next show how to extract a good approximation of $\mathcal{B}^p$ from a (t,ϵ)-biased distribution $\mathcal{D}$ over $\mathbb{Z}_q^k$. To this end we define the following filter function $\mathrm{FILT}_t : \mathbb{Z}_q \to \{0,1\}$.

Definition 5.9.

$$\mathrm{FILT}_\tau(x) = \begin{cases} 1 & x < \tau \\ 0 & else \end{cases}$$

Note 5.4. *The FILT function is basically a low-pass filter on x with threshold τ.*

Lemma 5.9. *Let $\mathcal{D}$ be a (t,ϵ)-biased distribution over $\mathbb{Z}_q^k$ for $t \geq 2$. Let $\tau > 0$ Finally, let $p = (\tau+1)/q$. Then:* $\mathrm{FILT}_\tau(\mathcal{D})$ *is a $(\sqrt{q}\cdot\epsilon, q\cdot\epsilon)$-good approximation of $\mathcal{B}^p$.*

Proof. Since $\mathcal{D}$ is $(2, \epsilon)$-biased, then by lemma 5.3:

1. Each component of $\mathcal{D}$ is $\sqrt{q} \cdot \epsilon$-close to uniform
2. Each pair of components is $q \cdot \epsilon$-close to uniform.

The claim follows from the definitions of FILT and p. □

5.2.4 Increasing Bounded Independence

We now have all of the ingredients required to complete our goal of increasing the independence parameter. For simplicity we define a combined function that achieves the required purpose.

Definition 5.10. *Define the BOOST function as* $\text{BOOST}_\tau : \mathbb{Z}_q \times \mathbb{Z}_q \times \mathbb{Z}_q \to \mathbb{Z}_q$ *as:*

$$\text{BOOST}_\tau(x, y, z) = \text{ADD}(x, \text{XIST}(y, \text{FILT}_\tau(z)))$$

Lemma 5.10. *Let $\mathcal{D}$ be a (t, ϵ)-biased probability distribution over $\mathbb{Z}_q^k$ with $t \geq 2$. Let $\tau > 0$ and define $p = (\tau + 1)/q$. Assume that $\epsilon_1 = \sqrt{q} \cdot \epsilon > p$. Then,* $\text{BOOST}_\tau(\mathcal{D}, \mathcal{D}, \mathcal{D})$ *is $(t', \nu + \epsilon(1-\nu))$-biased with:*

$$t' = \frac{t}{2\left(p + \sqrt{q} \cdot \epsilon\right)}$$

$$\nu = \frac{q\epsilon}{p^2 - 2\sqrt{q} \cdot \epsilon p + q\epsilon^2} + \frac{p + \sqrt{q} \cdot \epsilon}{(t+1) \cdot \left(p^2 - 2\sqrt{q} \cdot \epsilon p + q\epsilon^2\right)}$$

Proof. By lemma 5.9: $\text{FILT}_\tau(\mathcal{D})$ is a $(\sqrt{q} \cdot \epsilon, q \cdot \epsilon)$-good approximation of $\mathcal{P}^p$ for $p = (\tau + 1)/q$.

The key idea at this point is to limit the test size to at least $t + 1$. This will allow us to increase the test size significantly while retaining a bounded bias. We then use corollary 5.4.1 to bound the bias of smaller tests.

By lemma 5.5: $\text{XIST}(\mathcal{D}, \text{FILT}_\tau(\mathcal{D}))$ is $\epsilon' = \nu + \epsilon(1 - \nu)$-biased w.r.t. all tests $\chi_{\boldsymbol{s}}$ with weight $t + 1 \leq \text{wt}(\boldsymbol{s}) \leq t'$ for:

$$\begin{aligned}
\nu &= \frac{\text{wt}(\boldsymbol{s})\,\epsilon_2 + \epsilon_1 + p}{\text{wt}(\boldsymbol{s}) \cdot (p - \epsilon_1)^2} \\
&= \frac{\epsilon_2}{(p - \epsilon_1)^2} + \frac{p + \epsilon_1}{\text{wt}(\boldsymbol{s}) \cdot (p - \epsilon_1)^2} \\
&\leq \frac{q\epsilon}{p^2 - 2\sqrt{q} \cdot \epsilon p + q\epsilon^2} + \frac{p + \sqrt{q} \cdot \epsilon}{(t+1) \cdot \left(p^2 - 2\sqrt{q} \cdot \epsilon p + q\epsilon^2\right)}
\end{aligned}$$

And:

$$\begin{aligned}
t' &= \frac{t}{2\left(p + \epsilon_1\right)} \\
&= \frac{t}{2\left(p + \sqrt{q} \cdot \epsilon\right)}
\end{aligned}$$

Applying corollary 5.4.1 and noting that $\nu + \epsilon(1-\nu) \geq \epsilon$ we obtain the lemma. □

We now give explicit assignments to the parameters to obtain a useful version of the lemma.

Corollary 5.10.1. *Let $\mathcal{D}$ be a (t, ϵ)-biased probability distribution over $\mathbb{Z}_q^k$ with $\epsilon = \frac{p^2}{8q}$. Let $\tau = \left\lfloor \frac{2q}{t} \right\rfloor$. Then* $\text{BOOST}_\tau(\mathcal{D}, \mathcal{D}, \mathcal{D})$ *is $(\frac{4qt^2}{18q+9t}, \frac{11}{14} + \frac{3}{14}\epsilon)$-biased.*

Note 5.5. *When $t \ll q$, then $t' \approx \frac{1}{5}t^2$. However, when $t = \Omega(q)$ then $t' \approx \frac{1}{7}qt$. So q is an upper bound on the growth rate of t that is possible to attain using this method.*

Proof. First note that $p = \left(\left\lfloor \frac{2q}{t} \right\rfloor + 1\right)/q$. Therefore: $\frac{2}{t} \leq p \leq \frac{2}{t} + \frac{1}{q}$.

Also notice that $\epsilon \cdot \sqrt{q} = \frac{p^2}{8\sqrt{q}} \leq \frac{p}{8}$.

Using lemma 5.10 we obtain that $\text{BOOST}_\tau(\mathcal{D}, \mathcal{D}, \mathcal{D})$ is $(t', \nu + \epsilon(1-\nu))$-biased for:

$$\begin{aligned}
\nu &\leq \frac{q\epsilon}{p^2 - 2\sqrt{q}\cdot\epsilon p + q\epsilon^2} + \frac{p + \sqrt{q}\cdot\epsilon}{(t+1)\cdot\left(p^2 - 2\sqrt{q}\cdot\epsilon p + q\epsilon^2\right)} \\
&\leq \frac{1}{8}\cdot\frac{p^2}{p^2 - \frac{p^3}{4q} + \frac{p^4}{64q^2}} + \frac{\left(1+\frac{1}{8}\right)p}{(t+1)\cdot\left(p^2 - \frac{p^3}{4q} + \frac{p^4}{64q^2}\right)} \\
&\leq \frac{1}{8}\cdot\frac{p^2}{p^2 - \frac{p^2}{8}} + \frac{9}{8}\cdot\frac{p}{(t+1)\cdot\left(p^2 - \frac{p^2}{8}\right)} \\
&\leq \frac{1}{8}\cdot\frac{8}{7} + \frac{9}{8}\cdot\frac{8}{7}\cdot\frac{1}{p(t+1)} \\
&< \frac{11}{14}
\end{aligned}$$

And:

$$\begin{aligned}
t' &= \frac{t}{2\left(p + \epsilon\sqrt{q}\right)} \\
&\geq \frac{t}{2\cdot\frac{9}{8}p} \\
&\geq \frac{4t}{9\left(\frac{2}{t} + \frac{1}{q}\right)} \\
&= \frac{4qt^2}{18q + 9t}
\end{aligned}$$

□

Note 5.6. *Corollary 5.10.1 is only useful for values of t larger than, say, 8. However, the same basic operation with $\tau \approx \frac{q}{2}$ can be shown to increase the value of t for small values of t. This*

increase, although smaller that the one shown here, is sufficient to allow "escaping" lower values of t into a domain in which our lemmas hold. In order not to make this discussion too lengthy, we omit further details.

We also remark that in [GV15] it is shown that addition over $\mathbb{Z}_q^k$ of a constant number of samples allows one to increase the value of t by 1 without affecting ϵ. This approach can also be used until reaching sufficiently large values of t without affecting the asymptotic efficiency of our construction.

5.3 Bias Reduction via Expander Graphs

Close inspection of the results of section 4 reveals that the main bottleneck of our construction is the final bias-reduction phase. One way to see this is by observing that in all other stages the "cost" in randomness of the construction is proportional to $\frac{k}{k'}$, whereas in the final stage it is proportional to k'. This means that even if one starts with $k = k'/2$ for instance, this stage still requires a lot of randomness.

Therefore, we are interested in increasing the efficiency of this stage. This is done using a common de-randomization technique originating in [AKS87]. Specifically, we replace independent sampling with a random walk on an expander graph.

The notion of expander graphs will be recalled shortly, but first we consider how graphs in general relate to this stage. Let us consider a (k, ϵ)-biased hash function family. This family defines a distribution $\mathcal{D}$ over specific hash functions.

Let H be the support of $\mathcal{D}$, and assume for simplicity that $\mathcal{D}$ is uniform on H. Consider the full graph on H. That is, a graph G whose nodes are the functions $h \in H$ and that has a single edge between every two nodes, including self-loops. Observe that sampling two functions h_1, h_2 independently from $\mathcal{D}$ is equivalent to sampling one function h_1 from $\mathcal{D}$ and then sampling a neighbour of h_1 at random from G. This remains true for larger sets of samples $h_1, \ldots, h_t$. This means that sampling them independently at random from $\mathcal{D}$ is equivalent to sampling the first from $\mathcal{D}$ and then for $2 \leq i \leq t$ sampling h_i as a random neighbour of h_{i-1} in G. This latter process of iteratively sampling a neighbour in G of the previous sample is called a *random walk* on G.

To this point we merely gave some definitions and pointed to the connections between them. Now comes the crucial point. It turns out that if we replace G with a so-called *expander* graph then we can retain some of the useful properties that come from having a full graph. Specifically, we will show that the bias of $\sum_{i=1}^{t} h_i$, where $\{h_i\}_{i\in[t]}$ is a random walk on an expander G over the support H of $\mathcal{D}$, decreases almost as much as it would have if G were a full graph. We next recall the definition of expanders.

Definition 5.11. *We say a graph G is d-*regular *if all of its vertices have exactly d neighbours.*

Definition 5.12. *Let G be a d-regular undirected graph over vertex set V. One way to define G is by its adjacency matrix, which is a $|V| \times |V|$ matrix $\hat{M}$ in which the cell in location $\hat{M}_{i,j}$ contains the number of edges between node i and node j. We define its* normalized adjacency matrix M *as* $M = \frac{1}{d}\hat{M}$.

Note that we allow the graph G to contain parallel edges, and that since G is undirected: M_G is symmetric.

Expander graphs are graphs that have a striking property: *any* subset of vertices that is not too large has a large neighbourhood. This is just one way to characterize expansion, and is called *Vertex Expansion*. We mention this definition here since it seems to us to be more intuitive than others. In [Alo86] it was shown that this notion is roughly equivalent to the following definition, which we use in this text.

Definition 5.13 (Spectral expansion)**.** *Let G be a d-regular graph over n vertices, and let M be its normalized adjacency matrix. By normalized we mean that each entry in the adjacency matrix is divided by d: the degree of the graph. Let $\lambda_1, \lambda_2, \ldots \lambda_n$ be the n eigenvalues of M ordered such that:*

$$1 = |\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_n|$$

Then the spectral expansion *of G is $\lambda(G) = |\lambda_2|$.*

We denote by $\mathbf{1} \in \mathbb{C}^n$ the all-ones vector of cardinality n. It is possible to prove that $\lambda_1 = 1$, with $M\mathbf{1} = \lambda_1\mathbf{1} = \mathbf{1}$. What makes a "good" expander is having $\lambda(G) = |\lambda_2|$ as small as possible, and in any case bounded away from 1. For further information regarding this topic, we refer the reader to [HLW06]. For our purposes, we only require the following lemma:

Lemma 5.11. *Let $\mathcal{D}$ be a (k, ϵ)-biased probability distribution over $\mathbb{Z}_q^l$. Let G be a d-regular graph with spectral expansion parameter λ, whose vertices are labeled by samples of $\mathcal{D}$ such that for all $\boldsymbol{x} \in \mathbb{Z}_q^k$: the number of vertices labeled $\boldsymbol{x}$ is proportional to $\mathcal{D}(\boldsymbol{x})$. For $t \in \mathbb{N}$ let $\mathcal{D}_t$ be the distribution over tuples of t vertices of G that results from a random-walk of length $t-1$ on G.*

Let us define:

$$\mu = \mu(\epsilon, \lambda) = 2 \cdot \max\left(\epsilon + \epsilon\lambda, \lambda + \lambda^2\right)$$

Then $\mathcal{D}_{2t+1}$ is (k, μ^t)-biased.

Using this lemma we obtain the following result.

Corollary 5.11.1. *Let $\mathcal{F} : \{0,1\}^n \times \{0,1\}^r \to \{0,1\}^l$ be a (k, ϵ)-biased family of hash functions for some $\epsilon \leq \frac{1}{5}$. Let G be a d-regular expander graph with expansion parameter $\lambda \leq \frac{1}{5}$, whose vertices are labeled by functions $f \in \mathcal{F}$ such that the number of vertices labeled f is proportional to the probability of drawing f from $\mathcal{F}$.*

Let $\mathcal{D}_{G,t}$ be the distribution over t-tuples of functions from $\mathcal{F}$ which is defined by a random walk of length $t-1$ on G. Define the hash function family $\mathcal{F}_{G,t} : \{0,1\}^n \times \{0,1\}^{r'} \to \{0,1\}^l$ by taking the sum of all functions sampled from $\mathcal{D}_{G,t}$.

Then $\mathcal{F}_{G,2t+1}$ is $(k, 2^{-t})$-biased. Furthermore: $r' = r + 2t \log d$.

Proof. Since $\epsilon, \lambda \leq \frac{1}{5}$, then: $\epsilon + \epsilon\lambda, \lambda + \lambda^2 \leq \frac{6}{12}$. Therefore $\mu(\epsilon, \lambda) \leq \frac{1}{2}$. The bound on the bias then follows from lemma 5.11.

We note further that the randomness required to sample from $\mathcal{D}_{G,2t+1}$ is equal to the randomness required to sample from $\mathcal{D}$ and then to sample a neighbour of the current vertex $2t$ times. Since G is d-regular, sampling a neighbour requires $\log d$ random bits. The corollary follows. □

Corollary 5.11.1 provides us with what one needs to improve upon our generic construction. What remains is to prove lemma 5.11.

Proof of lemma 5.11. Let $\mathcal{D}$ and G be as defined in the lemma statement and assume that the vertex set V of G has cardinality $|V| = n$. Let $\boldsymbol{s} \in \mathbb{Z}_q^l$ be a linear test with $1 \leq \text{wt}(\boldsymbol{s}) \leq k$.

For $i \in \mathbb{Z}_q$ let $V_i \subseteq V = \{\boldsymbol{v} \in V \mid \langle \boldsymbol{s}, \boldsymbol{v} \rangle = i\}$ be the set of vertices which are labeled in such a way that $\chi_{\boldsymbol{s}}$ applied on them, results in i. Note that the sets V_i form a partition of the vertices V.

Let Π_i be the projection matrix on the set V_i. That is, for all $i \in \mathbb{Z}_q$: Π_i is a diagonal matrix with 1 entries in indices $j \in V_i$ and 0 entries everywhere else.

Let $\boldsymbol{w} \in (\mathbb{Z}_q^l)^t$ denote the labels of the vertices in a random walk of length $t-1$ on G, and define $\boldsymbol{W} = \sum_{i \in t} \boldsymbol{w_i} \in \mathbb{Z}_q^l$ as the modulo-q sum of the labels. For all vectors $\boldsymbol{u} \in \mathbb{Z}_q^t$ define $A_{\boldsymbol{u}}$ to be the event in which for all $i \in [t]$: $\boldsymbol{w_i} \in V_{u_i}$. Noting that $\frac{1}{n} \cdot \mathbf{1}$ can be viewed as the uniform distribution over the vertices of V, it is straightforward to verify that the expression

$$\Pi_{u_t} M \Pi_{u_{t-1}} M \dots M \Pi_{u_1} \cdot \frac{1}{n} \mathbf{1}$$

is the probability distribution over vertices given by a random walk of length $t-1$ on G subject to vertex number i in the walk belonging to V_{u_i}. Therefore:

$$\begin{aligned} \Pr[A_{\boldsymbol{u}}] &= \langle \mathbf{1}, \Pi_{u_t} M \Pi_{u_{t-1}} M \dots M \Pi_{u_1} \frac{1}{n} \mathbf{1} \rangle \\ &= \langle \boldsymbol{\zeta}, \Pi_{u_t} M \Pi_{u_{t-1}} M \dots M \Pi_{u_1} \boldsymbol{\zeta} \rangle \end{aligned}$$

Where $\boldsymbol{\zeta} = \frac{1}{\sqrt{n}} \mathbf{1}$. This rescaling of $\frac{1}{n} \cdot \mathbf{1}$ into $\boldsymbol{\zeta}$ is done in order to increase its L_2 norm from $\frac{1}{\sqrt{n}}$ to 1. This simplifies calculations later on.

We now define the following parameter:

$$\begin{aligned} B_t^{\parallel} &= \text{bias}_{\mathcal{D}_t}(\boldsymbol{s}) \\ &= \left| \mathop{\mathbb{E}}_{\boldsymbol{W}} \left[e^{\frac{2\pi i}{q} \langle \boldsymbol{s}, \boldsymbol{W} \rangle} \right] \right| \\ &= \left| \mathop{\mathbb{E}}_{\boldsymbol{W}} \left[e^{\frac{2\pi i}{q} \sum_{j \in [t]} \langle \boldsymbol{s}, \boldsymbol{w_j} \rangle} \right] \right| \\ &= \left| \sum_{\boldsymbol{u} \in \mathbb{Z}_q^t} e^{\frac{2\pi i}{q} \sum_{j \in [t]} u_j} \Pr_{\boldsymbol{W}} [A_{\boldsymbol{u}}] \right| \end{aligned}$$

Let $\mathbf{1}^{\perp} = \{\boldsymbol{v} \in \mathbb{C}^n \mid \boldsymbol{v} \perp \mathbf{1} \wedge \|\boldsymbol{v}\| \leq 1\}$ denote the set of vectors perpendicular to $\mathbf{1}$ with L_2

norm at most 1. We also define the following parameter:

$$B_t^{\perp} = \max_{\boldsymbol{v}\in\mathbf{1}^{\perp}} \left| \sum_{\boldsymbol{u}\in\mathbb{Z}_q^t} e^{\frac{2\pi i}{q}\sum_{j\in[t]} u_j} \langle \boldsymbol{\zeta}, \Pi_{u_t} M \Pi_{u_{t-1}} M \dots M \Pi_{u_1} \boldsymbol{v} \rangle \right|$$

We are interested in finding a recursive bound for $B_t^{\|}$. That is, a bound that relates to $B_{t-1}^{\|}$. Unfortunately, the bound we are able to get depends on $B_{t-1}^{\perp}$, which is the reason we defined this quantity. Therefore, we also need a recursive bound on $B_t^{\perp}$. The following claim, whose proof we delay, furnishes these bounds.

Claim 5.12. *The following two inequalities hold:*

1. $B_t^{\|} \leq \epsilon B_{t-1}^{\|} + \lambda B_{t-1}^{\perp}$
2. $B_{t-1}^{\perp} \leq B_{t-1}^{\|} + \lambda B_{t-1}^{\perp}$

Using this claim we may derive:

$$\begin{aligned} B_t^{\|} &\leq \epsilon B_{t-1}^{\|} + \lambda B_{t-1}^{\perp} \\ &\leq \epsilon^2 B_{t-2}^{\|} + \epsilon\lambda B_{t-2}^{\perp} + \lambda B_{t-2}^{\|} + \lambda^2 B_{t-2}^{\perp} \\ &= \left(\epsilon^2 + \lambda\right) B_{t-2}^{\|} + \left(\epsilon\lambda + \lambda^2\right) B_{t-2}^{\perp} \end{aligned}$$

And also:

$$\begin{aligned} B_t^{\perp} &\leq B_{t-1}^{\|} + \lambda B_{t-1}^{\perp} \\ &\leq \epsilon B_{t-2}^{\|} + \lambda B_{t-2}^{\perp} + \epsilon\lambda B_{t-2}^{\|} + \lambda^2 B_{t-2}^{\perp} \\ &= \left(\epsilon + \epsilon\lambda\right) B_{t-2}^{\|} + \left(\lambda + \lambda^2\right) B_{t-2}^{\perp} \end{aligned}$$

Define $B_t = \max\left(B_t^{\|}, B_t^{\perp}\right)$. Then:

$$\begin{aligned} B_t &\leq \max\left(\left(\epsilon^2 + \lambda\right) B_{t-2} + \left(\epsilon\lambda + \lambda^2\right) B_{t-2}, \left(\epsilon + \epsilon\lambda\right) B_{t-2} + \left(\lambda + \lambda^2\right) B_{t-2}\right) \\ &\leq \left(\epsilon + \epsilon\lambda\right) B_{t-2} + \left(\lambda + \lambda^2\right) B_{t-2} \\ &\leq 2 \cdot max\left(\epsilon + \epsilon\lambda, \lambda + \lambda^2\right) B_{t-2} \\ &= \mu \cdot B_{t-2} \end{aligned}$$

Noting that $B_1 \leq 1$ we obtain the following result:

$$B_{2t+1} \leq \mu^t \cdot B_1 \leq \mu^t$$

□

What remains is to prove claim 5.12, which is done next.

Proof of claim 5.12. Note that $\Pi_1\boldsymbol{\zeta} = \frac{|V_{u_1}|}{n}\boldsymbol{\zeta} + \boldsymbol{v}^\perp$ for some $\boldsymbol{v}^\perp \in \mathbf{1}^\perp$. Therefore, by the expansion properties of G:

$$M\Pi_{u_1}\zeta = M\left(\frac{|V_{u_1}|}{n}\boldsymbol{\zeta} + \boldsymbol{v}^\perp\right) = \frac{|V_{u_1}|}{n}\boldsymbol{\zeta} + M\boldsymbol{v}^\perp \tag{2}$$

And $\|M\boldsymbol{v}^\perp\| \leq \lambda$. Therefore:

$$\begin{aligned}
B_t^{\|} &= \left|\sum_{\boldsymbol{u}\in\mathbb{Z}_q^t} e^{\frac{2\pi i}{q}\sum_{j\in[t]}u_j} \Pr_{\boldsymbol{W}}[A_{\boldsymbol{u}}]\right| \\
&= \left|\sum_{\boldsymbol{u}\in\mathbb{Z}_q^t} e^{\frac{2\pi i}{q}\sum_{j\in[t]}u_j} \langle\boldsymbol{\zeta}, \Pi_{u_t}M\Pi_{u_{t-1}}M\ldots M\Pi_{u_1}\boldsymbol{\zeta}\rangle\right| \\
&\leq \left|\sum_{\boldsymbol{u}\in\mathbb{Z}_q^{t-1}} e^{\frac{2\pi i}{q}\sum_{j\in[t]}u_j} \sum_{u_0\in\mathbb{Z}_q} e^{\frac{2\pi i}{q}u_0}\frac{|V_{u_0}|}{n}\langle\boldsymbol{\zeta}, \Pi_{u_{t-1}}M\Pi_{u_{t-2}}M\ldots M\Pi_{u_1}\boldsymbol{\zeta}\rangle\right| \\
&\quad + \left|\sum_{\boldsymbol{u}\in\mathbb{Z}_q^{t-1}} e^{\frac{2\pi i}{q}\sum_{j\in[t]}u_j} \sum_{u_0\in\mathbb{Z}_q} e^{\frac{2\pi i}{q}u_0}\lambda\langle\boldsymbol{\zeta}, \Pi_{u_{t-1}}M\Pi_{u_{t-2}}M\ldots M\Pi_{u_1}\boldsymbol{v}\rangle\right| \\
&= \left|\sum_{\boldsymbol{u}\in\mathbb{Z}_q^{t-1}} e^{\frac{2\pi i}{q}\sum_{j\in[t]}u_j} \langle\boldsymbol{\zeta}, \Pi_{u_{t-1}}M\Pi_{u_{t-2}}M\ldots M\Pi_{u_1}\boldsymbol{\zeta}\rangle \cdot \mathbb{E}\left[e^{\frac{2\pi i}{q}u_0}\right]\right| \\
&\quad + \left|\sum_{\boldsymbol{u}\in\mathbb{Z}_q^{t-1}} e^{\frac{2\pi i}{q}\sum_{j\in[t]}u_j} \langle\boldsymbol{\zeta}, \Pi_{u_{t-1}}M\Pi_{u_{t-2}}M\ldots M\Pi_{u_1}\boldsymbol{v}\rangle \cdot \lambda\sum_{u_0\in\mathbb{Z}_q} e^{\frac{2\pi i}{q}u_0}\right| \\
&\leq \epsilon B_{t-1}^{\|} + \lambda B_{t-1}^{\perp}
\end{aligned}$$

For some $\boldsymbol{v} \in \mathbf{1}^\perp$. The first two equalities are by definitions. The third row inequality is by equation 2, a triangle inequality and splitting the sum into two parts. The fourth row equality is by definition of expectation and independence of the inner products from u_0. The last inequality is by assumption on $\mathcal{D}$ and the fact that the sum over all q-th roots of unity is 1.

This establishes item 1 of the claim. As regards item 2: let $\boldsymbol{v} \in \mathbf{1}^\perp$ be the vector that maximizes $B_t^\perp$ and define:

$$\boldsymbol{z} = \sum_{u_0\in\mathbb{Z}_q} e^{\frac{2\pi i}{q}u_0}\Pi_{u_0}\boldsymbol{v}$$

Note that $P = \sum_{u_0\in\mathbb{Z}_q} e^{\frac{2\pi i}{q}u_0}\Pi_{u_0}$ is an $n \times n$ diagonal matrix over the complex field $\mathbb{C}$, where $P_{i,i} = e^{\frac{2\pi i}{q}u_0}$. Therefore, the eigenvalues of P are all complex roots of unity, which means that this is a unitary matrix. Therefore $\|\boldsymbol{z}\| = \|P\boldsymbol{v}\| = \|\boldsymbol{v}\| \leq 1$.

We can therefore decompose $\boldsymbol{z}$ into $\boldsymbol{z} = \boldsymbol{z}^{\parallel} + \boldsymbol{z}^{\perp}$, where $\boldsymbol{z}^{\perp} \in \mathbf{1}^{\perp}$ and $\boldsymbol{z}^{\parallel} = c \cdot \boldsymbol{\zeta}$ for some $c \leq 1$. Lastly, observe that $\|M(\boldsymbol{z}^{\parallel})\| = \|\boldsymbol{z}^{\parallel}\|$ and $\|M(\boldsymbol{z}^{\perp})\| \leq \lambda \|\boldsymbol{z}^{\perp}\|$. Then, similar to the analysis for $B_t^{\parallel}$:

$$\begin{aligned}
B_t^{\perp} &= \left| \sum_{\boldsymbol{u} \in \mathbb{Z}_q^t} e^{\frac{2\pi i}{q} \sum_{j \in [t]} u_j} \langle \boldsymbol{\zeta}, \Pi_{u_t} M \Pi_{u_{t-1}} M \dots M \Pi_{u_1} \boldsymbol{v} \rangle \right| \\
&= \left| \sum_{\boldsymbol{u} \in \mathbb{Z}_q^{t-1}} e^{\frac{2\pi i}{q} \sum_{j \in [t]} u_j} \langle \boldsymbol{\zeta}, \Pi_{u_{t-1}} M \Pi_{u_{t-2}} M \dots M \Pi_{u_1} M P \boldsymbol{v} \rangle \right| \\
&\leq \left| \sum_{\boldsymbol{u} \in \mathbb{Z}_q^{t-1}} e^{\frac{2\pi i}{q} \sum_{j \in [t]} u_j} \langle \boldsymbol{\zeta}, \Pi_{u_{t-1}} M \Pi_{u_{t-2}} M \dots M \Pi_{u_1} \boldsymbol{\zeta} \rangle \right| \\
&\quad + \lambda \left| \sum_{\boldsymbol{u} \in \mathbb{Z}_q^{t-1}} e^{\frac{2\pi i}{q} \sum_{j \in [t]} u_j} \langle \boldsymbol{\zeta}, \Pi_{u_{t-1}} M \Pi_{u_{t-2}} M \dots M \Pi_{u_1} \boldsymbol{v'} \rangle \right| \\
&\leq B_{t-1}^{\parallel} + \lambda B_{t-1}^{\perp}
\end{aligned}$$

Where the second equality is due to linearity, and $\boldsymbol{v'}$ is some vector in $\mathbf{1}^{\perp}$.

□

6 More General And Efficient Construction

6.1 Putting it all together

We now use the results obtained in this section to improve upon the generic construction of section 4. The improvements will be two-fold. Firstly, we generalize the construction to allow for arbitrary input and output sizes. Secondly, we drastically reduce the amount of randomness needed by the construction.

We assume access to a family of (k_0, ϵ_0)-wise independent hash functions $\mathcal{F}_0 : D \to \mathbb{Z}_q$, and are interested in constructing a family of (k, ϵ)-wise independent hash functions with the same domain and range.

To achieve this, we use the same construction used before, but now with our new improvements applied. Using this method we prove the following theorem:

Theorem 6.1. *Let $\mathcal{F}_0 : D \times \{0,1\}^{r_0} \to \mathbb{Z}_q$ be a (k_0, ϵ_0)-biased hash function family with $k_0 \geq 2$ and constant $\epsilon_0 < 1$. Then for all $k \leq \frac{1}{7} q^2$ and $\epsilon > 0$: it is possible to construct a (k, ϵ)-biased hash function family $\mathcal{F} : D \times \{0,1\}^r \to \mathbb{Z}_q$ using only black-box access to $\mathcal{F}$, with:*

$$r = r_0 \cdot \text{polylog}\,(k) + O(\log \frac{1}{\epsilon})$$

Proof. We follow the same stages previously introduced:

1. Reduce the original bias down to a constant by repeatedly using addition modulo q.
2. Increase the independence parameter to k by repeated, interleaved applications of the BOOST function and addition modulo q.
3. Reduce the final bias down to ϵ using addition modulo q.

Each stage i acts on a distribution over hash functions $\mathcal{D}_{i-1}$ and generates a new one $\mathcal{D}_i$ by repeatedly sampling from $\mathcal{D}_i$ and applying some predetermined function to the samples. The final stage, however, is changed. Instead of sampling directly from $\mathcal{D}_i$: we sample from a random walk of suitable length on an expander whose vertices form the support of $\mathcal{D}_i$.

Note that we cannot use this approach in either of the first two stages, since the bias in these stages is too large. However, if we further assume that ϵ is constant, then the dominating term in the amount of randomness will be the one stemming from the final stage. This means that further improvements to the first two stages will not be very helpful in our current context.

The analysis of the first stage is not changed at all from the one in 4. By corollary 5.4.1: we need $\log\log 8 - \log\log\epsilon_0$ rounds of ADD to reduce the bias down to $\frac{1}{8}$. Since we assume ϵ_0 to be some constant, this adds a constant factor to r_0.

The second stage now uses the BOOST function. By corollary 5.10.1: each BOOST step takes 3 samples from a $(t, \frac{1}{8})$-biased family of functions and outputs a sample from a $(t', \frac{11}{14} + \frac{3}{14} \cdot \frac{1}{8} < 0.82)$-biased function family, such that:

$$t' = \frac{4qt^2}{18q + 9t} \geq \frac{4qt^2}{27q} > \frac{1}{7}t^2$$

Between each two applications of the BOOST function, we must reduce the bias down to $\frac{1}{8}$ once more. This requires 4 ADD steps. The total number of BOOST steps required is $O(\log\log k)$. Therefore this stage adds a $\text{polylog}\,(k)$ factor to the required number of random bits.

For the final stage let G be a d-regular graph with spectral expansion parameter $\lambda \leq \frac{1}{5}$ whose vertices are labeled as desired. Then by corollary 5.11.1 if we take the addition modulo q of the samples from a length-$2\log\frac{1}{\epsilon}$ random walk on G then the result will have bias ϵ.

This stage costs an additional $O(\log\frac{1}{\epsilon'})$ random bits. In total, the construction uses $r = r_0 \cdot O(1) \cdot \text{polylog}\,(k) + O(\log\frac{1}{\epsilon})$ random bits.

□

Corollary 6.1.1. *Let $\mathcal{F} : D \times \{0,1\}^{r_0} \to \mathbb{Z}_q$ be a (k_0, ϵ_0)-wise independent hash function family with $k_0 \geq 2$ and constant $\epsilon_0 < 1$. Then for all $k \leq \frac{1}{7}q^2$ and $\epsilon > 0$: it is possible to construct a (k, ϵ)-wise independent hash function family $\mathcal{F}' : D \times \{0,1\}^{r'} \to \mathbb{Z}_q$ using only black-box access to $\mathcal{F}$, with:*

$$r = r_0 \cdot \text{polylog}\,(k) + O(k\log q + \log\frac{1}{\epsilon})$$

Proof. Setting $\epsilon' = q^{-k/2}\epsilon$: by theorem 6.1 it is possible to construct a (k, ϵ')-biased hash function family using the stated amount of randomness. By lemma 5.3 this is a (k, ϵ)-wise independent hash function family. □

6.2 Back To Circuits

We once again consider our new family of hash functions, and generalize it to allow for larger alphabet outputs. At the same time, let us consider how our new theorems affect this construction.

One way this can be done is by using a (k, ϵ)-biased construction for single-bit outputs and independently sampling from this construction. This yields a construction with the following amount of required random bits:

$$r = n \log n \cdot \text{polylog}\,(k) + O(kl + \log \frac{1}{\epsilon})$$

While this may seem to be good, the quasi-linear dependency on n is bad, since in many cases kl may be much smaller than n. However, we show further that it is possible to achieve a logarithmic dependency on n.

Recall that in section 4.2 we defined a "seed" function that is a random sample from the input bits. We generalize this in a natural way: for any $l \in \mathbb{N}$ we define $\mathcal{F}_l : \{0,1\}^n \times \{0,1\}^{r_0} \to \{0,1\}^l$ as a random element of $\{0,1\}^l$ by sampling l bits from the input independently and uniformly.

Though we defined the output of the hash family as belonging to $\{0,1\}^l$, we here identify this set with $\mathbb{Z}_{2^l}$ instead, for compatibility with our previous definitions and theorems. Note that our definition implies $r_0 = l \log n$ and that the special case of $\mathcal{F}_1$ is the same function family defined previously, in section 4.

As in the former case, we wish to bound the pairwise bias of this function family. Looking back at our analysis from section 4 we notice the reason we could bound the bias of the "seed" family $\mathcal{F}_1$ to begin with was that we applied a deterministic function on the input before sampling from it. In the analysis we took advantage of three properties of the modified input:

1. Symmetry, i.e. relative weight of exactly $\frac{1}{2}$
2. Relative distance of at least $\frac{1}{n+1}$
3. Relative distance not exceeding $1 - \frac{1}{n+1}$

The first property was achieved by basically concatenating the input with its complement. The second property is an intrinsic property of having distinct inputs. The third property is a result of appending constant bits to the input.

The first property allowed us to bound the bias of size-1 tests by 0. The last two properties taken together allowed us to bound the bias of size-2 tests by $1 - \frac{2}{n+1}$. We wish to generalize these properties. To this end, we employ balanced linear codes.

Definition 6.1. *a linear code C is said to be ϵ-balanced if all of its codewords $c \in C$, except the zero codeword, have relative hamming weight $\frac{1}{2} - \epsilon \leq \mathrm{wt}(c) \leq \frac{1}{2} + \epsilon$.*

We adjust our definition of the seed family to the following one.

Definition 6.2. *Let C be a linear code of constant rate R and let* $\mathrm{Enc} : \{0,1\}^n \to \{0,1\}^{n/R}$ *be its encoding function. Define $\hat{\mathcal{F}}_{C,l} : \{0,1\}^n \times \{0,1\}^{r_0} \to \{0,1\}^l$ as the following function family. For all $\boldsymbol{x} \in \{0,1\}^n$, $\boldsymbol{\rho} \in \{0,1\}^{r_0}$ and $i \in [r_0]$:*

$$\hat{\mathcal{F}_{C,l}}(\boldsymbol{x}, \boldsymbol{\rho})_i = \begin{cases} \mathrm{Enc}(\boldsymbol{x})_i & \rho_i \leq \frac{n}{R} \\ 1 - \mathrm{Enc}(\boldsymbol{x})_i & \frac{n}{R} + 1 \leq \rho_i \leq 2\frac{n}{R} \end{cases}$$

Note that $r_0 = \log(2\frac{n}{R}) = O(\log n)$.

We prove the following lemma in appendix B:

Lemma 6.2. *If C is ϵ_C-balanced then $\hat{\mathcal{F}}_{C,l}$ is $(2, 2\epsilon_C)$-biased over Z_{2^l}.*

This lemma implies that if C is an ϵ-biased linear code, for some constant $\epsilon < \frac{1}{2}$, having constant rate $R > 0$ then using $\hat{F}_{C,l}$ as a "seed" hash family and applying the general construction from theorem 6.1, we end up with a (k, ϵ)-wise independent hash function family with key length $r = O(kl + \log\frac{1}{\epsilon}) + \log n \mathrm{polylog}(k)$.

Note 6.1. *The improvement suggested above, i.e. using a balanced code to reduce initial bias, reduces the asymptotic key length in the case when $k = O(n^{1-\delta})$ for any $\delta > 0$. However, this comes at a cost to the simplicity of the implementation by a circuit. This eventually depends on the specific type of code used in a concrete construction.*

6.3 Utilizing Good Seed Functions

In most cases that come to mind, the asymptotic amount of randomness needed by our generic constructions is independent of the randomness required by the original "seed" function. In fact, inspection of corollary 6.1.1 shows that whenever sampling from the original seed family requires $r_0 = O(k^{1-\delta})$ random bits for some $\delta > 0$ then the randomness required to sample from the final family is $r = O(k \log q + \log\frac{1}{\epsilon})$.

This is both a blessing an a curse. On the one hand: it means we do not have to work hard to produce a very good "seed" hash function family. However, it also means that we are not taking full advantage of the properties of a strong hash function when we are given one.

We commented once before that the final bias-reduction step is the most "costly", since its size depends on k rather than $\frac{k}{k_0}$. The reason is that our analysis techniques for BOOST tries to squeeze out the maximum possible improvement to the bounded independence parameter t. This turns out to come at the cost of bumping the bias back up to a constant.

We remark that it is possible to prove that BOOST improves a (t, ϵ)-biased function to a, say, $(1.5t, c \cdot \epsilon)$-biased one for some constant c. This is done using very similar techniques to the ones shown here. Using this method, the bias-reduction stage is folded neatly into the independence-parameter-increasing stage, and provides us with the $\frac{k}{k_0}$ dependency we wanted. This is a more efficient approach when $k_0 \approx k$. However, when one is interested in not-too-small bias and $k \gg k_0$: the approach presented here works best.

7 Reducing Formula Size

In the previous section we reduced the amount of randomness, i.e. key size, required by our construction. This sometimes comes at a cost to the formula size, as the efficiency of the different predicates we use in changing the parameters k and ϵ is affected by the amount of entropy that is streamed into them.

We are now interested in reducing the size of the furmulae that compute our functions, which can thus increase the key size. In this context we do not mind this inefficiency in randomness, and therefore refrain from explicitly stating it in our function family definitions. Instead, we consider our constructions to be *random functions* $f : \{0,1\}^n \to \{0,1\}$ with an implicit extra random input of arbitrary, polynomial in n, size.

The basic construction used is the original one from subsection 4.2, without any of the improvements introduced in section 5. Our goal here is to construct an $(n, 2^{-n})$-biased function with formula size $\tilde{O}(n^2)$.

We recall that the construction consists of three stages:

1. Reducing the initial bias down to a constant.
2. Increasing the independence parameter from 2 to n.
3. Reducing the bias down to 2^{-n}

Our efforts are concentrated now on improving the first two stages. Specifically, we prove the following lemma:

Lemma 7.1. *There exists an $(n, \frac{1}{2})$ random function $f : \{0,1\}^n \to \{0,1\}$ that can be calculated by formulae of size $O(n \log^2 n)$.*

Using this lemma, we obtain the following theorem:

Theorem 7.2. *There exists an $(n, 2^{-n})$ random function $f : \{0,1\}^n \to \{0,1\}$ that can be calculated by formulae of size $O(n^2 \log^2 n)$.*

Proof. Let $f_{final} = \prod_{i \in n} f_i$, where each f_i is an independent copy of f from lemma 7.1. Then by corollary 3.2.1: f_{final} is $(n, 2^{-n})$-biased. Recalling from lemma 7.1 that each copy of f_i has formula

size $O(n \log^2 n)$ and noting that f_{final} consists of combining n copies of f, allow establishing the required bound on the size of f_{final}. □

As a first step to proving lemma 7.1 we prove the following claim.

Claim 7.3. *There exists a* $(2, 0)$ *random function* $f : \{0,1\}^n \to \{0,1\}$ *that can be calculated by formulae of size* $O(n)$.

Proof. For this proof we employ a standard technique for creating pairwise independent distributions. It is presented here in full for completeness. We define the random function $f(\boldsymbol{x}) = \langle \boldsymbol{x}, \boldsymbol{r} \rangle + b$, where $\boldsymbol{r} \in \{0,1\}^n$ is a uniformly random vector and b is a uniformly random bit. All operations are over GF(2), i.e. $+$ denotes XOR and multiplication is AND.

Let $\boldsymbol{x}, \boldsymbol{y} \in \{0,1\}^n$ be to distinct inputs. Then, by independence of b from $\boldsymbol{x}$ and $\boldsymbol{r}$, the bitwise bias of $f(\boldsymbol{x})$ is:

$$\begin{aligned}
\left| \mathop{\mathbb{E}}_{\boldsymbol{r} \xleftarrow{R} \{0,1\}^n, b \xleftarrow{R} \{0,1\}} \left[(-1)^{f(\boldsymbol{x})} \right] \right| &= \left| \mathop{\mathbb{E}}_{\boldsymbol{r} \xleftarrow{R} \{0,1\}^n, b \xleftarrow{R} \{0,1\}} \left[(-1)^{\langle \boldsymbol{x}, \boldsymbol{r} \rangle + b} \right] \right| \\
&\leq \left| \mathop{\mathbb{E}}_{\boldsymbol{r} \xleftarrow{R} \{0,1\}^n} \left[(-1)^{\langle \boldsymbol{x}, \boldsymbol{r} \rangle} \right] \right| \cdot \left| \mathop{\mathbb{E}}_{b \xleftarrow{R} \{0,1\}} \left[(-1)^{b} \right] \right| \\
&= 0
\end{aligned}$$

Now we need to bound the pairwise bias. Since $\boldsymbol{x} \neq \boldsymbol{y}$ there exists an index i s.t. $x_i \neq y_i$. Without loss of generality, let us assume $i = 1$. Fix all bits of $\boldsymbol{r}$ except r_1, and let:

$$z = \sum_{i \neq 1} \left((x_i \cdot r_i) + (y_i \cdot r_i) \right)$$

Noticing that $x_1 + y_1 = 1$, one has:

$$\begin{aligned}
\left| \mathop{\mathbb{E}}_{r_1 \xleftarrow{R} \{0,1\}, b \xleftarrow{R} \{0,1\}} \left[(-1)^{f(\boldsymbol{x}) + f(\boldsymbol{y})} \right] \right| &= \left| \mathop{\mathbb{E}}_{r_1 \xleftarrow{R} \{0,1\}, b \xleftarrow{R} \{0,1\}} \left[(-1)^{\langle \boldsymbol{x}, \boldsymbol{r} \rangle + b + \langle \boldsymbol{y}, \boldsymbol{r} \rangle + b} \right] \right| \\
&= \left| \mathop{\mathbb{E}}_{r_1 \xleftarrow{R} \{0,1\}} \left[(-1)^{z + x_1 \cdot r_1 + y_1 \cdot r_1} \right] \right| \\
&= \left| (-1)^z \mathop{\mathbb{E}}_{r_1 \xleftarrow{R} \{0,1\}} \left[(-1)^{(x_1 + y_1) \cdot r_1} \right] \right| \\
&= \left| (-1)^z \mathop{\mathbb{E}}_{r_1 \xleftarrow{R} \{0,1\}} \left[(-1)^{r_1} \right] \right| \\
&= 0
\end{aligned}$$

This concludes the proof that f is $(2,0)$-biased. We observe that f is the XOR of up to n Boolean variables, each of which is the result of an AND between an input x_i and a random bit r_i. Therefore, the formula size of f is at most $2n = O(n)$. □

We now establish the most important ingredient in our construction: the method by which we bootstrap pairwise independence to almost k-wise independence. In order to do so we require the following strong lemma from [VV86].

Lemma 7.4 (Valiant-Vazirani restated)**.** *Let H be a pairwise-independent hash function family of hash functions of the form $\{0,1\}^n \to \{0,1\}^{l+2}$, and let $S \subseteq \{0,1\}^n$ s.t. $2^l \leq |S| < 2^{l+1}$. For any function $h \in H$, denote by $A_h(S) = \{\boldsymbol{x} \in T \mid h(\boldsymbol{x}) = \mathbf{1}\}$. Then:*

$$\Pr_{h \xleftarrow{R} H}\left[|A_h(S)| = 1\right] \geq \frac{1}{8}$$

In plain words, the probability that there is a unique element in T on which the hash function returns zero is at least $\frac{1}{8}$.

Corollary 7.4.1. *Let $f : \{0,1\}^n \to \{0,1\}$ be a pairwise independent random function, and let $f_1, \ldots, f_{l+2}$ be independent copies of f. Define the function $g : \{0,1\}^n \to \{0,1\}$ as:*

$$g(\boldsymbol{x}) = b \cdot \prod_{i=1}^{l+2} f_i(\boldsymbol{x})$$

Where b is an independent random bit. Finally, let $S \subseteq \{0,1\}^n$ s.t. $2^l \leq |S| < 2^{l+1}$.

Then g is $\frac{7}{8}$-biased w.r.t. the linear test χ_S.

Proof. Denote by $H = (f_1, \ldots, f_{l+2})$ the hash function family that maps every $(x) \in \{0,1\}^n$ to $(f_1(\boldsymbol{x}), \ldots, f_{l+2}(\boldsymbol{x})) \in \{0,1\}^{l+2}$. Then H is a pairwise-independent hash function family with functions of the form $\{0,1\}^n \to \{0,1\}^{l+2}$.

Let A denote the event in which for exactly one input $\boldsymbol{x} \in S$: the product $\prod_{i=1}^{l+2}$ is equal to 1, and let $\bar{A}$ denote the complement of A. Since $2^l \leq |S| < 2^{l+1}$, by lemma 7.4:

$$\begin{aligned}
\Pr[A] &= \Pr\left[\left|\left\{\boldsymbol{x} \in S \mid \prod_{i=1}^{l+2} f_i(\boldsymbol{x}) = 1\right\}\right| = 1\right] \\
&= \Pr\left[\left|\left\{\boldsymbol{x} \in S \mid \forall i \in [l+2] : f_i(\boldsymbol{x}) = 1\right\}\right| = 1\right] \\
&= \Pr_{h \xleftarrow{R} H}\left[|A_h(S)| = 1\right] \\
&\geq \frac{1}{8}
\end{aligned}$$

Let $\boldsymbol{X} \in \{0,1\}^{k \times n}$ be a set of k inputs $\boldsymbol{X^1}, \ldots \boldsymbol{X^k} \in \{0,1\}^n$. Then the bias of $g(\boldsymbol{X})$ w.r.t. χ_S is:

$$\begin{aligned}
\left|\mathbb{E}\left[(-1)^{\sum_{i\in S} g(\boldsymbol{X^i})}\right]\right| &= \left|\mathbb{E}\left[(-1)^{\sum_{i\in S} b\cdot\prod_{j=1}^{l+2} X_j^i}\right]\right| \\
&= \left|\mathbb{E}\left[(-1)^{\sum_{i\in S} b\cdot\prod_{j=1}^{l+2} X_j^i}\middle| A\right]\cdot \Pr[A] + \mathbb{E}\left[(-1)^{\sum_{i\in S} b\cdot\prod_{j=1}^{l+2} X_j^i}\middle|\bar{A}\right]\cdot \Pr\left[\bar{A}\right]\right| \\
&\leq \frac{1}{8}\cdot\mathbb{E}\left[(-1)^b\right] + \frac{7}{8} \\
&= \frac{7}{8}
\end{aligned}$$

In which the inequality stems from the trivial bound on the expectation in the event $\bar{A}$ and the fact that when event A happens: the product $\prod_{i=1}^{l+2} X_j^i$ is equal to 1 for exactly 1 choice of value i and equal to 0 in all other cases. □

We can now prove our lemma.

Proof of lemma 7.1. For all $i \in [\lceil \log n \rceil + 1]$ let $g_i = b_i \cdot \prod_{j=1}^{i} f_{i,j}$, where $f_{i,j}$are independent copies of the pairwise independent random functions from claim 7.3.

Then by corollary 7.4.1: g_i is $\frac{7}{8}$-biased w.r.t. all tests χ_S where $2^{i-2} \leq |S| < 2^{i-1}$.

Define $g = \sum_i g_i$. Then by corollary 3.2.1: g is $(2^{\lceil \log n \rceil}, \frac{7}{8})$-biased, and therefore also $(n, \frac{7}{8})$-biased. Therefore, also by corollary 3.2.1: XOR-ing 6 copies of g yields an $(n, \frac{1}{2})$-biased random function.

The formula size of each g_i is $2^i \cdot O(n)$. Thus, the total formula size is:

$$6 \cdot \sum_{i=0}^{\lceil \log n \rceil + 1} i \cdot O(n) = O(n \log^2 n)$$

□

Note 7.1. *The construction presented here is the smallest one known to us for achieving $(n, 2^{-n})$-bias. However, its formula can be described by three layers of high fan-in XOR and AND gates. Therefore, it cannot be a PRF. A similar construction can be achieved using the XIST gate, which also achieves $\tilde{O}(n^2)$ formula size, albeit with a higher power in the logarithmic factor. That construction cannot be readily "flattened", which gives some hope that a good PRF candidate may eventually be found. At this stage, though, we are uncertain in this regard.*

Examining the final construction presented above, one can see that it contains $2\log n + c_1$ levels of XOR gates and $2\log\log n + c_2$ levels of AND gates, for some constants c_1 and c_2. This is why the formulae have size roughly $2^{2\log n + 2\log\log n} = n^2 \log^2 n$.

When moving on to De-Morgan formulae, it is possible to replace each XOR gate with the following predicate:

$$\text{XOR}(x, y) = (x \vee y) \wedge \neg(x \wedge Y)$$

Where $\wedge, \vee$ and $\neg$ are the AND, OR and NOT gates, respectively.

Since we are dealing with formulae (i.e. fan-out 1), the values of x and y need to be computed twice each in order to implement this predicate. In conclusion, we have the following corollary.

Corollary 7.4.2. *There exists an $(n, 2^{-n})$-biased random function $f : \{0,1\}^n \to \{0,1\}$ which can be implemented by De-Morgan formulae of size $O(n^4 \log^2 n)$.*

Appendices

A Proof of Lemma 5.3

In this section we prove a version of the Diaconis-Shahshahani bias-to-statistical-distance lemma from [Dia88] which is suitable for our purposes. The proof follows the outline presented in [NN90], in which it was stated for distributions over $\{-1,1\}^k$. We restate it here for convenience.

Lemma 5.3. *Let $\mathcal{F} : D \to \mathbb{Z}_q$ be a (k,ϵ)-biased hash function family for some domain D. Then $\mathcal{F}$ is $(k, q^{k/2}\epsilon)$-wise independent.*

Proof. Let $\boldsymbol{x} \in D^k$ be a set of k distinct inputs and $\mathcal{D} = \mathcal{D}_{\mathcal{F},\boldsymbol{x}}$.

Recall that the bias of $\mathcal{D}$ w.r.t. a linear test $\boldsymbol{s} \in \mathbb{Z}_q^k$ is defined as:

$$\text{bias}_{\mathcal{D}}(\boldsymbol{s}) = \mathop{\mathbb{E}}_{\boldsymbol{X}\sim\mathcal{D}}[\chi_{\boldsymbol{s}}(\boldsymbol{X})] = \sum_{\boldsymbol{x}\in\mathbb{Z}_q^k} \mathcal{D}(\boldsymbol{x})\chi_{\boldsymbol{s}}(\boldsymbol{x}) = q^k\langle\mathcal{D},\chi_{\boldsymbol{s}}\rangle \tag{3}$$

Let $\hat{\mathcal{D}}$ denote the Fourier coefficients of $\mathcal{D}$. Using (3):

$$\hat{\mathcal{D}}(\boldsymbol{s}) = \langle\mathcal{D},\chi_{\boldsymbol{s}}\rangle = q^{-k}\text{bias}_{\mathcal{D}}(\boldsymbol{s}) \tag{4}$$

Let us calculate $\hat{\mathcal{D}}(\mathbf{0})$:

$$\hat{\mathcal{D}}(\mathbf{0}) = \langle\mathcal{D},\chi_{\mathbf{0}}\rangle = q^{-k}\sum_{\boldsymbol{x}\in\mathbb{Z}_q^k}\mathcal{D}(\boldsymbol{x})\cdot e^{\frac{2\pi i}{q}\sum_{j\in[k]}x_j\cdot 0} = q^{-k}\sum_{\boldsymbol{x}\in\mathbb{Z}_q^k}\mathcal{D}(\boldsymbol{x}) = q^{-k} \tag{5}$$

Now, using (5):

$$\begin{aligned}
|\mathcal{D}-\mathcal{U}|^2 &= \left(\sum_{\boldsymbol{x}\in\mathbb{Z}_q^k}\left|\mathcal{D}(\boldsymbol{x})-q^{-k}\right|\right)^2 \\
&\le q^k\cdot\sum_{\boldsymbol{x}\in\mathbb{Z}_q^k}\left(\mathcal{D}(\boldsymbol{x})-q^{-k}\right)^2 \\
&= q^k\sum_{\boldsymbol{x}\in\mathbb{Z}_q^k}\left(\sum_{\boldsymbol{s}\in\mathbb{Z}_q^k}\hat{\mathcal{D}}(\boldsymbol{s})\chi_{\boldsymbol{s}}(\boldsymbol{x})-q^{-k}\right)^2 \\
&= q^k\sum_{\boldsymbol{x}\in\mathbb{Z}_q^k}\left(\sum_{\boldsymbol{s}\neq\mathbf{0}}\hat{\mathcal{D}}(\boldsymbol{s})\chi_{\boldsymbol{s}}(\boldsymbol{x})\right)^2
\end{aligned}$$

Opening this expression, since functions $\chi_{\boldsymbol{s}}$ are orthonormal and using equation (4) we get the following:

$$\begin{aligned}
q^k \sum_{\boldsymbol{x}\in\mathbb{Z}_q^k} \left(\sum_{\boldsymbol{s}\neq\boldsymbol{0}} \hat{\mathcal{D}}(\boldsymbol{s})\chi_{\boldsymbol{s}}(\boldsymbol{x}) \right)^2 &= q^k \sum_{\boldsymbol{x}\in\mathbb{Z}_q^k} \sum_{\boldsymbol{s_1}\neq\boldsymbol{0}} \sum_{\boldsymbol{s_2}\neq\boldsymbol{0}} \hat{\mathcal{D}}(\boldsymbol{s_1})\hat{\mathcal{D}}(\boldsymbol{s_2})\chi_{\boldsymbol{s_1}}(\boldsymbol{x})\chi_{\boldsymbol{s_2}}(\boldsymbol{x}) \\
&= q^k \sum_{\boldsymbol{s_1}\neq\boldsymbol{0}} \sum_{\boldsymbol{s_2}\neq\boldsymbol{0}} \hat{\mathcal{D}}(\boldsymbol{s_1})\hat{\mathcal{D}}(\boldsymbol{s_2}) \sum_{\boldsymbol{x}\in\mathbb{Z}_q^k} \chi_{\boldsymbol{s_1}}(\boldsymbol{x})\chi_{\boldsymbol{s_2}}(\boldsymbol{x}) \\
&= q^{2k} \sum_{\boldsymbol{s_1}\neq\boldsymbol{0}} \sum_{\boldsymbol{s_2}\neq\boldsymbol{0}} \hat{\mathcal{D}}(\boldsymbol{s_1})\hat{\mathcal{D}}(\boldsymbol{s_2}) \langle \chi_{\boldsymbol{s_1}}(\boldsymbol{x})\chi_{\boldsymbol{s_2}}(\boldsymbol{x}) \rangle \\
&= q^{2k} \sum_{\boldsymbol{s}\neq\boldsymbol{0}} \hat{\mathcal{D}}(\boldsymbol{s_1})^2 \\
&= q^{2k} \sum_{\boldsymbol{s}\neq\boldsymbol{0}} \left(q^{-k} \text{bias}_{\mathcal{D}}(\boldsymbol{s}) \right)^2 \\
&= \sum_{\boldsymbol{s}\neq\boldsymbol{0}} \text{bias}_{\mathcal{D}}(\boldsymbol{s})^2 \\
&\leq q^k \epsilon^2
\end{aligned}$$

Which implies:

$$|\mathcal{D} - \mathcal{U}| \leq q^{k/2}\epsilon$$

□

B Proof of Lemma 6.2

In this appendix we prove lemma 6.2. This entails identifying bit strings in $\mathbb{Z}_2^l$ with integers in $\mathbb{Z}_{2^l}$. We use these forms interchangeably.

We restate the lemma here for convenience.

Lemma 6.2. *Let C be an ϵ-balanced linear code and let* $\text{Enc} : \mathbb{Z}_2^n \to \mathbb{Z}_2^{\frac{n}{R}}$ *be its encoding function. Let $\mathcal{F}$ be a family of functions of the form $Z_2^n \to \mathbb{Z}_2^l$ defined as follows. For any input $\boldsymbol{x} \in \{0,1\}^n$ and for each output bit $i \in [l]$: independently sample a uniformly random bit from the codeword* $\text{Enc}(\boldsymbol{x})$ *and flip it with probability $\frac{1}{2}$. Then $\mathcal{F}$ is $(2, 2\epsilon)$-biased.*

Proof of lemma 6.2. Let $\boldsymbol{X} = (\boldsymbol{x^1}, \boldsymbol{x^2}) \in \mathbb{Z}_{2^l}^2$ be a pair of distinct inputs, and let $\boldsymbol{Y} = (\boldsymbol{y^1}, \boldsymbol{y^2}) = (F(\boldsymbol{x^1}), F(\boldsymbol{x^2}))$ for a randomly sampled $F \sim \mathcal{F}$. Similarly, let $\boldsymbol{S} = (\boldsymbol{s^1}, \boldsymbol{s^2}) \in \mathbb{Z}_{2^l}^2$ define a non-empty linear test. That is to say there exists $i \in \{1, 2\}$ s.t. $\boldsymbol{s^i} \neq 0$.

We are interested in the quantity:

$$\text{bias}_{\hat{F}}(\boldsymbol{X}) = \left| \mathop{\mathbb{E}}_{\boldsymbol{\rho} \xleftarrow{R} \{0,1\}^{r_0}} \left[e^{\frac{2\pi i}{2^l} \langle \boldsymbol{S}, \boldsymbol{Y} \rangle} \right] \right|$$

Where the expectation is over sampling a function from $\mathcal{F}$.

Note that for all $\boldsymbol{y} \in \mathbb{Z}_{2^l}$, we may write:

$$\boldsymbol{y} = \sum_{j \in [l]} 2^{j-1} y_j$$

This yields:

$$\begin{aligned} e^{\frac{2\pi i}{2^l}\langle \boldsymbol{S}, \boldsymbol{Y} \rangle} &= e^{\frac{2\pi i}{2^l}\left(\boldsymbol{s^1} \cdot \boldsymbol{y^1} + \boldsymbol{s^2} \cdot \boldsymbol{y^2}\right)} \\ &= e^{\frac{2\pi i}{2^l}\boldsymbol{s^1} \sum_{j \in [l]} 2^{j-1} y_j^1} \cdot e^{\frac{2\pi i}{2^l}\boldsymbol{s^2} \sum_{j \in [l]} 2^{j-1} y_j^2} \\ &= z_1^{\sum_{j \in [l]} 2^{j-1} y_j^1} \cdot z_2^{\sum_{j \in [l]} 2^{j-1} y_j^2} \\ &= \prod_{j \in [l]} z_1^{2^{j-1} y_j^1} \cdot z_2^{2^{j-1} y_j^2} \end{aligned} \tag{6}$$

With $z_j = e^{\frac{2\pi i}{2^l}\boldsymbol{s^i}}$ for $j \in \{1, 2\}$. Note that $z_j^{2^l} = 1$ and that $z_j = 1 \iff \boldsymbol{s^j} = 0$.

Going back to the bias, we have:

$$\begin{aligned} \left| \mathbb{E}\left[e^{\frac{2\pi i}{2^l}\langle \boldsymbol{S}, \boldsymbol{Y} \rangle} \right] \right| &= \left| \mathbb{E}\left[\prod_{j \in [l]} z_1^{2^{j-1} y_j^1} \cdot z_2^{2^{j-1} y_j^2} \right] \right| \\ &= \left| \prod_{j \in [l]} \mathbb{E}\left[z_1^{2^{j-1} y_j^1} \cdot z_2^{2^{j-1} y_j^2} \right] \right| \\ &\leq \prod_{j \in [l]} \left| \mathbb{E}\left[z_1^{2^{j-1} y_j^1} \cdot z_2^{2^{j-1} y_j^2} \right] \right| \\ &\leq \min_{j \in [l]} \left| \mathbb{E}\left[z_1^{2^{j-1} y_j^1} \cdot z_2^{2^{j-1} y_j^2} \right] \right| \end{aligned} \tag{7}$$

Where we used (6) in the first equality and independence of each pair (y_j^1, y_j^2) from the other pairs of the same form in the second one. The third row inequality is a triangle inequality and the final one follows from the fact that all factors in the expression are non-negative.

Let $p = \Pr\left[y_j^1 = y_j^2\right]$ denote the probability that $\boldsymbol{y^1}$ and $\boldsymbol{y^2}$ agree on index j. Note that p is independent of j since each pair (y_j^1, y_j^2) is distributed identically and independently from other such pairs. Also, since C is ϵ-balanced: $\frac{1}{2} - \epsilon \leq p \leq \frac{1}{2} + \epsilon$. Note that since each output bit is flipped w.p. $\frac{1}{2}$, then:

$$\Pr\left[y_j^1 = 0\right] = \Pr\left[y_j^1 = 1\right] = \frac{1}{2}$$

Let us calculate the expectation in the final expression from (7):

$$\begin{aligned}\mathbb{E}\left[z_1^{2^{j-1}y_j^1}\cdot z_2^{2^{j-1}y_j^2}\right] =& 1\cdot \Pr\left[y_j^1=y_j^2=0\right]+z_1^{2^{j-1}}\cdot\Pr\left[y_j^1=1\wedge y_j^2\neq y_j^1\right]\\ &+z_2^{2^{j-1}}\cdot\Pr\left[y_j^1=0\wedge y_j^2\neq y_j^1\right]+(z_1\cdot z_2)^{2^{j-1}}\cdot\Pr\left[y_j^1=y_j^2=1\right] \qquad (8)\\ =&\frac{p}{2}+\frac{1-p}{2}z_1^{2^{j-1}}+\frac{1-p}{2}z_2^{2^{j-1}}+\frac{p}{2}(z_1\cdot z_2)^{2^{j-1}}\end{aligned}$$

Now, let k_i be the order of z_i as a root of unity for $i\in\{1,2\}$. Note that $k_i=2^{\kappa_i}$ for some $\kappa_i\in\mathbb{N}$. Also note that $\kappa_i=1\iff z_i=1$, and therefore for at least one of the values i: $\kappa_i>1$.

We have two cases. If $\kappa_1\neq\kappa_2$ then assume without loss of generality that $\kappa_1>\kappa_2$. Therefore, for $j=\kappa_1$: $z_1^{2^j}=-1$ and $z_2^{2^j}=1$. In this case the expression in (8) evaluates to:

$$\frac{p}{2}+\frac{1-p}{2}z_1^{2^{j-1}}+\frac{1-p}{2}z_2^{2^{j-1}}+\frac{p}{2}(z_1\cdot z_2)^{2^{j-1}}=\frac{p}{2}(1-1)+\frac{1-p}{2}(1-1)=0$$

In which the final equality follows from the definition of p and the fact that $\Pr\left[y_j^1=0\right]=\frac{1}{2}$.

If $\kappa_1=\kappa_2$, then for $j=\kappa$: $z_1^{2^{j-1}}=z_2^{2^{j-1}}=-1$. Therefore, the expression can be bounded by:

$$\begin{aligned}\frac{p}{2}+\frac{1-p}{2}z_1^{2^{j-1}}+\frac{1-p}{2}z_2^{2^{j-1}}+\frac{p}{2}(z_1\cdot z_2)^{2^{j-1}} &= p-(1-p)\\ &=2p-1\\ &\leq 2\epsilon\end{aligned}$$

□

Acknowledgements

The insight that our methods can be used generically for any class of hash function families was suggested to us by Moni Naor.

The idea to use the Valiant construction as a basis for PRF candidates with small-sized formulae was suggested by Yuval Ishai.

Lemma 5.11 was originally motivated by a construction proposed by Eyal Rosenman and Avi Wigderson. The version presented here is a generalization of a simpler and more efficient form that was suggested and proved to us by Amnon Ta-Shma.

The proof of lemma 5.5 presented here is based on an improvement suggested by Prashant Vasudevan. This improvement allowed us to get better parameters for this lemma.

Bibliography

[Nec66] E. I. Nechiporuk. “On a Boolean function”. In: vol. 7. 4. 1966, pp. 999–1000.

[CW79] J.Lawrence Carter and Mark N. Wegman. “Universal classes of hash functions”. In: *Journal of Computer and System Sciences* 18.2 (1979), pp. 143–154. ISSN: 0022-0000. DOI: http://dx.doi.org/10.1016/0022-0000(79)90044-8. URL: http://www.sciencedirect.com/science/article/pii/0022000079900448.

[WC81] Wegman and Carter. “New hash functions and their use in authentication and set equality”. In: *Journal of Computer and System Sciences* 22.3 (1981), pp. 265–279. ISSN: 0022-0000. DOI: http://dx.doi.org/10.1016/0022-0000(81)90033-7. URL: http://www.sciencedirect.com/science/article/pii/0022000081900337.

[Val84] L.G Valiant. “Short monotone formulae for the majority function”. In: *Journal of Algorithms* 5.3 (1984), pp. 363–366. ISSN: 0196-6774. DOI: http://dx.doi.org/10.1016/0196-6774(84)90016-6. URL: http://www.sciencedirect.com/science/article/pii/0196677484900166.

[Alo86] Noga Alon. “Eigenvalues and expanders”. In: *Combinatorica* 6.2 (1986), pp. 83–96. ISSN: 1439-6912. DOI: 10.1007/BF02579166. URL: http://dx.doi.org/10.1007/BF02579166.

[VV86] L.G. Valiant and V.V. Vazirani. “NP is as easy as detecting unique solutions”. In: *Theoretical Computer Science* 47 (1986), pp. 85–93. ISSN: 0304-3975. DOI: http://dx.doi.org/10.1016/0304-3975(86)90135-0. URL: http://www.sciencedirect.com/science/article/pii/0304397586901350.

[Vaz86] Umesh Virkumar Vazirani. “Randomness, Adversaries and Computation (Random Polynomial Time)”. AAI8718194. PhD thesis. 1986.

[AKS87] M. Ajtai, J. Komlos, and E. Szemeredi. “Deterministic Simulation in LOGSPACE”. In: *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*. STOC ’87. New York, New York, USA: ACM, 1987, pp. 132–140. ISBN: 0-89791-221-7. DOI: 10.1145/28395.28410. URL: http://doi.acm.org/10.1145/28395.28410.

[And87] A.E. Andreev. “On A Method For Obtaining More Than Quadratic Effective Lower Bounds For The Complexity of π-schemes”. In: *Moscow Univ. Math. Bull.* 1. 1987, pp. 63–66.

[Dia88] Persi W. Diaconis. *Group representations in probability and statistics*. Lecture notes. Hayward, Calif. Institute of Mathematical Statistics, 1988. ISBN: 0-940600-14-5. URL: http://opac.inria.fr/record=b1087294.

[Bab89] L. Babai. *Fourier Transforms and Equations over Finite Abelian Groups, Lecture Notes*. 1989.

[NN90] J. Naor and M. Naor. “Small-bias Probability Spaces: Efficient Constructions and Applications”. In: *Proceedings of the Twenty-second Annual ACM Symposium on Theory of Computing*. STOC ’90. Baltimore, Maryland, USA: ACM, 1990, pp. 213–223. ISBN: 0-89791-361-2. DOI: 10.1145/100216.100244. URL: http://doi.acm.org/10.1145/100216.100244.

[IN93] Russell Impagliazzo and Noam Nisan. “The Effect of Random Restrictions on Formula Size”. In: *Random Structures and Algorithms, Vol 4, No*. 1993, pp. 121–134.

[RR97] Alexander A Razborov and Steven Rudich. “Natural Proofs”. In: *Journal of Computer and System Sciences* 55.1 (1997), pp. 24–35. ISSN: 0022-0000. DOI: http://dx.doi.org/10.1006/jcss.1997.1494. URL: http://www.sciencedirect.com/science/article/pii/S002200009791494X.

[AMN98] Yossi Azar, Rajeev Motwani, and (Seffi) Joseph Naor. “Approximating Probability Distributions Using Small Sample Spaces”. In: *Combinatorica* 18.2 (1998), pp. 151–171. ISSN: 1439-6912. DOI: 10.1007/PL00009813. URL: http://dx.doi.org/10.1007/PL00009813.

[Hås98] Johan Håstad. “The shrinkage exponent of de Morgan formulas is 2”. In: *SIAM JOURNAL ON COMPUTING* (1998).

[PR04] Rasmus Pagh and Flemming Friche Rodler. “Cuckoo Hashing”. In: *J. Algorithms* 51.2 (May 2004), pp. 122–144. ISSN: 0196-6774. DOI: 10.1016/j.jalgor.2003.12.002. URL: http://dx.doi.org/10.1016/j.jalgor.2003.12.002.

[Sie04] Siegel. “On Universal Classes of Extremely Random Constant-Time Hash Functions”. In: *SIAM J. Comput.* 33.3 (Mar. 2004), pp. 505–543. ISSN: 0097-5397. DOI: 10.1137/S0097539701386216. URL: http://dx.doi.org/10.1137/S0097539701386216.

[TZ04] Mikkel Thorup and Yin Zhang. “Tabulation Based 4-universal Hashing with Applications to Second Moment Estimation”. In: *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA ’04. New Orleans, Louisiana: Society for Industrial and Applied Mathematics, 2004, pp. 615–624. ISBN: 0-89871-558-X. URL: http://dl.acm.org/citation.cfm?id=982792.982884.

[HLW06] Shlomo Hoory, Nathan Linial, and Avi Wigderson. “Expander graphs and their applications”. In: *BULL. AMER. MATH. SOC.* 43.4 (2006), pp. 439–561.

[PPR11] Anna Pagh, Rasmus Pagh, and Milan Ružić. “Linear Probing with 5-wise Independence”. In: *SIAM Rev.* 53.3 (Aug. 2011), pp. 547–558. ISSN: 0036-1445. DOI: 10.1137/110827831. URL: http://dx.doi.org/10.1137/110827831.

[Hus+12] Mohammad Iftekhar Husain et al. “Almost Universal Hash Families are also Storage Enforcing”. In: *CoRR* abs/1205.1462 (2012). URL: http://arxiv.org/abs/1205.1462.

[PT13] Pătraşcu and Thorup. “Twisted Tabulation Hashing”. In: *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA ’13. New Orleans, Louisiana: Society for Industrial and Applied Mathematics, 2013, pp. 209–228. ISBN: 978-1-611972-51-1. URL: http://dl.acm.org/citation.cfm?id=2627817.2627833.

[ODo14] Ryan O’Donnell. *Analysis of Boolean Functions*. New York, NY, USA: Cambridge University Press, 2014. ISBN: 1107038324, 9781107038325.

[GV15] Timothy Gowers and Emanuele Viola. “The Communication Complexity of Interleaved Group Products”. In: *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*. STOC ’15. Portland, Oregon, USA: ACM, 2015, pp. 351–360. ISBN: 978-1-4503-3536-2. DOI: 10.1145/2746539.2746560. URL: http://doi.acm.org/10.1145/2746539.2746560.

תקציר

משפחות של פונקציות גיבוב נקראות כמעט k בלתי תלויות אם ההתפלגות של הפלטים שלהן על פני כל קבוצה של k קלטים שונים קרובה להתפלגות האחידה בנורמה L_1; תכונה זו נקראת גם אי-תלות מוגבלת. תכונה קרובה אליה היא זו הנקראת כמעט k חוסר-הטיה, המוגדרת ע"י כך שההתפלגות של פלטים של k קלטים שונים קרובה להתפלגות האחידה בנורמה L_∞.

בעבודה זו חקרנו שיטות להגדלה של אי-תלות מוגבלת של משפחות פונקציות גיבוב. כלומר, בהנתן משפחה של פונקציות גיבוב כמעט k בלתי-תלויה ,או בלתי-מוטה, מטרתנו היא ליצור משפחה חדשה של משפחה שהינה כמעט k' בלתי-תלויה או בלתי-מוטה עבור $k' > k$. ההעתקות (טרנספורמציות) שלנו הינן כלליות במובן שהן מתייחסות למשפחת פונקציות הגיבוב המקורית כאל קופסא שחורה וברוב המקרים אינן דורשות להניח דבר לגבי משפחה זו מעבר לתכונה הבסיסית של כמעט k חוסר-תלות. למיטב ידיעתנו, שיטה כזו לא פורסמה עד כה.

בכדי להשיג את מטרותינו, אנו משתמשים בדגימה חוזרת מהמשפחה המקורית וחיבור של הדגימות באמצעות פונקציה כלשהי. אנו מזהים שני סוגים של פונקציות שיעילות למטרה זו: הסוג הראשון הוא פונקציות אשר מקטינות את המרחק בנורמה L_∞ בין ההתפלגות של הפלטים להתפלגות האחידה; הסוג השני מאפשר לנו להגדיל את פרמטר אי-התלות k. לבסוף, אנו מחברים את שני סוגי הפונקציות הללו בתהליך איטרטיבי אשר בסופו של דבר משיג את התכונות הדרושות.

עבודה זו בוצעה בהדרכתו של אלון רוזן. העבודה מבוססת על מחקר משותף עם אנדרי בוגדנוב ואלון רוזן.

המרכז הבינתחומי הרצליה

בית הספר ארזי למדעי המחשב

התכנית לתואר שני (M.Sc.) - מסלול מחקרי

שיטה כללית להרחבה של אי-תלות מוגבלת

מאת

ארבל דויטש פלד

עבודת תזה המוגשת כחלק מהדרישות לשם קבלת תואר מוסמך M.Sc.

במסלול המחקרי בבית הספר אפי ארזי למדעי המחשב, המרכז הבינתחומי הרצליה

דצמבר 2016