Electronic Colloquium on Computational Complexity, Report No. 59 (2026)

Locally Computable High Independence Hashing

Yevgeniy Dodis
NYU

Shachar Lovett*
UCSD

Daniel Wichs †
Northeastern and NTT Research

March 30, 2026

Abstract

We consider (almost) k-wise independent hash functions, whose evaluations on any k inputs are (almost) uniformly random, for very large values of k. Such hash functions need to have a large key that grows linearly with k. However, it may be possible to evaluate them in sub-linear time by only reading a small subset of $t \ll k$ locations during each evaluation; we call such hash functions t-*local*. Local hash functions were previously studied in several works starting with Siegel (FOCS'89, SICOMP'04). For a hash function with n-bit input and output size, we get the following new results:

- There *exist* (non-constructively) perfectly k-wise independent t-local hash functions with key size $O(kn)$ and locality of $t = O(n)$ *bits*. An analogous prior result of Larsen et al. (ICALP '24) had a locality of $t = O(n)$ *words* consisting of $w = O(n)$ bits each, and hence a suboptimal $O(n^2)$ bits total. Furthermore, we show that such hash functions could be made *explicit* if we had explicit optimal constructions of *unbalanced bipartite lossless expanders*. Plugging in currently best known suboptimal explicit expanders yields correspondingly suboptimal hash functions.
- Perfectly k-wise independent local hash functions generically yield expanders with corresponding parameters. This is true even if the locations accessed by the hash function can be chosen adaptively and shows that progress on explicit hash functions inherently requires progress on explicit expanders.
- We initiate the study of ε-almost k-wise independent hash functions, where any k adaptive queries to the hash function are ε-statistically indistinguishable from k queries to a random function. We construct an *explicit* family of such hash functions with optimal key size $O(kn)$ bits, optimal locality $t = O(n)$ bits, and $\varepsilon = 2^{-n}$, significantly improving over the best known parameters for explicit perfectly independent hashing.
- More generally, if we consider a word model with larger word size w, then we get an explicit, efficient construction of ε-almost k-wise independent hash functions with key size $O(kn/w)$ words, locality $t = O(n/\sqrt{w})$ words, and statistical distance $\varepsilon = 2^{-n}$, which we show to be nearly optimal. Such parameters go beyond what is possible for perfect independence.

We discuss applications to nearly optimal bounded-use information-theoretic cryptography.

*Research supported by Simons Investigator Award 929894 and NSF award CCF-2425349.
†Research supported by NSF CNS-2349972 and CNS-2055510.

ISSN 1433-8092

Contents

1 Introduction **1**
1.1 Prior Work 2
1.2 Our Results 3
1.2.1 Perfectly Independent Hashing 4
1.2.2 Almost Independent Hashing 4

2 Preliminaries **5**

3 Definitions of Independent Local Hashing **7**

4 Perfectly Independent Local Hashing **8**
4.1 Construction from Lossless Expanders 8
4.2 Perfect Local Hashing Implies Expanders 9

5 Almost Independent Local Hashing **10**
5.1 Lower Bound 10
5.2 Optimal Bit-Local Construction 12
5.3 Optimal Word-Local Construction 15

6 Summary and Open Problems **18**

1 Introduction

We consider a family of (almost) k-wise independent hash functions $\mathcal{F} = \{f_s : \{0,1\}^n \to \{0,1\}^n\}$ keyed by s. We say that such a function family is *perfectly* k-wise independent if the evaluations of $f_s(x_1), \ldots, f_s(x_k)$ for any k distinct inputs $x_1, \ldots, x_k$ are uniformly random and independent over a random choice of the key s. We say that such a family is ε-*almost* k-wise independent if no statistical distinguisher that makes at most k adaptive oracle queries can distinguish between having oracle access to f_s for a random s versus oracle access to a truly random function with advantage greater than ε.

Such (almost) independent hash functions have numerous applications in algorithm design, data structures and cryptography. We are particularly interested in a setting where k is huge – e.g., on the order of thousands, millions or even billions. It is easy to see that the bit-length of the key s needs to be at least kn to ensure that k outputs are (almost) uniform, and hence the key size must also get huge as k does. However, it may not be necessary to read the entire key during each evaluation of $f_s(x)$. We say that a family $\mathcal{F}$ is t-*local* if, for any input x, the evaluation of $f_s(x)$ only reads at most t locations of the key s. Our goal is to achieve locality $t \ll k$ so that the function evaluation remains extremely efficient even as the independence k and the corresponding key size get huge. By default we consider the *bit model* where we think of $s \in \{0,1\}^\ell$ as a bit string and require the evaluation $f_s(x)$ to only read at most t bits of s. We call such functions t-bit-local. We also generalize to the *word model*, where we think of $s \in \Sigma^\ell$ as a string over some alphabet $\Sigma = \{0,1\}^w$ with word size w, and require the evaluation $f_s(x)$ to only read at most t words in s. We call such functions t-word-local. A bit-local construction is also word-local with the same t, but one may hope to achieve smaller word locality as the word size gets larger.

A Cryptographic Motivation. Our main motivation comes from cryptography with *information-theoretic (IT) security*, where we do not place any constrains on the computational power of the adversary. IT security is very attractive as it enables provably secure schemes that resist advances in computational power, novel cryptanalysis, or the possibility of quantum computers. It is well known that IT security requires large secret keys. For symmetric-key encryption, the keys size must be as large as the total size of all of the messages to be encrypted [Sha49], and for authentication, it must grow with the total number of messages to be authenticated [GN93]. By itself, this might not be a problem, as local storage is cheap and it may be possible to store huge secret keys (many gigabytes or even terabytes) in practice. However, it is highly impractical to read the entire huge secret key during each encryption/authentication operation. Therefore, we would like to have *locally computable* solutions, where each operation is very efficient and only reads a small subset of locations within the secret key.

It is easy to come up with *stateful* solutions to this problem. For encryption, the users can store a huge random key, and to encrypt each successive message they could one-time pad (XOR) the next unused chunk of key bits with the message being encrypted. Similarly, for authentication, users can authenticate each successive message by using the next fresh chunk of key bits as a short key of a one-time information-theoretic message authentication code. However, both of these trivial solutions require the users to be stateful and synchronized. Unfortunately, stateful solutions are cumbersome to use and the reliance on state can lead to both correctness and security issues. If the parties get out of sync and use different chunks of the key (e.g., to encrypt and decrypt) then correctness fails. If a device is reset and fails to properly keep state then it may reuse key bits which completely breaks security. Lastly, if we want many mutually trusting parties to securely communicate using one shared secret key then it will not be possible to keep consistent state between them – for example, if Alice and Bob separately authenticate a message to Charlie without any coordination, they may use the same chunk of the secret key and violate security.[1]

Due to the above issues, the gold standard for symmetric-key cryptography is to have *stateless* solutions. In the computational setting, the standard techniques for constructing stateless symmetric-key encryption and authentication is to use *pseudorandom functions (PRFs)*. The same techniques port over to the

[1] While this can be fixed by having a separate key between each pair of communicating parties, if the number of parties is large and the communication pattern between them is uneven and unknown a-priori, then this would add a large overhead beyond what is necessary.

information-theoretic setting and allow us to construct stateless encryption and authentication that is secure for up to k uses by relying on ε-almost k-wise independent hash function family $\mathcal{F}$ in place of pseudorandom functions. The shared key is the key s of a function $f_s : \{0,1\}^n \to \{0,1\}^n \in \mathcal{F}$. To encrypt a message $\mathsf{msg} \in \{0,1\}^n$, we can choose a random "nonce" $r \leftarrow \{0,1\}^n$ and set the ciphertext to be $(r, f_s(r) \oplus \mathsf{msg})$. As long as the nonces do not repeat, the above is secure for encrypting up to k messages, and therefore achieves overall statistical security $O(k^2/2^n) + \varepsilon$. For authentication, we can just think of $f_s(\mathsf{msg})$ as the authentication tag of msg, and this allows us to securely authenticate up to $(k-1)$ messages ensuring that the adversary cannot guess a valid tag for any new message, except with probability $2^{-n} + \varepsilon$. Importantly, if the function family $\mathcal{F}$ is t-local for some $t \ll k$, then the price for handling a large number of uses k is only paid by having a correspondingly large key size, but not in the efficiency of the scheme.[2]

Parameters. Motivated by the above cryptographic applications, throughout the introduction we focus on hash functions

$$\mathcal{F} = \{f_s : \{0,1\}^n \to \{0,1\}^n\}_{s \in \Sigma^\ell},$$

where we the *input* and *output* size are both n, which we also think of as the "security parameter". We want to handle large independence $k = \mathsf{poly}(n)$. In the case of ε-almost k-wise independence, we will aim for exponentially small error $\varepsilon = 2^{-n}$. We consider both the bit model $\Sigma = \{0,1\}$ and the word model $\Sigma = \{0,1\}^w$ with word size $w = \mathsf{poly}(n)$. The goal is to minimize the key size ℓ to ideally just the minimal $\ell = O(kn)$ bits or $O(kn/w)$ words and to minimize the locality t to be as small as possible.

Locality vs. RAM Run-Time. In this work we focus on minimizing the locality t of the hash functions, which is also known as its cell-probe complexity. However, we also generally desire *explicit* constructions, where $f_s(x)$ must run in polynomial time. One could ask for a stronger property that the total run-time of $f_s(x)$ is bounded by t in the word-RAM model. While this is desirable, having explicit t-local constructions may often be good enough in cases where local computation is much faster than memory access.

1.1 Prior Work

The notion of locally computable k-wise independent hashing was initiated by Siegel [Sie89, Sie04]. We summarize what is known in terms of lower bounds, non-constructive existential results and efficient constructions. Throughout we focus on hash functions having input/output size n-bits and independence $k = \mathsf{poly}(n)$. The prior work focused on the word model with word size $w = \Theta(n)$.

Lower bound. Siegel showed a lower bound for perfect k-wise independent hashing. When the key size is $\mathsf{poly}(n)$, then the locality must be at least $t = \Omega(n/\log n)$ words when $k > t$. Furthermore if the key size is just the optimal $O(k)$ words, then the locality must be at least $t = \Omega(n)$ when $k > t$.

The lower-bound on t in the word model, also applies to the bit model, and shows that one needs to read at least t bits. However, it does not show that the locality has to get any larger in the bit model beyond this, leaving open the possibility of just reading t *bits* rather than t *words*.

Existential (Non-Constructive) Results. Siegel [Sie89, Sie04] shows that perfect k-wise independent local hashing with good parameters can be constructed from certain types of optimal expander graphs. However, we do not have good explicit (efficient) constructions of such expanders and only know of their existence

[2] An astute reader may notice that for encryption we only need a weak version of (almost) k-wise independence for random inputs rather than worst case inputs. This does indeed make the problem simpler and allows for alternative solutions based on local randomness extractors [Vad04, DY21]. For authentication, we only need each new output to be unpredictable given the previous outputs, but not necessarily (almost) uniformly random. However, in the information theoretic setting there is not a major difference between unpredictability and uniform randomness – we can always convert the former to the latter using randomness extractors. Therefore, (almost) k-wise independent hashing seems to be inherent in constructing stateless and deterministic information-theoretic authentication schemes.

via a non-constructive probabilistic method argument. Therefore this result is only a non-constructive existential result. Moreover, even if one were to plug in optimal existential expanders, the achieved parameters are suboptimal and do not meet the lower bound.

The was recently improved by the work of Larsen et al. [LPP+24] who gave an improved probabilistic method construction of k-wise independent t-word-local hashing with parameters matching Siegel's lower bound. However, this requires an additional use of a probabilistic method argument beyond expanders, and therefore does not show that optimal expanders suffice. Furthermore, it only gives an optimal construction in the word model. If we instead want to work in the bit model, then the construction incurs another factor of n overhead in bit-locality by having to read the entire word, but the lower bound does not require this! Therefore it was left open if one can meet the lower bound in the bit model, even existentially.

Explicit WHP Perfect Independence. Starting with Siegel [Sie89, Sie04], a line of works [ÖP03, PP08, Tho13, CPT15] culminates in the constructions of efficient local hash functions that are *with high probability* $1-\varepsilon$ *perfectly k-wise independent*, which we refer to as *ε-WHP k-wise independent*. For these, we think of the key $s = (s_0, s_1)$ as consisting of two components and, with high probability $1-\varepsilon$ over the choice of s_0, the function f_{s_0,s_1} is perfectly k-wise independent over a random choice of s_1. The key size and the locality are measured in terms of the entire key s.

Note that ε-WHP k-wise independence is a strong notion that implies ε-almost k-wise independence, but not vice versa. Since ε-WHP k-wise independent hash functions are also perfectly k-wise independent for some choice of s_0, they are subject to the same lower bounds as perfect k-wise independence. However, it is potentially easier to come up with explicit constructions.

The prior works generally bound the total RAM run time t of evaluating the hash function, and not just the locality. For our parameter setting, the state of the art results from [CPT15] give ε-WHP k-wise independent hashing with word size $w = \Theta(n)$ and:

- for any constant $\delta > 0$ they achieve key size $O(kn^\delta)$ words, locality $t = \widetilde{O}(n)$ words and error $\varepsilon = 1/n^\delta$,
- or they achieve key size $\widetilde{O}(kn^2)$ words, locality $t = \widetilde{O}(n^2)$ words and error $\varepsilon = 2^{-n}$.

The first result nearly matches Siegel's space and locality lower bound in the word model when we make δ small, but the locality becomes suboptimal by a factor of $O(n)$ in the bit model. Moreover, the error probability is too high for cryptoraphic applications, where we want error $\varepsilon = 2^{-n}$. The second result gets the desired error probability 2^{-n} but both the key size and the locality are both suboptimal, even in the word model.

1.2 Our Results

Prior work leaves several interesting open directions that we explore here, especially for the setting of parameters that is of interest for the cryptographic motivation.

Firstly, we close off gaps in our understanding of perfectly k-wise independent local hash functions. We (non-constructively) show the existence of such hash functions with optimal key size $O(kn)$ bits and bit-locality $O(n)$, matching Siegel's lower bound. Previously, we only had an analogous result in the word model with word-locality $O(n)$ when the word size is $w = \Theta(n)$ bits, but in the bit model this would give a sub-optimal bit locality of $O(n^2)$. We also show that a certain forms of strongly explicit expenders suffice to get optimal explicit hash functions, and conversely, that they are also *necessary*. Therefore progress on explicit perfectly independent local hash function requires corresponding progress on explicit expanders.

Secondly, we initiate the study of *ε-almost k-wise independent local* hash functions, which is weaker than ε-WHP k-wise independence, but suffices for cryptographic applications. We give explicit constructions with optimal parameters and nearly matching lower bounds. In the bit model, our parameters essentially match the optimal parameters for perfect k-wise independence, but in this setting we can achieve them *explicitly*, while doing so remains open for the perfect setting. In the word model with larger word size, the parameters are provably *better* than what can be achieved for perfect k-wide independence, making this setting attractive if statistical security is sufficient (e.g., for cryptography).

We now provide details on each of these results in turn.

1.2.1 Perfectly Independent Hashing

Existential and Explicit Constructions. We show the existence of a perfect k-wise independent family of hash functions $f_s : \{0,1\}^n \to \{0,1\}^n$ in the bit model with key size $\ell = O(kn)$ bits and locality $t = O(n)$ bits, which is optimal by Siegel's lower bound. In fact, we show how to construct such a hash family using *unbalanced bipartite lossless expanders* with optimal parameters. We know that such expanders exist via a non-constructive probabilistic method argument, which implies the corresponding existence of the above hash functions.

If we had a *strongly explicit* construction of such expander graphs, where given a node v and an index i as inputs, we could efficiently compute i'th neighbor of v, then we would get an *explicit* construction of the above hash functions where $f_s(x)$ could be computable in $\mathsf{poly}(n, \log k)$ time. Unfortunately, we do not know (strongly) explicit constructions with optimal parameters. However, we can use the best known (albeit suboptimal) strongly explicit constructions [GUV07] to get corresponding suboptimal parameters for perfectly k-wise independent hash functions. For any constant $\alpha > 0$ the locality is $t = O((n \log k)^{1+1/\alpha})$ bits and key size $O(k^{1+\alpha} t^2)$ bits.

Necessity of Expanders. We show that any explicit perfectly k-wise independent local hash function yields a strongly explicit expander graph with corresponding parameters. In particular, consider a t-local hash function with input/output size n bits and key size ℓ words over a word alphabet $\Sigma = \{0,1\}^w$. Then for a key s, consider the bipartite graph G_s that for every "left vertex" $x \in \{0,1\}^n$ has up to t edges to the "right vertices" $i \in [\ell]$ that correspond to the locations of the key accessed by the computation $f_s(x)$. We show that for every s the graph G_s must be a $(k, A = n/w)$-expander, meaning that any subset of k left vertices has at least kA neighbors. This is essentially obvious for hash functions with *static access*, where the set of key locations accessed by $f_s(x)$ does not dependent on the key s and therefore the above graph is the same for all keys s. We show that it also holds in the general case when the hash function may have *adaptive access*.

The above has several interesting corollaries. Firstly, by combining it with known expander lower bounds, we also re-derive Siegel's lower bound on the key length vs locality tradeoffs for perfect k-wise independence, and extend it to arbitrarily large word sizes. Secondly, it shows that progress on explicit local k-wise independent hash functions *requires* corresponding progress on strongly explicit expanders.

1.2.2 Almost Independent Hashing

We initiate the study of ε-almost k-wise independent t-local hash functions $f_s : \{0,1\}^n \to \{0,1\}^n$ with error $\varepsilon = 2^{-n}$. We consider both the bit-model with $s \in \{0,1\}^\ell$ and the word-model with $s \in \Sigma^\ell$ for word alphabet $\Sigma = \{0,1\}^w$ for some $w = \mathsf{poly}(n)$.

Lower Bound. We first note that Siegel's lower bound does not apply to almost k-wise independence. However, we show that for a sufficiently large $k = \mathsf{poly}(n)$, when the key size is $\ell = \mathsf{poly}(n)$ we must have locality $t = \Omega(n/(\sqrt{w} \log n))$. In particular, if the word size is $w = O(1)$, the locality must be $t = \widetilde{\Omega}(n)$, so one cannot do much better than what is existentially possible for perfect k-wise independence. However, the hope is that one can get explicit constructions without requiring improvements in explicit expanders. For larger word size there is hope to do better than what is even existentially possible in the perfect case. In particular when the word size is $w = O(n)$, the lower bound only requires locality $t = \widetilde{\Omega}(\sqrt{n})$ and when the word size is $w = O(n^2)$, the lower bound only requires locality $t = \Omega(1)$. We match these bounds below.

Constructions. We construct explicit hash functions that nearly match the above lower bounds. In the bit-model, we construct explicit hash functions with optimal key size $\ell = O(kn)$ bits and optimal locality $t = O(n)$. In the word model with sufficiently large word size $w = \Omega(\log^2 n + \log^2 k)$, we get optimal key

size $\ell = O(kn/w)$ words and nearly optimal word-locality $t = O(n/\sqrt{w})$. In particular, for $w = \Theta(n)$ we get word-locality $t = O(\sqrt{n})$, and for $w = \Theta(n^2)$ we get word-locality $t = O(1)$. As we mentioned, such parameters are *provably impossible* in the perfect independence setting.

Paper organization. In Section 2 we give preliminary definitions. In Section 3 we give formal definitions of local hash functions. Then in Section 4 we give lower and upper bounds for perfectly independent local hash functions, and show their intimate connection to expanders, and in Section 5 we give lower and upper bounds for almost independent local hash functions. Finally we summarize and pose a few open problems in Section 6.

2 Preliminaries

For an integer n we define $[n] := \{1, \ldots, n\}$ and for integers $n \leq m$ we define $[n, m] = \{n, n+1, \ldots, m\}$.

Definition 2.1 (Universality and ρ-almost universality)**.** *Let $\mathcal{H}$ be a family of functions $h : \{0,1\}^n \to \{0,1\}^\lambda$, and let $h \leftarrow \mathcal{H}$ denote a uniformly random choice from the family. We say that:*

- *$\mathcal{H}$ is* universal *if for all distinct $x, x' \in \{0,1\}^n$,*

$$\Pr_{h \leftarrow \mathcal{H}}\left[h(x) = h(x')\right] = 2^{-\lambda}.$$

- *$\mathcal{H}$ is* γ-almost universal *if for all distinct $x, x' \in \{0,1\}^n$,*

$$\Pr_{h \leftarrow \mathcal{H}}\left[h(x) = h(x')\right] \leq \gamma$$

The standard construction of γ-almost universal hash functions achieves the following.

Lemma 2.2 ([DH90])**.** *For any n, λ there is a γ-almost universal hash family $\mathcal{H}$ of functions $h : \{0,1\}^n \to \{0,1\}^\lambda$ with $\gamma \leq \lceil n/\lambda \rceil 2^{-\lambda} = O(n)2^{-\lambda}$ where each hash function $h \in \mathcal{H}$ can be described using $\log|\mathcal{H}| = \lambda$ bits.*

Proof. Use a hash function based on polynomial evaluation where $x \in \{0,1\}^n$ is interpreted as a polynomial of degree $d := \lceil n/\lambda \rceil$ over $\mathbb{F}_{2^\lambda}$ that we evaluate on random point in $\mathbb{F}_{2^\lambda}$ contained in the description of h. The probability that two different polynomials x, x' collide on the point specified by h is $\leq d/2^\lambda$. □

Expander Graphs. We recall the definition of bipartite expanders and known parameters.

Definition 2.3 ((lossless) bipartite expanders)**.** *Let $G : [U] \times [D] \to [V]$ be a function, which we think of as representing a bipartite graph with U left vertices, V right vertices, and left-degree at most D. For $x \in [U]$ we let $I(x) := \{G(x,i) : i \in [D]\} \subseteq [V]$ be the neighbors of x, and for a set $X \subseteq [U]$ we define $I(X) := \cup_{x \in X} I(x)$ to be the neighbors of X.*

- *We say G is a (k, A)-*expander *if for every set $X \subset [U]$ of size $|X| \leq k$, $|I(X)| \geq A|X|$.*
- *We say G is a (k, ε)-*lossless expander *if it is a (k, A)-expander for $A = (1-\varepsilon)D$.*
- *We say that G is* strongly explicit *if it can be computed in polynomial time.*

We are interested in unbalanced lossless expanders where $|U| = 2^n$ and $|V| = \mathsf{poly}(n)$.

Lemma 2.4 (Existential expanders; see e.g., [GUV07])**.** *For every $U \in \mathbb{N}$, $k \leq U$, and $\varepsilon > 0$, there exist (k, ε)-lossless expanders with degree $D = O(\log(U/V)/\varepsilon)$ and $V = O(kD/\varepsilon)$.*

Lemma 2.5 (Explicit expanders [GUV07]). *For all constants $\alpha > 0$: for every $U \in \mathbb{N}$, $k \le U$, and $\varepsilon > 0$, there is an explicit (k, ε)-lossless expander $G : [U] \times [D] \to [V]$ with degree $D = O\Big(\frac{(\log U)(\log k)}{\varepsilon}\Big)^{1+1/\alpha}$ and $V \le D^2 \cdot k^{1+\alpha}$.*

Lemma 2.6 (expander lower bound [RT97]). *If $G : [U] \times [D] \to [V]$ is a (k, A)-expander and $A \ge 2D/k$ then $D = \Omega\Big(\frac{\log(U/k)}{\log(V/(kA))}\Big)$. In particular, if $U = 2^n$ and $k, V = \mathsf{poly}(n)$ and $A \ge 2D/k$ then $D = \Omega(n/\log n)$; if in addition $V = O(kA)$ then $D = \Omega(n)$.*

Averaging samplers. We will use averaging samplers; see [Gol11] for a survey on the topic. Averaging samplers allow to approximate the average of a function while making a few queries to it, in a randomness efficient manner. They can be equivalently modeled as functions, which is how we choose to model them here; below, U corresponds to the internal randomness of the sampler, D to the set of queries, and V to the domain being sampled.

Definition 2.7 (Averaging sampler). *Let $\mathsf{Samp} : [U] \times [D] \to [V]$ and $\varepsilon, \delta \in (0, 1)$. We say that Samp is an averaging sampler with accuracy δ and error ε if, for any set $T \subset V$, it satisfies*

$$\Pr_{x \in [U]}\left[\left|\Pr_{y \in [D]}[\mathsf{Samp}(x, y) \in T] - \frac{|T|}{|V|}\right| \le \delta\right] \ge 1 - \varepsilon.$$

We say a sampler is explicit *if the function Samp can be computed in polynomial time.*

There are many constructions of averaging samplers, the most recent one from [XZ25] which gives a nice comparison table of many of the previous works. However, in our application we have some conditions which are different from typical applications:

1. For each $x \in U$, we need the outputs of $\mathsf{Samp}(x, y)$ to be distinct when ranging over y.
2. The parameter range of interest to us is where ε is exponentially small in our parameters, but δ is just a small enough absolute constant.
3. We would like freedom to choose any D (within some constraints), instead of minimizing it, which is one of the goals of many of the existing constructions.

We first formalize the first condition.

Definition 2.8 (Averaging sampler with distinct elements). *An averaging sampler $\mathsf{Samp} : [U] \times [D] \to [V]$ has* distinct elements, *if for each $x \in U$, the elements $(\mathsf{Samp}(x, y) : y \in [D])$ are all distinct.*

If we do not require distinct elements, than a standard construction based on expander random walks [Gil98] gives an explicit construction suitable for us.

Theorem 2.9 ([Gil98]). *Let $D, V \ge 1$, $\varepsilon, \delta \in (0, 1)$. There exists some $D_0 = O(\log(1/\varepsilon)/\delta^2)$ such that for all $D \ge D_0$ there exists an averaging sampler $\mathsf{Samp} : [U] \times [D] \to [V]$ with $|U| \le |V|(1/\delta)^{O(D)}$.*

For some of our applications it will be convenient to have distinct elements. For this we use a construction due to Vadhan [Vad04] that modifies the expander random walk based construction to avoid repeated elements.

Theorem 2.10 ([Vad04]). *Let $D, V \ge 1$, $\varepsilon, \delta \in (0, 1)$. There exists some $D_0 = O(\log(1/\varepsilon)/\delta^2)$ such that for for all $V \ge D \ge D_0$ there exists an averaging sampler $\mathsf{Samp} : [U] \times [D] \to [V]$ with distinct elements, where $|U| \le |V|(1/\delta)^{O(D)}$.*

Concentration Bounds. We will also rely on the following Chernoff-type bound for random variables with limited independence.

Lemma 2.11 (Limited Independence Chernoff [BR94, SSS95]). *Let $\delta > 0$ be a constant and let $X_1, \ldots, X_n$ be k-wise independent random variables in the interval $[0,1]$ with $X = \sum_{i \in [n]} X_i$ having expectation $E[X] \leq \mu$. Then $\Pr[X \geq (1+\delta)\mu] \leq 2^{-\Omega_\delta(\min\{k,\mu\})}$.*

H-Coefficient Technique. In this section, we describe the *H-coefficient technique* [Pat08], that will be useful for analyzing the best advantage of an *adaptive* adversary in distinguishing between real game G_0 and an ideal game G_1. We denote $T(G_0)$ and $T(G_1)$ to be the distribution of transcripts that are communicated between the challenger and adversary in the corresponding game.

Now, imagine we can partition the set of transcripts into two disjoint sets, called "good" and "bad" transcripts. Critically, this partition function *can use some private randomness known by the challenger.* For good transcripts, imagine we can prove a lower bound $(1-\mu)$ on the ratio of the probability of any fixed good transcript τ appearing in the real game G_0 compared to the probability of τ appearing in the ideal game G_1. Notice, since the entire transcript is fixed, we do not care if the adversary is adaptive in computing the corresponding probabilities in this experiment. In contrast, bad transcripts are typically chosen where a lower bound on the ratio cannot be proven. Instead, we derive an upper bound ε on the probability that any bad transcript τ will ever appear. Furthermore, this bound will be done using the *ideal experiment*, which typically simplifies the proofs, since adaptivity of the attacker is usually not a problem in the ideal world. The H-coefficient technique states that, if both bounds above can be successfully computed, then the best distinguishing advantage of an *adaptive* adversary is upper bounded by $\mu + \varepsilon$.

Lemma 2.12 (H-Coefficient Technique [Pat08]). *For two experiments G_0 and G_1, if there exists a set of (so called "good") transcripts $\mathcal{T}$ (possibly dependent on the randomness of the challenger) and values $\varepsilon, \mu \geq 0$ satisfying*

1. *$\Pr[T(G_0) = \tau] / \Pr[T(G_1) = \tau] \geq 1 - \mu$ for all $\tau \in \mathcal{T}$;*
2. *$\Pr[T(G_1) \notin \mathcal{T}] \leq \varepsilon$;*

then the best advantage for any adaptive adversary trying to distinguish G_0 from G_1 is at most $\mu + \varepsilon$.

3 Definitions of Independent Local Hashing

Independent Hashing and Locality. We first define perfect and almost k-wise independent hash functions. Then we define what it means for them to be local.

Definition 3.1 (perfect and almost k-wise independent hashing). *Let $\mathcal{F} = \{f_s : \{0,1\}^n \to \{0,1\}^m\}_{s \in \Sigma^\ell}$ be a function family over some word alphabet Σ. We say that $\mathcal{F}$ is* perfectly k-wise independent *if for all distinct $x_1, \ldots, x_k \in \{0,1\}^n$ and for any $y_1, \ldots, y_k \in \{0,1\}^m$ we have $\Pr_{s \leftarrow \Sigma^\ell}[f_s(x_1) = y_1 \wedge \cdots \wedge f_s(x_k) = y_k] = 2^{-mk}$.*

*We say that $\mathcal{F}$ is ε-*almost k-wise independent *if for any distinguisher D that makes at most k adaptive queries to its oracle we have $|\Pr_{s \leftarrow \Sigma^\ell}[D^{f_s(\cdot)} = 1] - \Pr_R[D^{R(\cdot)} = 1]| \leq \varepsilon$, where $R : \{0,1\}^n \to \{0,1\}^m$ is a uniformly random function.*

Note that the above default definition of almost k-wise independence is strong and considers *adaptive distinguishers* making adaptive queries to its oracle. One could also consider a weaker *selectively secure* notion where the k queries all need to be chosen by the distinguisher upfront. We will only consider the selective security notion for our lower bounds, which makes them stronger.

Definition 3.2 (local functions). *Let $\mathcal{F} = \{f_s : \{0,1\}^n \to \{0,1\}^m\}_{s \in \Sigma^\ell}$ be a function family. We say $\mathcal{F}$ is t-word-local if for all s, x the evaluation of $f_s(x)$ only reads at most t locations in s. When $\Sigma = \{0,1\}$ we say that $\mathcal{F}$ is t-bit-local.*

We say that $\mathcal{F}$ has *static access* if the location of s accessed by $f_s(x)$ only depend on x and not on s. Otherwise we say that it has *adaptive access*.

We say that a t-word-local (or t-bit-local) hash function family $\mathcal{F}$ is *explicit* if $f_s(x)$ can be computed in time in $\mathsf{poly}(n, m, t, w, \log \ell)$ given oracle access to the key s.

4 Perfectly Independent Local Hashing

4.1 Construction from Lossless Expanders

Overview. We construct a family $\mathcal{F}$ of k-wise perfectly independent hash functions $f_s : \{0,1\}^n \to \{0,1\}^m$ indexed by a seed $s \in \{0,1\}^\ell$. The first part of the construction is based on bipartite expander graphs: we use them to map an input $x \in \{0,1\}^n$ into a subset $I(x) \subset [\ell]$ of size (at most) d. The expansion property guarantees that any k distinct inputs are mapped to about kd distinct indices. Let $s_{I(x)} \in \{0,1\}^d$ denote the part of the seed indexed by x, padded with 0s if $|I(x)| < d$. The output of the hash function is $f_s(x) = \mathbf{A}s_{I(x)}$, where $\mathbf{A} \in \mathbb{F}_2^{m \times d}$ is a matrix with the property that any subset of its columns of size close to d has full span. We call such matrices *rank-robust* matrices, and show that they are equivalent to the generator matrices of good enough error correcting codes, and using known constructions one can take $d = O(m)$. The overall proof then relies on the expansion guarantees of the expander graphs and elementary linear algebra, and obtains optimal locality of $d = O(m)$ bits. (The work of Siegel [Sie89] relies on a related but somewhat different method of combining expanders with a Vandermonde matrix to construct a hash family, but the resulting construction does not achieve optimal parameters even in the word model.)

Rank-Robust Matrices. We start by definiting rank-robust matrices.

Definition 4.1 (rank-robust matrix)**.** *Let $m \le r \le d$. We say that a matrix $\mathbf{A} \in \mathbb{F}_2^{m \times d}$ is r-rank-robust if every r columns of $\mathbf{A}$ have rank m.*

A matrix $\mathbf{A} \in \mathbb{F}_2^{m \times d}$ is r-rank-robust iff for all $x \in \mathbb{F}_2^m$, the vector $x\mathbf{A}$ has 0s in at most $r-1$ positions. Therefore, $\mathbf{A}$ being r-rank-robust is equivalent to $\mathbf{A}$ being the generator matrix of a linear code over $\mathbb{F}_2$ with message length m, codeword length d and distance $d - r + 1$. For any constant $\delta < 1/2$, there are explicit constructions of such rank-robust matrices (computable in $\mathsf{poly}(m, d)$-time) for any $r > (1-\delta)d$ with $m = \Omega_\delta(d)$, see for example [Ta-17] for the best known dependence on δ (we would apply it for a fixed constant δ, so the specific dependence would not matter for us).

Construction. Let $G : [U] \times [d] \to [\ell]$ be a (k, ε)-lossless expander graph with $U = 2^n$. Let $\mathbf{A} \in \mathbb{F}_2^{m \times d}$ be r-robust with $r = (1 - 2\varepsilon)d$. Define the function family $\mathcal{F} = \{f_s : \{0,1\}^n \to \{0,1\}^m\}_{s \in \{0,1\}^\ell}$ as follows:

- On input $x \in \{0,1\}^n$, which we identify as a value in $[U]$, let $I(x) := \{G(x, i) : i \in [d]\} \subseteq [\ell]$ denote the neighbors of x.
- Let $s_{I(x)} \in \mathbb{F}_2^d$ denote the values of s in positions $I(x)$, padded with 0s if $I(x) < d$.
- Output $\mathbf{A}s_{I(x)} \in \mathbb{F}_2^m$.

It is easy to see that this function family is d-bit-local.

Theorem 4.2. *The function family $\mathcal{F}$ is perfectly k-wise independent.*

Proof. For any x, the function $f_s(x)$ is linear in $s \in \mathbb{F}_2^\ell$, and therefore we can write $f_s(x) = \mathbf{A}_x s$ for the matrix $\mathbf{A}_x \in \mathbb{F}_2^{m \times \ell}$ that has the columns of $\mathbf{A}$ in the $\le d$ positions selected by $I(x)$ and 0s elsewhere. To show k-wise independence, it suffices to show that for any subset $X \subseteq \{0,1\}^n$ with $|X| \le k$ the matrix $\mathbf{A}_X \in \mathbb{F}_2^{m|X| \times \ell}$, defined by stacking the matrices $\mathbf{A}_x$ for $x \in X$ on top of each other, has rank $m|X|$. We do so by induction on $|X|$. It is true for $|X| = 1$ since, for any x, expansion tells us that $|I(x)| \ge (1-\varepsilon)d \ge r$ and r-rank-robustness tells us the that the the selected r columns of $\mathbf{A}$ are full rank. To show that it holds

for all $|X| \leq k$, the expansion property tells us that the neighborhood of X, denoted by $I(X) := \{G(x,i) : x \in X, i \in [d]\} = \cup_{x \in X} I(x)$, is of size $|I(X)| \geq (1-\varepsilon)d|X|$. This means that at least $(1-2\varepsilon)d|X|$ of the values in $I(X)$ are unique, meaning that they appear in a single set $I(x)$, and hence there exists some $x \in X$ that has at least $(1-2\varepsilon)d$ unique neighbors – i.e., for $X' = X \setminus \{x\}$ we have $|I(x) \setminus I(X')| \geq (1-2\varepsilon)d \geq r$. By induction, we can assume $\mathbf{A}_{X'}$ has rank $(|X|-1)m$ and by r-rank-robustness $\mathbf{A}_x$ has rank m even when restricted to just the columns in which $\mathbf{A}_{X'}$ is 0s. Therefore $\mathbf{A}_X$ must have rank $m|X|$. □

Existential Parameters. Plugging in the existential parameters for expanders (Lemma 2.4) we get the following result.

Corollary 4.3. *There exist perfectly k-wise-independent t-bit-local functions $\mathcal{F} = \{f_s : \{0,1\}^n \to \{0,1\}^m\}_{s \in \{0,1\}^\ell}$ with key size $\ell = O(k(n+m))$ bits and locality $t = O(n+m)$.*

Proof. Use (e.g.,) a $(k, \varepsilon = .2)$-lossless expander $G : [2^n] \times [d] \to [\ell]$ with and $\ell = O(kd)$. Use a r-rank-robust matrix $\mathbf{A} \in \mathbb{F}_2^{m \times d}$ with $r = .6d$. This can be done simultaneously with a sufficiently large $d = O(n+m)$ via Lemma 2.4. □

Explicit Parameters. Plugging in the best known explicit parameters for expanders (Lemma 2.5) we get the following result.

Corollary 4.4. *For any constant $\alpha > 0$, there exist* explicit *perfectly k-wise-independent t-bit-local functions $\mathcal{F} = \{f_s : \{0,1\}^n \to \{0,1\}^m\}_{s \in \{0,1\}^\ell}$ with locality $t = O\left(n(\log k)\right)^{1+1/\alpha} + O(m)$ and key size is $\ell = O(t^2 \cdot k^{1+\alpha})$.*

Proof. Use (e.g.,) an explicit $(k, \varepsilon = .2)$-lossless expander $G : [2^n] \times [d] \to [\ell]$ and an r-rank-robust matrix $\mathbf{A} \in \mathbb{F}_2^{m \times d}$ with $r = .6d$. This can be done simultaneously with a sufficiently large $d = O\left(n(\log k)\right)^{1+1/\alpha} + O(m)$ with $\ell = O(t^2 \cdot k^{1+\alpha})$ via Lemma 2.5. □

4.2 Perfect Local Hashing Implies Expanders

Overview. We show that any explicit perfectly k-wise independent t-local hash family with input and output size n and key size ℓ in the bit model yields a $(k, A = n)$-expander graph $G : [2^n] \times [t] \to [\ell]$. For hash functions with static access, where the locations of the key accessed by $f_s(x)$ for each input x do not depend on s, this is easy to see. We can define the "access graph" G where the left vertices denote the inputs $x \in \{0,1\}^n$, the right vertices the locations $[\ell]$ of the key, and $G(x,i)$ just the i'th location accessed by the computation $f_s(x)$. If the graph is not expanding then there is some subset $X \subseteq \{0,1\}^n$ of size $|X| \leq k$ such that the evaluations of $f_s(x)$ for $x \in X$ depend on fewer than $|X|n$ bits of the key and hence the outputs cannot be uniformly random. We extend this result to hash functions with adaptive access. In this setting we have a different access graph G_s for each key s, and we show that each such graph is a good expander. This may not be obvious at first – perhaps for some small fraction of keys s the graphs G_s are not expanding and the outputs don't have enough entropy conditioned on this occurring, but it happens with low enough probability to perfectly balance out and the overall output distribution remains uniform. We show this is not the case.

Formal Statement and Proof. We show the following general theorem for arbitrary alphabets Σ and input sizes n and output sizes m. We then provide corollaries for particular parameters fo interest.

Theorem 4.5. *Let $\mathcal{F} = \{f_s : \{0,1\}^n \to \{0,1\}^m\}_{s \in \Sigma^\ell}$ be a perfectly k-wise independent t-word-local hash with word alphabet $\Sigma = \{0,1\}^w$. For every s define the graph $G_s : [2^n] \times [t] \to [\ell]$ such that $G_s(x,i)$ is the i'th location accessed by the computation $f_s(x)$. Then for every s, the graph G_s is a $(k, A = m/w)$-expander.*

Proof. Fix any $X = \{x_1, \ldots, x_k\} \subseteq \{0,1\}^n$. Let

$$\delta := \Pr_s[\ |I(X)| < kA \text{ in the graph } G_s\ ].$$

We show that $\delta = 0$, which proves the lemma.

Consider the process of sampling $f_s(X) = (f_s(x_1), \ldots, f_s(x_k))$ for $s \leftarrow \Sigma^\ell$ via "lazy sampling" where, each time the computation touches a new location of s, we sample a fresh random word in Σ for it. Let $s^* \in \Sigma^{\leq kt}$ be the corresponding variable-length sequence of sampled words. Let $\mathcal{Y}^*$ be the set of values $Y \in \{0,1\}^{mk}$ that can be "lazy sampled" as above using some sequence s^* of length $|s^*| < kA$. Since the first $kA - 1$ words of s^* determine whether the computation needs to read more words or not, there are at most $\delta 2^{w(kA-1)}$ sequences s^* of length $|s^*| \leq kA - 1$ for which the lazy sampling terminates and therefore $|\mathcal{Y}^*| \leq \delta 2^{w(kA-1)}$. This shows:

$$\Pr_{s \leftarrow \{0,1\}^\ell}[f_s(X) \in \mathcal{Y}^*] = \delta$$
$$\Pr_{Y \leftarrow \{0,1\}^{km}}[Y \in \mathcal{Y}^*] = |\mathcal{Y}^*|/2^{km} \leq \delta \underbrace{2^{w(kA-1)}/2^{km}}_{\rho} \leq \delta\rho$$

for $\rho < 1$. Therefore the statistical distance between $f_s(X)$ for $s \leftarrow \{0,1\}^\ell$ and uniform is $\geq (1-\rho)\delta$. But since the function is perfectly k-wise independent, this must mean $\delta = 0$. □

Corollary 4.6. *We get the following results:*

- *Let $\mathcal{F} = \{f_s : \{0,1\}^n \to \{0,1\}^n\}_{s \in \Sigma^\ell}$ be a perfectly k-wise independent t-word-local hash with word alphabet $\Sigma = \{0,1\}^w$ and with $k, \ell = \mathsf{poly}(n)$ and $k \geq 2tw/n$. Then $t = \Omega(n/\log n)$. If furthermore the storage is just $\ell = O(kn/w)$ then $t = \Omega(n)$.*
- *Given an explicit $\mathcal{F} = \{f_s : \{0,1\}^n \to \{0,1\}^n\}_{s \in \{0,1\}^\ell}$ that is perfectly k-wise independent and t-bit-local for $t = O(n)$ and $\ell = O(kn)$, we can construct an explicit (k, ε)-lossless expanders $G : [U] \times [D] \to [V]$ for some constant $\varepsilon < 1$, with $D = O(\log U)$ and $V = O(k \log U)$.*
- *Given an explicit $\mathcal{F} = \{f_s : \{0,1\}^n \to \{0,1\}^n\}_{s \in \Sigma^\ell}$ over the alphabet $\Sigma = \{0,1\}^w$ that is perfectly k-wise independent and t-word-local for $t = O(n)$ and $\ell = O(kn/w)$, we can construct an explicit $(k, A = n/w)$-expander $G : [U] \times [D] \to [V]$, with $D = O(\log U)$ and $V = O(k \log U/w)$.*

Proof. The first part follows by combining our theorem that perfect k-wise independence yields corresponding expanders with the expander lower bounds from Lemma 2.6. The other two parts just frollow from the theorem with concrete parameter choices. □

The first part of the corollary essentially re-derives Siegel's lower bound [Sie89, Sie04], but extends it to arbitrary alphabet sizes. It shows that for $n = m$, if we have optimal key size $\ell = O(km/w)$ words then we cannot get better locality than the $O(n)$ even if we consider arbitrarily large word size.

The second two parts of the corollary tell us that any progress in explicit perfect k-wise independent hashing requires corresponding progress in expanders. Note that this does *not* generalize to ε-almost k-wise independent hashing and does *not* even imply that in that case the graph G_s would be a good expander with high probability over s. It does show that for any fixed set X, with high probability the graph G_s is expanding on X with $I(X) \geq km$, but it may never hold that G_s is simultanously expanding for all X. In other words, this leaves open the possibility of constructing explicit ε-statistical k-wise independent hashing with optimal parameters without corresponding improvements in explicit expanders (or even in efficiently sampling a good expander with high probability).

5 Almost Independent Local Hashing

5.1 Lower Bound

Overview. Below we give a lower bound on the locality of ε-almost-independent hash functions. It shows that in the case of input/output size $n = m$ and $\varepsilon = 2^{-n}$, for word size $w = O(1)$ the locality must be at least $t = \widetilde{\Omega}(n)$, and for word size $w = \Theta(n)$ the locality must be at least $t = \widetilde{\Omega}(\sqrt{n})$. When the word size

is allowed to be sufficiently large $w = \Theta(n^2)$ the bound becomes trivial and allows for constant $t = O(1)$. Surprisingly, we will see these bounds are tight and we have matching upper bounds! This is contrast to perfect k-wide independence where we have a lower bound of $t = \Omega(n/\log n)$ even for arbitrarily large word size $w = \mathsf{poly}(n)$ (See the first part of Corollary 4.6). In all cases, these bound naturally only kick in for sufficiently large $k > tw/n$; otherwise we could simply store an entire non-local perfect k-wise independent hash inside t words.

At a high level, we construct a statistical distinguisher that chooses $d \leq k$ random inputs x_i and checks if the oracle outputs y_i can all be explained as evaluations of $f_s(x_i)$ for some key s such that all d evaluation only touch the same t locations of s. We show that if the locality t is small enough and d chosen carefully then this happens with sufficiently high probability given opracle access to f_s but small probability given oracle access to a random function and therefor gives too large of a distinguishing advantage.

Theorem 5.1. *Let $\mathcal{F} = \{f_s : \{0,1\}^n \to \{0,1\}^m\}_{s \in \Sigma^\ell}$ be an ε-almost k-wise independent t-word-local hash with word alphabet $\Sigma = \{0,1\}^w$. If $k > (tw + \log(1/\varepsilon))/m$ then $t \geq \Omega\left(\min\left\{m/\log \ell, \sqrt{\frac{m \log 1/\varepsilon}{w \log \ell}}\right\}\right)$. This holds even if $\mathcal{F}$ only satisfies the weaker notion of selective security.*

In particular, when $m = n$, $\ell = \mathsf{poly}(n)$, $\varepsilon = 2^{-n}$ then for $k > tw/n + 1$ we have $t \geq \Omega\left(n/(\sqrt{w} \log n)\right)$.

Proof. If $t \geq m/(2\log(e\ell))$ then we are done, so for the remainder of the proof assume otherwise that $t < m/(2\log(e\ell))$. We show that $t \geq \Omega\left(\sqrt{\frac{m \log 1/\varepsilon}{w \log \ell}}\right)$.

Consider the distinguisher D that chooses $d = \lceil (tw + \log(1/\varepsilon))/m \rceil$ random distinct points $X = (x_1, \ldots, x_d) \in \{0,1\}^{nd}$. It queries the oracle on the points X and gets back values $Y = (y_1, \ldots, y_d) \in \{0,1\}^{md}$. The distinguisher checks if there exists some $s \in \Sigma^\ell$ such that $f_s(x_i) = y_i$ for all $i \in [d]$ *and* the d evaluations overall only touch $\leq t$ words of s. If so it outputs 1 else it outputs 0.

Fix any $s \in \Sigma^\ell$. Since $\mathcal{F}$ is t-word local, we can partition the 2^n inputs $x \in \{0,1\}^n$ into $\binom{\ell}{t}$ parts according to the subset of t locations that the computation $f_s(x)$ reads. There must be some part of size at least $2^n/\binom{\ell}{t}$, and therefore $\Pr[D^{f_s(\cdot)} = 1]$ is at least the probability that $x_1, \ldots, x_d$ all fall into the same part, which is:

$$\Pr[D^{f_s(\cdot)} = 1] \geq \binom{2^n/\binom{\ell}{t}}{d} \binom{2^n}{d}^{-1} \geq \left(\frac{1}{e\ell}\right)^{td} \tag{1}$$

On the other hand, for any choice of $x_1, \ldots, x_d$, the number of values $Y = (y_1, \ldots, y_d) \in \{0,1\}^{md}$ for which D outputs 1 is at most 2^{wt} (i.e., such values can be "lazy sampled" using t words as in the proof of Theorem 4.5) and hence:

$$\Pr[D^{R(\cdot)} = 1] \leq 2^{wt - md} \leq \varepsilon \tag{2}$$

Since $k \geq d$ the ε-almost k-wise independence property requires that $|\Pr[D^{f_s(\cdot)} = 1] - \Pr[D^{R(\cdot)} = 1]| \leq \varepsilon$ and hence $\left(\frac{1}{e\ell}\right)^{td} \leq 2\varepsilon$. By taking logarithms, plugging in our choice of d and relying on $t < m/(2\log(e\ell))$ we get:

$$\begin{aligned}
\left(\frac{1}{e\ell}\right)^{td} \leq 2\varepsilon \quad &\Rightarrow \quad td\log(e\ell) \geq \log(1/\varepsilon) - 1 \\
&\Rightarrow \quad t(tw/m + \log(1/\varepsilon)/m)\log(e\ell) \geq \log(1/\varepsilon) - 1 \\
&\Rightarrow \quad (t^2 w/m)\log(e\ell) \geq \log(1/\varepsilon)/2 - 1 \\
&\Rightarrow \quad t \geq \Omega\left(\sqrt{\frac{m \log 1/\varepsilon}{w \log \ell}}\right)
\end{aligned}$$

as we wanted to show. □

When m is small, for example $m = 1$, the above lower bound becomes trivial. Below we give yet another (incomparable) lower bound that also holds for small m, showing that the locality cannot be too small. For example in the bit model with security $\varepsilon = 2^{-n}$ it shows that the locality needs to be at least $n/\log n$.

Theorem 5.2. *Let $\mathcal{F} = \{f_s : \{0,1\}^n \to \{0,1\}^m\}_{s\in\Sigma^\ell}$ be an ε-almost k-wise independent t-word-local hash with word alphabet $\Sigma = \{0,1\}^w$. Then $t \geq \min\{\frac{n-\log k}{\log \ell}, \log((\varepsilon + 2^{-km})^{-1})/w\}$. In particular if $k = \mathsf{poly}(n), \ell = \mathsf{poly}(n)$ and $\log(1/\varepsilon) = O(km)$ then $t \geq \Omega(\min\{n/\log n, \log(1/\varepsilon)/w\})$. This holds even if $\mathcal{F}$ only satisfies the weaker notion of selective security.*

Proof. Assume that $t \leq \frac{n-\log k}{\log \ell}$, or equivalently $k\ell^t \leq 2^n$. We show that then $t \geq \log((\varepsilon + 2^{-km})^{-1})/w$ or equivalently, $2^{-wt} - 2^{-km} \leq \varepsilon$.

Let $s^* \in \Sigma^\ell$ be some fixed key, say all 0s. Define $X_0 := \{0,1\}^n$ and for $i = 1, \ldots, t$ we define sets X_i such that $X_i \geq 2^n/\ell^i$ and the evaluation of $f_{s^*}(x)$ accesses the same first i locations for all $x \in X_i$. We do so as follows:

- For $x \in X_{i-1}$ let $j_x \in [\ell]$ be the i'th key location accessed during the evaluation of $f_{s^*}(x)$.
- For $j \in [\ell]$ let $X_{i,j} := \{x \in X_{i-1} : j_x = j\}$ be the subset of values X_{i-1} for which the ith location accessed is j. Since these sets partition X_{i-1}, there must be some j^* such that $|X_{i,j^*}| \geq |X_{i-1}/\ell$. Define $X_i := X_{i.j^*}$.

Finally $|X_t| \geq 2^n/\ell^t \geq k$. By construction there is some set $T \subseteq [\ell]$ of size $|T| \leq t$ such that, for all $x \in X_t$, the evaluation of $f_{s^*}(x)$ only accesses the key in locations T, denotes by S^*T. Take any distinct values $x_1, \ldots, x_k \in X_t$. Define the selective distinguisher D that queries its oracle on $x_1, \ldots, x_k$, receives $y_1, \ldots, y_k$ and outputs 1 if $y_i = f_{s^*}(x_i)$. We have

$$\Pr_s[D^{f_s} = 1] \geq \Pr[s_T = s^*_T] = 2^{-wt}$$
$$\Pr[D^R = 1] \leq 2^{-km}$$

and therefore the securiy of $\mathcal{F}$ rquires $2^{-wt} - 2^{-km} \leq \varepsilon$ as we wanted to show. □

5.2 Optimal Bit-Local Construction

Overview. While we give our statistical construction for general m, n and ε, below we sketch the key idea for the most important case $m = \log(1/\varepsilon) = n$. We will set seed length $\ell = O(kn)$ and design special, keyed hash function h mapping n-bits strings x into d-element subsets of $[\ell]$, for $d = O(n)$. The function h will be sampled from a new type of hash family we call *overlap-avoiding*, whose construction (see Lemma 5.4) will form the main technical contribution of this section. For our concrete parameters, it would mean that for any distinct inputs $x_1, \ldots, x_k$, the probability that the union of sets $h(x_1), \ldots, h(x_{k-1})$ covers more than 3/4 of $h(x_k)$ is at most our desired error 2^{-n}. Critically, we will also ensure that h is efficient and has $O(n)$-bit description, as h will be part of the seed and always contribute to locality, for any input x.

Now, the overall seed s for f_s will consist of an overlap-avoiding hash h, and a random ℓ-bit string R. The evaluate our function $f_s(x)$ one first computes subset $h(x)$ of size d, reads the corresponding bits z from R, and finally applied $3d/4$-robust matrix $\mathbf{A}$ (see Definition 4.1) to z to output $y = \mathbf{A}z$. Intuitively, overlap-avoidance of h will guarantee that each substrings z_i among k substrings $z_1, \ldots, z_k$ of R (read when evaluating $f_s(x_1), \ldots, f_s(x_k)$) will have 3/4 fresh bits, completely untouched by other evaluations. After which the robustness of $\mathbf{A}$ will show perfect independence of $y_1 \ldots, y_k$ (modulo statistical error caused by the overlap-avoidance failure).

Overlap Avoiding Hashing. We first formally define overlap avoiding hash functions.

Definition 5.3 (Overlap Avoiding)**.** *Let $C_{\ell,d} = \{S \subseteq [\ell] : |S| = d\}$. Consider a hash family $\mathcal{H}$ consisting of functions $h : \{0,1\}^n \to C_{\ell,d}$ that output sets. We say $\mathcal{H}$ is (k, δ, ε)-overlap avoiding if for any $x_1, \ldots, x_k \in \{0,1\}^n$:*

$$\Pr_{h\leftarrow\mathcal{H}}[\ |\ h(x_k)\ \setminus\ (h(x_1) \cup \cdots \cup h(x_{k-1}))\ |\ \geq (1-\delta)d] \geq 1 - \varepsilon.$$

The next lemma gives an explicit construction of overlap avoiding hash functions. The construction is based on a combination of pairwise independent hash functions and averaging samplers.

Lemma 5.4. *Let $n, d, k \geq 1$ and $\varepsilon, \delta \in (0,1)$, and assume $d = \Omega((\log 1/\varepsilon)/\delta^2)$. There exists an explicit construction of a (k, δ, ε)-overlap avoiding hash family $\mathcal{H}$ consisting of functions $h : \{0,1\}^n \to C_{\ell,d}$, with $\ell = O(dk/\delta)$. Furthermore, the hash functions have description length $\log |\mathcal{H}| = n + \log(k) + O(d \log(1/\delta))$.*

Proof. Setting parameters, let $r \geq k/2\delta$ be a power of two and $\ell = rd$.

First, let $\mathcal{F} = \{f_v : \{0,1\}^n \to [r]\}_{v \in [V]}$ be a family of pairwise independent hash functions. There are standard explicit constructions for such a family of size $V = O(2^n r)$ [3]. For any distinct $x_1, \ldots, x_k \in \{0,1\}^n$ we have by definition

$$\begin{aligned} & \Pr_{v \in [V]}[f_v(x_k) \in \{f_v(x_1), \ldots, f_v(x_{k-1})\}] \\ \leq & \sum_{i=1}^{k-1} \Pr_{v \in [V]}[f_v(x_k) = f_v(x_i)] \\ = & (k-1)/r \leq \delta/2. \end{aligned}$$

Next, let $\mathsf{Samp} : [U] \times [d] \to [V]$ be an averaging sampler with accuracy $\delta/2$ and error ε (here we do not need the condition of distinct elements). Applying Theorem 2.9 and using our assumption that $d = \Omega((\log 1/\varepsilon)/\delta^2)$, gives an explicit and efficient construction with $U \leq V(1/\delta)^{O(d)}$.

We now combine both to construct the hash family $\mathcal{H}$. First, for $u \in [U], i \in [d]$ define

$$g_{u,i}(x) = f_{\mathsf{Samp}(u,i)}(x) + r(i-1).$$

Note that $g_{u,i}(x) \in [r(i-1)+1, ri]$, and in particular that the ranges of $g_{u,i}$ for distinct i are disjoint. Next, define $h_u(x) = \{g_{u,1}(x), \ldots, g_{u,d}(x)\} \in C_{\ell,d}$ and set $\mathcal{H} = \{h_u : u \in [U]\}$.

To conclude, we show that $\mathcal{H}$ satisfies the required conditions, and bound its size. Fix distinct $x_1, \ldots, x_k \in \{0,1\}^n$. Define the set of "bad inner seeds" for $x_1, \ldots, x_k$ as:

$$B = \{v \in [V] : f_v(x_k) \in \{f_v(x_1), \ldots, f_v(x_{k-1})\}\}.$$

As we saw, $\frac{|B|}{V} = \Pr_{v \in [V]}[v \in B] \leq \delta/2$. Given $u \in [U]$, define

$$p(u) = \Pr_{i \in [d]}[\mathsf{Samp}(x,i) \in B].$$

Note that $p(u)$ measures that fraction of $i \in [d]$ for which $g_{u,i}(x_k) \in \{g_{u,i}(x_1), \ldots, g_{u,i}(x_{k-1})\}$. Since the ranges of $g_{u,i}$ for distinct i are disjoint, we have

$$|h_u(x_k) \setminus (h_u(x_1) \cup \cdots \cup h_u(x_{k-1}))| = (1 - p(u))d.$$

As Samp is a sampler with accuracy $\delta/2$ and error ε, this concludes the proof since

$$\Pr_{u \in [U]}[p(u) \geq \delta] \leq \Pr_{u \in [U]}\left[\left|\Pr_{i \in [d]}[\mathsf{Samp}(x,i) \in B] - \frac{|B|}{V}\right| \geq \delta/2\right] \leq \varepsilon.$$

Finally, we bound the family size as

$$|\mathcal{H}| = U \leq V(1/\delta)^{O(d)} \leq 2^n k (1/\delta)^{O(d)}.$$

□

[3] Since r is a power of two there is a finite field $\mathbb{F}_r$. Choose minimal m so that $r^{m-1} > 2^n$. Let $u_1, \ldots, u_{2^n} \in \mathbb{F}_r^m$ be distinct 0nonzero vectors, such that no two are a multiple of each other. Sample $v \in \mathbb{F}_r^m$ and for $x \in [2^n]$ define $f_v(x) = \langle u_x, v \rangle$.

From overlap avoiding to k-wise independence. Before giving the construction of ε-almost k-wise independence hash functions with small locality from overlap avoiding families, we recall that the following well-known property (e.g., [CGH+85]) of r-robust matrices $\mathbf{A} \in \mathbb{F}_2^{m \times d}$ from Definition 4.1. Given a set $V \subset [d]$ of size r and a string $z \in \{0,1\}^d$, let $z|_V$ denote the r-bit restriction of z to its bits z_i for $i \in V$.

Lemma 5.5 (Theorem 2 in [CGH+85])**.** *Assume $\mathbf{A} \in \mathbb{F}_2^{m \times d}$ is an r-robust matrix, $V \subset [d]$ is a set of size r, and Z is any distribution whose r bits $Z|_V$ are uniformly random and independent of the remaining bits $E = Z|_{[d]\setminus V}$. Then the distribution $Y = \mathbf{A}Z$ is (perfectly) uniform over $\{0,1\}^m$, even conditioned on E.*[4]

We now define our hash function family $\mathcal{F} = \{f_s \ : \ \{0,1\}^n \to \{0,1\}^m\}_{s \in \{0,1\}^\ell}$ as follows, from any $(k, \delta, \varepsilon_0)$-overlap-avoiding family $\mathcal{H}$ over $C_{\ell_0,d}$, and an r-robust matrix $\mathbf{A} \in \mathbb{F}_2^{m \times d}$ for $r = (1-\delta)d$.

- The key $s = (R, h)$ consists of randomness $R \in \{0,1\}^{\ell_0}$, and an overlap avoiding hash $h \leftarrow \mathcal{H}$. The total key length is $\ell = |s| = \ell_0 + |h|$.
- Given input $x \in \{0,1\}^n$, compute $S = h(x) \in C_{\ell_0,d}$, and let $z = R|_S$. Then output $y = \mathbf{A}z$. The total bit-locality is $t = d + |h|$.

Lemma 5.6. *If $\mathcal{H}$ is $(k, \delta, \varepsilon_0)$-overlap-avoiding over $C_{\ell_0,d}$, and $\mathbf{A} \in \mathbb{F}_2^{m \times d}$ is $(1-\delta)d$-robust, then $\mathcal{F}$ is ε-almost and k-wise independent family from n to m bits, with key length $\ell = \ell_0 + \log|\mathcal{H}|$, locality $t = d + \log|\mathcal{H}|$ and error $\varepsilon = k\varepsilon_0$.*

Proof. We start with a very simple selective proof first, before arguing that we can extend it to the general adaptive case. Consider any fixed inputs $x_1, \ldots, x_k$, and let $S_i = h(x_i)$ and $Z_i = R|_{S_i}$. By the overlap-avoiding property applied k times, except with probability $\varepsilon = \varepsilon_0 k$, each Z_i has $(1-\delta)d$ perfectly uniform random bits, even if we condition on the $Z_{-i} = Z_1 \ldots Z_{i-1}, Z_{i+1}, \ldots, Z_k$. Conditioned on this event, by Lemma 5.5, the k outputs $y_i = f_s(x_i) = \mathbf{A}Z_i$ are perfectly uniform and independent of each other.

We now show that the same proof template extends to the fully adaptive case, using the H-coefficient technique from Lemma 2.12. Consider an adversary D that adaptively selects k values x_i and gets the outputs y_i one by one. In the real game G_0, these outputs are computed as $y_i = f_s(x_i)$, while in the ideal game G_1 these values are random m-bit strings. We now define the set of good transcripts $\mathcal{T}$, which recall can depend on the private randomness of the challenger; in our case, the hash function h. We say that the transcript $\tau = (x_1, y_1, \ldots, x_k, y_k)$ is good, if all k inputs x_i satisfy the overlap-avoiding property:

$$| \ h(x_i) \ \setminus \ (h(x_1) \cup \cdots \cup h(x_{i-1}) \cup h(x_{i+1}) \cdots \cup h(x_k)) \ | \ \geq (1-\delta)d \tag{3}$$

We can now establish the two properties of good and bad transcripts needed for the H-coefficient lemma.

First, let us establish the upper bound on the probability of a bad transcript in the *ideal game*, when all the values y_i are random. In this case, D chooses all the values x_i independently of the hash function h. Thus, we can rely on overlap-avoiding propery of h to argue that for each $i \in [k]$, Equation (3) fails with probability at most ε_0. By union bound, any of them fails with probability at most $\varepsilon = k\varepsilon_0$.

Finally, we will argue that any good transcript τ (which satisfies Equation (3) for all the $i \in [k]$) happens with identical probability in both the real and the ideal games. This follows from the $(1-\delta)d$-resilience of $\mathbf{A}$ in Lemma 5.5, and the fact that Equation (3) holds for all i. Namely, the distribution of all y_i in the real game is perfectly uniform: the same as in the ideal game. This implies $\mu = 0$ in Lemma 2.12, and yields an overall distinguishing advantage $\varepsilon + \mu = \varepsilon = k\varepsilon_0$. □

Parameters. We now instantiate this result with the overlap-avoiding family $\mathcal{H}$ from Lemma 5.4, and the standard coding-based construction of robust matrices mentioned after Definition 4.1.[5] For simplicity of notation, we will lower bound the output length m by $\Omega(\log(kn/\varepsilon))$, as there is no advantage for locality to consider smaller values m, and one can always truncate the output size without affecting security.

[4] Stated differently, the function $\mathsf{RF}(Z) := \mathbf{A}Z$ is what is known as (perfectly) *r-resilient* [CGH+85]: as long of the input distribution Z has r truly random bits conditioned on the other bits of Z, the output $\mathsf{RF}(Z)$ is perfectly uniform conditioned on these bits.

[5] When $n \gg m + \log(k/\varepsilon)$, we additionally pre-hash our n-bit input using an $O(\varepsilon/k^2)$-almost universal hash function.

Theorem 5.7. *For any $k, n \geq 1$, $\varepsilon > 0$, and $m \geq \Omega(\log(kn/\varepsilon))$ there exists an efficient ε-almost k-wise independent t-bit-local hash family $\mathcal{F} = \{f_s : \{0,1\}^n \to \{0,1\}^m\}_{s\in\{0,1\}^\ell}$ with simultaneously optimal seed length $\ell = O(km)$ and bit-locality $t = O(m)$.*

In particular, we get an efficient, (2^{-n})-almost k-wise independent, t-bit-local hash family $\mathcal{F} = \{f_s : \{0,1\}^n \to \{0,1\}^n\}_{s\in\{0,1\}^\ell}$ with simultaneously optimal seed length $\ell = O(kn)$ and bit-locality $t = O(n)$.

Proof. First, the special case follows from the general one, if we set $\varepsilon = 2^{-n}$, $m = \Omega(n)$ large enough, which suffices because the maximum possible $k \leq 2^n$; than truncate the output to n bits.

Second, for the general case, we will assume without loss of generality that $n \leq O(m)$. Otherwise, we can first hash the input $x \in \{0,1\}^n$ to a smaller input x' of size $n' = \log(k^2 n/\varepsilon) = O(m)$ using a (ε/k^2)-universal hash function g (see Definition 2.1), and then apply our construction on the n'-bit hash x'.[6] The universality of g ensure that any two inputs $x_i \neq x_j$ collide with probability at most ε/k^2, implying that none of the k^2 pairs x_i and x_j requested by the distinguisher collide with probability at most ε. Short of this collision, statistical security of $\mathcal{F}$ is enough to guarantee security. The standard construction of (ε/k^2)-universal hash g in Lemma 2.2 for n-bit inputs has output size $n' = \log(k^2 n/\varepsilon) = O(m)$, and the same for the description of g. This means that our input reduction from n to n' bits only costs us $|g| = O(m)$ bits for both seed length and bit-locality.

Coming back to the main case $n \leq O(m)$, recall that binary robust matrices only exist for $\delta < 1/2$, so we will choose any such value; e.g., $\delta = 1/4$. In this case, this corresponds to a binary code of distance $1/4$, so it can have a constant rate; i.e., one can achieve $d = O(m)$ (say, $d = 10m$ using Justesen codes [Jus03]). Since we assumed $m \geq \Omega(\log(k/\varepsilon))$, this means $d = \Omega(\log(k/\varepsilon))$ too, and we can apply Lemma 5.4 to get $(k, 1/4, \varepsilon/k)$-overlap-avoiding hash family $\mathcal{H}$ over $C_{\ell_0,d}$ with universe size $\ell_0 = O(dk/\delta) = O(km)$, and bit-size description of an individual hash function $|h| = \log|\mathcal{H}| = n + \log k + O(d\log(1/\delta)) = O(m)$, since we assumed $n = O(m)$ and $m = \Omega(\log(k/\varepsilon))$. Plugging these values of d and $\log|\mathcal{H}|$ into Lemma 5.6, we get the parameters claimed in the theorem. □

Applications to Cryptography. As we mentioned in the Introduction, such almost k-wise independent hash function are sufficient for *locally computable*, unconditionally secure, k-time encryption and authentication with negligible security failure, and optimal key length ℓ and bit-locality t.

Concretely, if we let λ denote "security parameter", we get the following k-time secure message authentication scheme, for any $k = \mathsf{poly}(\lambda)$, input length $n = \mathsf{poly}(\lambda)$. We set the output length $m = \lambda + 1$, security $\varepsilon = 2^{-\lambda-1}$, and the tag of each message $\mathsf{msg} \in \{0,1\}^n$ is simply $f_s(\mathsf{msg})$. Using parameters in Theorem 5.7, our key length $\ell = O(k\lambda)$, locality $t = O(\lambda)$, tag size $(\lambda + 1)$, and overall forgery probability at most $(2^{-m} + \varepsilon) = 2^{-\lambda}$. These parameters are simultaneously optimal across all dimensions and were not known before.

For k-time encryption, we get similarly optimal parameters using the folklore randomized encryption scheme $(\mathsf{nonce}, f_s(\mathsf{nonce}) \oplus \mathsf{message})$, where the value nonce is chosen at random from some domain of size $(\lambda + 2\log k) = O(\lambda)$, to avoid collisions across the k-uses. Using parameters in Theorem 5.7 and assuming $|\mathsf{msg}| = \Omega(\lambda)$, our key length $\ell = O(k \cdot |\mathsf{msg}|)$, locality $t = O(|\mathsf{msg}|)$, message overhead $O(\lambda)$, and semantic security $\varepsilon = 2^{-\lambda}$. These parameters are simultaneously optimal across all dimensions. As mentioned in Footnote 2, such parameters were asymptotically known from prior work [Vad04, DY21] (utilizing the randomness of nonce to think of it as a seed for a "locally computable extractor"), but we believe our approach is conceptually simpler, and only requires that the nonce values do not repeat. For example, they work even in the setting of weak local randomness, when nonce values only have entropy.

5.3 Optimal Word-Local Construction

Overview. In the bit model, our construction used a special "overlap avoiding" hash $h(x)$ to select some subset $\Theta(n)$ key bits and then multiplied the selected bits by a robust matrix to covert them into the output. As long as the selected bits for x_k only had a small constant fraction overlap with those of the previous

[6] This folklore technique is often called "Levin's trick", and attributed to this paper [Lev86].

$x_1, \ldots, x_{k-1}$, each output was guaranteed to be random and independent of the previous ones. We essentially use the same strategy in the word model, but rely on a special "word-based overlap avoiding" hash where the selected bits all fall into a much smaller subset of words. To do so, we think of the $\ell = O(nk/w)$ word key as consisting of $d = \Theta(n/\sqrt{w})$ chunks of ℓ/d words each, and we select one word per chunk. Then, in each selected word, we sub-select $\sqrt{w}$ out of w bits. This gives us $\Theta(n)$ selected bits in total. We want to argue that the selected bits for x_k only have a small constant fraction overlap with the selected bits for $x_1, \ldots, x_{k-1}$ with probability $1 - 2^{-\Omega(n)}$. The main difficulty is how to do the selection to ensure this using only an $O(d)$ word selection key so we can read it entirely during each evaluation.

We do the selection by having d independent selection keys, one for each chunk. For each chunk $v \in [d]$, we first hash the input x into a smaller $\sqrt{w}$-bit digest $\hat{x}_v$ using a $2^{-\Omega(\sqrt{w})}$-almost universal hash. Then we use a $\sqrt{w}$-wise independent hash to map $\hat{x}_v$ to the word i_v that we select in this chunk. Lastly we use another $\sqrt{w}$-wise independent hash to map $\hat{x}_v$ to a sampler seed that we then use to sub-sample the $\sqrt{w}$ bits inside the selected words. This gives us the right parameters since a $\sqrt{w}$-wise independent hash over $\sqrt{w}$ bits can fit inside a single word! To analyze the failure probability of having too large overlap, we first argue that the word selected in chunk v for x_k was not selected too many times by previous inputs $x_1, \ldots, x_{k-1}$ and then argue that the selected bits in x_k are unlikely to have much overlap with previously selected ones. All of this happens for chunk v except with probability $2^{-\Omega(\sqrt{w})}$. Then we argue that except with probability $2^{-\Omega(d\sqrt{w})} = 2^{-\Omega(n)}$ the above happens for a large fraction of chunks $v \in [d]$. The analysis is somewhat subtle and relies on $\sqrt{w}$-wise independent hashing to be able to apply good concentration bounds for variables with bounded independence.

Word-Based Overlap-Avoiding Hash. Our construction relies on a generalization of overlap avoiding hashing to the word model. Previously, an overlap avoiding hash $h(x)$ selected a subset of d out of ℓ bits, such that for any $x_1, \ldots, x_k$ the bits selected by $h(x_k)$ avoided too much overlap with the bits selected by $h(x_1), \ldots, h(x_{k-1})$. Now we will think of having ℓw bits partitioned into ℓ words of w bits each. The hash $h(x)$ selects at most d out of ℓ words and p bits in each word, for a total of dp bits altogether. The goal is still the same: the bits selected by $h(x_k)$ should avoid too much overlap with the bits selected by $h(x_1), \ldots, h(x_{k-1})$. A word-based overlap avoiding hash also satisfies the previous bit-based definition, but the goal now is to concentrate the selected bits into as few words as possible.

Definition 5.8 (Word-Based Overlap Avoiding Hash)**.** *Let $C_{\ell,w,d,p}$ be the set of all subsets $S \subseteq [\ell] \times [w]$ such that:*

- *for $I := \{i \in [\ell] \ : \ \exists j \quad s.t. \ \ (i,j) \in S\}$ we have $|I| \leq d$, and*
- *for all $i \in I$ we have $|\{j \ : \ (i,j) \in S\}| = p$.*

Consider a hash family $\mathcal{H}$ consisting of functions $h : \{0,1\}^n \to C_{\ell,w,d,p}$. We say $\mathcal{H}$ is (k, δ, ε)-overlap avoiding if for any $x_1, \ldots, x_k \in \{0,1\}^n$:

$$\Pr_{h \leftarrow \mathcal{H}}[\ | \ h(x_k) \ \setminus \ (h(x_1) \cup \cdots \cup h(x_{k-1})) \ | \ \geq (1-\delta)dp] \geq 1 - \varepsilon.$$

Lemma 5.9. *For any constant $\delta \in (0,1)$ and any k, w, n such that $\Omega(\log^2 k + \log^2 n) \leq w \leq \min\{n^2, k^2\}$ there is an explicit construction of a (k, δ, ε)-word-based overlap avoiding family $\mathcal{H}$ of hash functions $h : \{0,1\}^n \to C_{\ell,w,d,p}$ with $d = \Theta(n/\sqrt{w})$, $p = \Theta(\sqrt{w})$, $\ell = O(kn/w)$, and $\varepsilon = 2^{-\Omega(n)}$. Furthermore, the hash functions have description length $\log |\mathcal{H}| = O(dw)$ bits, or equivalently, $O(d)$ words in $\Sigma = \{0,1\}^w$.*

Construction of Word-Based Overlap-Avoiding Hash. For a given k, w, n and δ, we construct the hash family $\mathcal{H}$ consisting of functions $h : \{0,1\}^n \to C_{\ell,w,d,p}$ with

$$\begin{aligned} d &:= (4/\delta)(n/\sqrt{w}) = \Theta(n/\sqrt{w}), \\ \ell &:= (4/\delta)(kd/\sqrt{w}) = \Theta(kn/w), \\ p &:= \sqrt{w} \end{aligned}$$

We think of the ℓ words as consisting of d chunks, each of which contains ℓ/d words.[7] We will ensure that $h(x)$ selects 1 word per chunk, for d words total. In each word it selects $p = \sqrt{w}$ bits, for a total of $d\sqrt{w}$ bits altogether. The selection is done as follows.

Each hash function $h \in \mathcal{H}$ consists of three *component hash functions* h_v, h'_v, h''_v for every chunk $v \in [d]$, whose properties we list below. We also rely on an averaging sampler with distinct elements $\mathsf{Samp} : \{0,1\}^s \times [\sqrt{w}] \to [w]$ with accuracy $\delta/4$, error $\rho = 2^{-\Omega(\sqrt{w})}$ and seed length $s = O(\sqrt{w})$, as given by Theorem 2.10. To compute $h(x)$, initialize $S := \emptyset$. Do the following for each chunk $v \in [d]$:

- Compute $\hat{x}_v = h_v(x)$, where $h_v : \{0,1\}^n \to \{0,1\}^{\sqrt{w}}$ is a $2^{-\Omega(\sqrt{w})}$-universal hash.
- Compute $\hat{i}_v = h'_v(\hat{x}_v)$, where $h'_v : \{0,1\}^{\sqrt{w}} \to [\ell/d]$ is $\sqrt{w}$-wise independent. Let $i_v = (v-1)\cdot(\ell/d)+\hat{i}_v$ be the word selected in chunk v.
- Compute $\mathsf{seed}_v = h''_v(\hat{x}_v)$ where $h''_v : \{0,1\}^{\sqrt{w}} \to \{0,1\}^s$ is $\sqrt{w}$-wise independent. Let $j_{v,u} = \mathsf{Samp}(\mathsf{seed}_v, u)$ for $u \in [\sqrt{w}]$.
- Add the values $\{(i_v, j_{v,u})\}_{u\in[\sqrt{w}]}$ to the set S.

Proof of Lemma 5.9. We show that our construction above is (k, δ, ε)-overlap avoiding for $\varepsilon = 2^{-\Omega(n)}$.

Fix any $x_1, \ldots, x_k \in \{0,1\}^n$. For $q \in [k]$ we let $\hat{x}_v(q), i_v(q), j_{v,u}(q)$ for $v \in [d], u \in [\sqrt{w}]$ be the corresponding values computed during the evaluation of $h(x_q)$. For $v \in [d]$, define the following events over a random choice of $h \leftarrow \mathcal{H}$:

- A_v: the event that $\hat{x}_v(k) \in \{\hat{x}_v(1), \ldots, \hat{x}_v(k-1)\}$. In other words, the universal hash of x_k for chunk v collides with that of some other x_q.
- B_v: the event that $|\{q \in [k-1] : i_v(k) = i_v(q)\}| \geq (\delta/2)\sqrt{w}$. In other words, the word $i_v(k)$ selected by x_k in chunk v was previously selected at least $(\delta/2)\sqrt{w}$ times by $x_1, \ldots, x_{k-1}$.
- C_v: the event that $|\{u \in [\sqrt{w}] : \exists q \in [k-1] \text{ s.t. } (i_v(k), j_{v,u}(k)) = (i_v(q), j_{v,u}(q))\}| \geq \frac{3}{4}\delta\sqrt{w}$. In other words, more than $\frac{3}{4}\delta$ fraction of the bit locations $\{(i_v(k), j_{v,u}(k))\}_{u\in[\sqrt{w}]}$ selected for x_k in chunk v overlap with those previously selected for $x_1, \ldots, x_{k-1}$.

Then we can bound:

- $\Pr[A_v] \leq k2^{-\Omega(\sqrt{w})} \leq 2^{-\Omega(\sqrt{w})}$ by the universality of the hash h_v and $\sqrt{w} = \Omega(\log k)$.
- $\Pr[B_v \mid \neg A_v] \leq 2^{-\Omega(\sqrt{w})}$ by the $\sqrt{w}$-wise independence of the hash h'_v. Fix any choice of h_v for which the event A_v does not occur. Let X_q be a random variable (over the choice of h'_v) that is 1 if $i_v(k) = i_v(q)$ and let $X = \sum_{q\in[k-1]} X_q$. The values X_i are $(\sqrt{w}-1)$-wise independent and we have $E[X] = \sum_{q\in[k-1]} E[X_q] \leq kd/\ell \leq (\delta/4)\sqrt{w}$. Therefore, by a Chernoff bound for limited independence (Lemma 2.11) we have $\Pr[B_V \mid \neg A_v] = \Pr[X \geq \delta\sqrt{w}/2] \leq 2^{-\Omega(\sqrt{w})}$.
- $\Pr[C_v \mid \neg A_v \wedge \neg B_v] \leq 2^{-\Omega(\sqrt{w})}$. Let us fix any choice of h_v, h'_v for which the events A_v, B_v do not occurs. This choice also fixes the choice of the words $i_v(q)$ selected in chunk v for each input $q \in [k]$. Let $Q_v := \{q \in [k-1] : i_v(k) = i_v(q)\}$ be the set of inputs x_q for which we selected the same word $i_v(k)$ in chunk v as for input x_k. Since we conditioned on B_v not occurring, we have $|Q_v| \leq (\delta/2)\sqrt{w}$. Let $P_v = \{j_{v,u}(q) : q \in Q_v, u \in [\sqrt{w}]\}$ be the subset of bits in the word selected in chunk v by x_k that were previously selected by $x_1, \ldots, x_{k-1}$. Since $|Q_v| \leq (\delta/2)\sqrt{w}$ we have $|P_v| \leq (\delta/2)w$. Furthermore, by the $\sqrt{w}$-wise independence of h''_v, the value seed_k is completely independent of $\{\mathsf{seed}_q : q \in Q_v\}$ and therefore also independent of P_v. By the property of the sampler we therefore have $\Pr[|\{u \in [\sqrt{w}] : j_{v,u}(k) \in P_v| \geq \frac{3}{4}\delta\sqrt{w}] \leq \rho = 2^{-\Omega(\sqrt{w})}$.

[7] For simplicity, we assume all of the above values are integers. If not increase w to be a perfect square, increase n and k so they are multiples of $\sqrt{w}$ and decrease δ so that δ^{-1} is an integer; this only changes the values by constant factors and does not affect the asymptotics.

Overall we have $\Pr[C_v] \le \Pr[A_v] + \Pr[B_v \mid \neg A_v] + \Pr[C_v \mid \neg A_v \wedge \neg B_v] \le 2^{-\Omega(\sqrt{w})}$.

Finally, let D be the event that C_v occurs for more than $(\delta/4)d$ of the values $v \in [d]$. Since the events C_v are independent of each other we can bound $\Pr[D] \le \binom{d}{(\delta/4)d} 2^{-\Omega(\sqrt{w}d)} \le 2^{-\Omega(\sqrt{w}d)}$. If D does not occur than for at least $(1-\delta/4)d$ chunks we select at least $(1-3\delta/4)\sqrt{w}$ fresh (non-overlapping) bits and therefore at least $(1-\delta)dp$ fresh bits total.

To analyze the description length, we can use hash functions based on polynomial evaluation for h_v, h'_v, h''_v. For h_v we interpret the input x as a polynomial of degree $\le n/\sqrt{w}$ over $\mathbb{F}_{2^{\sqrt{w}}}$ and we evaluate it on a random point in $\mathbb{F}_{2^{\sqrt{w}}}$ specified by the description of the hash function; this is $(n/\sqrt{w})2^{-\sqrt{w}} = 2^{-\Omega(\sqrt{w})}$-universal when $\sqrt{w} = \Omega(\log n)$. For h'_v, h''_v we can interpret the description of the hash as a random $\sqrt{w}$-degree polynomial over $\mathbb{F}_2^{\max\{\sqrt{w},s\}}$ that we evaluate on the input. For each $v \in [d]$ the description length of h_v, h'_v, h''_v is therefore bounded by $O(w)$ bits, and hence can be stored using only $O(1)$ words. □

Theorem 5.10. *Let k, n, w be parameters such that $w = \Omega(\log^2 k + \log^2 n)$ and $k = \mathsf{poly}(n)$, and let $\Sigma = \{0,1\}^w$. Then there is a $(\varepsilon = 2^{-\Omega(n)})$-almost k-wise independent t-word-local hash family a family $\mathcal{F} = \{f_s : \{0,1\}^n \to \{0,1\}^n\}_{s \in \Sigma^\ell}$ with key length $\ell = O(\lceil kn/w \rceil)$ words and word-locality $t = O(\lceil\ n/\sqrt{w}\ \rceil)$. In particular, for $w = n$, the word-locality is $t = O(\sqrt{n})$ and for $w = n^2$ the locality is just $t = O(1)$.*

Proof. First we prove the theorem under the additional conditions $w \le \min\{k^2, n^2\}$. The construction is the same as in Lemma 5.6 for the bit case, but we now use a $(k, \delta, \varepsilon_0)$-word-based overlap avoiding family $\mathcal{H}$ of hash functions $h : \{0,1\}^n \to C_{\ell_0,w,d,p}$ with $\delta = .1, \varepsilon_0 = 2^{-\Omega(n)}$, and $\ell_0 = O(\lceil kn/w \rceil)$, $d = \Theta(\lceil\ n/\sqrt{w}\ \rceil)$, $p = \Theta(\sqrt{w})$ such that $dp = \Omega(n)$, which can be instantiated using Lemma 5.9. We can think of $\mathcal{H}$ as a $(k, \delta, \varepsilon_0)$-bit-based overlap avoiding family $h : \{0,1\}^n \to C_{\ell_0 w,dp}$. By combining this with a $(1-\delta)d$-robust matrix $\mathbf{A} \in \mathbb{F}_2^{m \times d}$, which exists for some $m = \Omega(d)$, and applying Lemma 5.6 we get a $\mathcal{F} = \{f_s : \{0,1\}^n \to \{0,1\}^m\}_{s \in \{0,1\}^{\ell'}}$ where $\ell' = \ell_0 w + \log|\mathcal{H}| = \ell_0 w + O(dw)$ bits, which is $(\varepsilon = k\varepsilon_0 = 2^{-\Omega(n)})$-almost k-wise independent. We can reinterpret $s \in \{0,1\}^{\ell'}$ as $s \in \Sigma^\ell$ for $\ell = \ell_0 + d = O(\lceil kn/w \rceil)$. Moreover, since $\mathcal{H}$ is word-based the construction only reads $O(d)$ words. This almost gives the theorem, except that we get output length $\{0,1\}^m$ for $m = \Omega(n)$ rather than $\{0,1\}^n$. To fix this we apply the above construction with an artificially inflated input size $n' \ge n$ with $n' = O(n)$ to ensure that the corresponding output size is $m \ge n$.

Lastly, we need to handle $w > \min\{k^2, n^2\}$. If $w \ge k^2$, then $n/\sqrt{w}$ words is more than nk bits and therefore the theorem is trivially achievable using a polynomial-evaluation based k-wise independent hash function with key size and locality nk bits. On the other hand, if $w \ge n^2$ then we can just apply the above theorem with a "reduced word size" $w' = n^2$, which already gives locality $t = O(1)$ in reduced words and optimal key size of $O(\lceil k/n \rceil n^2)$ bits. Then we get the result just by reinterpreting the key as consisting of w-bit "big words" and emulate each access to a "reduced word" by reading the corresponding "big word". □

6 Summary and Open Problems

In this work we studied two models for *local* hash functions: *perfectly independent* local hash functions and *almost independent* local hash functions. For perfectly independent local hash functions, we presented an optimal construction based on expander graphs, and showed that any such optimal construction in fact yields constructions of expander graphs. Then, we introduce the model of almost independent local hash functions, and give explicit constructions in both the bit-local and word-local models. In the case of hash functions having input and output size n and statistical error $\varepsilon = 2^{-n}$, our constructions are nearly optimal we provide almost matching lower bounds up to logarithmic factors.

An obvious and well-known question to construct better explicit expanders, which are necessary for better explicit perfectly independent local hash functions. Another open problem is to close the logarithmic gaps between the lower bounds and explicit constructions of almost-independent hashing when the input and output size is n and the error is $\varepsilon = 2^{-n}$. In addition, there are some gaps between lower bounds and constructions for general output size m and eror ε, and it remain open to give tight constructions and matching lower bounds for the full parameter range. In addition, currently we have two separate but

related constructions for almost independent hashing: one for bit-local hash functions and one for word-local hash functions. It would be nice to unify them into a single construction that works for all word sizes $1 \leq w \leq n^2$. Moreover, while this was not an emphasis of this work, it is of interest to optimize the computation time of the hash functions we construct, rather than just locality. Finally, it would be good to find additional applications of almost independent local hash functions, where one would benefit from their improved parameters (especially in the word model) as compared to perfectly independent functions.

References

[BR94] Mihir Bellare and John Rompel. Randomness-efficient oblivious sampling. In *35th Annual Symposium on Foundations of Computer Science, Santa Fe, New Mexico, USA, November 20-22, 1994*, pages 276–287. IEEE Computer Society, 1994.

[CGH+85] Benny Chor, Oded Goldreich, Johan Håstad, Joel Friedman, Steven Rudich, and Roman Smolensky. The bit extraction problem of t-resilient functions (preliminary version). In *26th Annual Symposium on Foundations of Computer Science, Portland, Oregon, USA, 21-23 October 1985*, pages 396–407. IEEE Computer Society, 1985.

[CPT15] Tobias Christiani, Rasmus Pagh, and Mikkel Thorup. From independence to expansion and back again. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 813–820. ACM, 2015.

[DH90] Martin Dietzfelbinger and Friedhelm Heide. A new universal class of hash functions and dynamic hashing in real time. In Mike Paterson, editor, *Automata, Languages and Programming, 17th International Colloquium, ICALP90, Warwick University, England, UK, July 16-20, 1990, Proceedings*, Lecture Notes in Computer Science, pages 6–19. Springer, 1990.

[DY21] Yevgeniy Dodis and Kevin Yeo. Doubly-affine extractors, and their applications. In Stefano Tessaro, editor, *2nd Conference on Information-Theoretic Cryptography, ITC 2021, July 23-26, 2021, Virtual Conference*, volume 199 of *LIPIcs*, pages 13:1–13:23. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[Gil98] David Gillman. A Chernoff bound for random walks on expander graphs. *SIAM Journal on Computing*, 27(4):1203–1220, 1998.

[GN93] Peter Gemmell and Moni Naor. Codes for interactive authentication. In Douglas R. Stinson, editor, *Advances in Cryptology - CRYPTO '93, 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 22-26, 1993, Proceedings*, volume 773 of *Lecture Notes in Computer Science*, pages 355–367. Springer, 1993.

[Gol11] Oded Goldreich. A sample of samplers: A computational perspective on sampling. In *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation: In Collaboration with Lidor Avigad, Mihir Bellare, Zvika Brakerski, Shafi Goldwasser, Shai Halevi, Tali Kaufman, Leonid Levin, Noam Nisan, Dana Ron, Madhu Sudan, Luca Trevisan, Salil Vadhan, Avi Wigderson, David Zuckerman*, pages 302–332. Springer, 2011.

[GUV07] Venkatesan Guruswami, Christopher Umans, and Salil P. Vadhan. Unbalanced expanders and randomness extractors from Parvaresh-Vardy codes. In *22nd Annual IEEE Conference on Computational Complexity (CCC 2007), 13-16 June 2007, San Diego, California, USA*, pages 96–108. IEEE Computer Society, 2007.

[Jus03] Jørn Justesen. Class of constructive asymptotically good algebraic codes. *IEEE Transactions on information theory*, 18(5):652–656, 2003.

[Lev86] Leonid A Levin. Average case complete problems. *SIAM Journal on Computing*, 15(1):285–286, 1986.

[LPP+24] Kasper Green Larsen, Rasmus Pagh, Giuseppe Persiano, Toniann Pitassi, Kevin Yeo, and Or Zamir. Optimal non-adaptive cell probe dictionaries and hashing. In Karl Bringmann, Martin Grohe, Gabriele Puppis, and Ola Svensson, editors, *51st International Colloquium on Automata, Languages, and Programming, ICALP 2024, July 8-12, 2024, Tallinn, Estonia*, volume 297 of *LIPIcs*, pages 104:1–104:12. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024.

[ÖP03] Anna Östlin and Rasmus Pagh. Uniform hashing in constant time and linear space. In Lawrence L. Larmore and Michel X. Goemans, editors, *Proceedings of the 35th Annual ACM Symposium on Theory of Computing, June 9-11, 2003, San Diego, CA, USA*, pages 622–628. ACM, 2003.

[Pat08] Jacques Patarin. The "coefficients H" technique. In *International Workshop on Selected Areas in Cryptography*, pages 328–345. Springer, 2008.

[PP08] Anna Pagh and Rasmus Pagh. Uniform hashing in constant time and optimal space. *SIAM J. Comput.*, 38(1):85–96, 2008.

[RT97] Jaikumar Radhakrishnan and Amnon Ta-Shma. Tight bounds for depth-two superconcentrators. In *38th Annual Symposium on Foundations of Computer Science, FOCS 1997, Miami Beach, Florida, USA, October 19-22, 1997*, pages 585–594. IEEE Computer Society, 1997.

[Sha49] Claude E Shannon. A theorem on coloring the lines of a network. *Journal of Mathematics and Physics*, 28(1-4):148–152, 1949.

[Sie89] Alan Siegel. On universal classes of fast high performance hash functions, their time-space tradeoff, and their applications (extended abstract). In *30th Annual Symposium on Foundations of Computer Science, Research Triangle Park, North Carolina, USA, 30 October - 1 November 1989*, pages 20–25. IEEE Computer Society, 1989.

[Sie04] Alan Siegel. On universal classes of extremely random constant-time hash functions. *SIAM J. Comput.*, 33(3):505–543, 2004.

[SSS95] Jeanette P. Schmidt, Alan Siegel, and Aravind Srinivasan. Chernoff-Hoeffding bounds for applications with limited independence. *SIAM J. Discret. Math.*, 8(2):223–250, 1995.

[Ta-17] Amnon Ta-Shma. Explicit, almost optimal, epsilon-balanced codes. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 238–251. ACM, 2017.

[Tho13] Mikkel Thorup. Simple tabulation, fast expanders, double tabulation, and high independence. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, Berkeley, CA, USA, October, 26-29, 2013*, pages 90–99. IEEE Computer Society, 2013.

[Vad04] Salil P Vadhan. Constructing locally computable extractors and cryptosystems in the bounded-storage model. *Journal of Cryptology*, 17(1):43–77, 2004.

[XZ25] Zhiyang Xun and David Zuckerman. Near-Optimal Averaging Samplers and Matrix Samplers. In *40th Computational Complexity Conference (CCC 2025)*, volume 339, pages 6:1–6:28, 2025.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il