

Derandomization of Cell Sampling

Alexander Golovnev*

Tom Gur[†]Igor Shinkar[‡]

Abstract

Since 1989, the best known lower bound on static data structures was Siegel’s classical cell sampling lower bound. Siegel showed an explicit problem with n inputs and m possible queries such that every data structure that answers queries by probing t memory cells requires space $s \geq \tilde{\Omega}(n \cdot (\frac{m}{n})^{1/t})$. In this work, we improve this bound for non-adaptive data structures to $s \geq \tilde{\Omega}(n \cdot (\frac{m}{n})^{1/(t-1)})$ for all $t \geq 2$.

For $t = 2$, we give a lower bound of $s > m - o(m)$, improving on the bound $s > m/2$ recently proved by Viola over \mathbb{F}_2 and Siegel’s bound $s \geq \tilde{\Omega}(\sqrt{mn})$ over other finite fields.

1 Introduction

For a finite field \mathbb{F} , a *static data structure problem* with n inputs and m possible queries is given by a function $f: \mathbb{F}^n \times [m] \rightarrow \mathbb{F}$. A non-adaptive static data structure (in the cell probe model) consists of two algorithms. The preprocessing algorithm takes an input $x \in \mathbb{F}^n$ and preprocesses it into s memory cells $P \in \mathbb{F}^s$. The query algorithm Q takes an index $i \in [m]$, then non-adaptively probes at most t memory cells from P , and has to compute $f(x, i)$. That is, we require that $Q(P, i) = f(x, i)$ for all $i \in [m]$. Here we assume that each input, memory cell, and query stores an element of the field \mathbb{F} .¹ We remark that in the cell probe model both the preprocessing and query algorithms are computationally unbounded.

Every data structure problem $f: \mathbb{F}^n \times [m] \rightarrow \mathbb{F}$ admits two trivial solutions:

- $s = m$ and $t = 1$, where in the preprocessing stage one precomputes the answers to all m queries. (This solution uses prohibitively large space.)
- $s = n$ and $t = n$, where one does not use preprocessing, but rather just stores the input. (This solution uses prohibitively large query time.)

A counting argument [Mil93] shows that a random data structure problem requires either almost trivial space $s \approx m$ or almost trivial query time $t \approx n$ (even in the case of adaptive data structures). The main challenge in this area is to prove a lower bound for an *explicit* problem where each

*Georgetown University. Email: alexgolovnev@gmail.com.

[†]University of Warwick. Email: tom.gur@warwick.ac.uk. Tom Gur is supported by the UKRI Future Leaders Fellowship MR/S031545/1.

[‡]Simon Fraser University. Email: ishinkar@sfu.ca.

¹Typically in the cell probe model, the maintained memory is modelled by a sequence of s cells each holding a w -bit string. The parameter w is called the *word size* of the model. In some scenarios the word size is equal to $\Theta(\log(n))$, however, w is often considered a parameter (see e.g., [Mil99]). In this work, the word size corresponds to the number of bits required to represent an element of the field \mathbb{F} .

output can be computed in polynomial time. The best known explicit lower bound was proven by Siegel [Sie04] in 1989, and his technique was further developed in [Pät11, PTW10, Lar12]. This technique is now called cell sampling, and it will be discussed in greater detail later in this section. For an explicit problem, cell sampling gives us a lower bound of

$$s \geq \tilde{\Omega} \left(n \cdot \left(\frac{m}{n} \right)^{1/t} \right),$$

and this lower bound holds even against adaptive data structures. In particular, for $m = \text{poly}(n)$, Siegel's result provides a problem that for linear space $s = O(n)$ requires logarithmic $t \geq \Omega(\log(n))$ query time. Alas, for super-linear space $s = n^{1+\varepsilon}$, this best known lower bound only gives us the trivial $t \geq \Omega(1)$ bound. It is a major challenge in this area to improve on Siegel's bound.

While for the case of $t = 1$, every non-trivial problem with m queries requires space $s \geq m$,² even the case of $t = 2$ is not well understood. The cell sampling technique for $t = 2$ gives a lower bound of $s \geq \tilde{\Omega}(\sqrt{mn})$, but this is still far from the optimal bound of $s \geq \Omega(m)$ for $m = \text{poly}(n)$. Only recently for the binary field \mathbb{F}_2 , Viola [Vio19] proved a strong lower bound of $s \geq m/2$ on the space complexity for the case of $t = 2$. Moreover, Viola [Vio19] showed that a better understanding of high lower bounds on the space complexity even for low values of t will lead to resolving a long-standing open problem in circuit complexity.

Our results. In this work, we further develop the cell sampling technique and improve its bound for the case of non-adaptive data structures to

$$s \geq \tilde{\Omega} \left(n \cdot \left(\frac{m}{n} \right)^{1/(t-1)} \right).$$

On the one hand, this new bound does not improve asymptotic lower bounds on the query time t for any value of s . On the other hand, for every fixed value of t , the new bound gives an asymptotically stronger lower bound on s . Furthermore, this bound essentially resolves the question for the case of $t = 2$: for every field, and every number of queries $m = \text{poly}(n)$, we give an explicit problem such that any data structure that probes $t = 2$ memory cells requires memory $s \geq m - o(m)$ (see item 1 of Theorem 1). This improves on the bound of Viola [Vio19], and answers a question asked by Rao and Natarajan Ramamoorthy [RN20].

Theorem 1. *Fix a finite field \mathbb{F} and a parameter $m = \text{poly}(n)$.*

1. *There exists an explicit problem with n inputs and m queries such that every non-adaptive static data structure solving it with query time $t = 2$ requires space $s \geq m - \tilde{O}(m/n)$.*
2. *For every $t \geq 3$, there exists an explicit problem with n inputs and m queries such that every non-adaptive static data structure solving it with query time t requires space*

$$s \geq \Omega \left(n \cdot \left(\frac{m}{n} \right)^{1/(t-1)} \cdot \frac{1}{2^t \log(n) \log(m)} \right).$$

²For example, any problem where every pair of queries has at least $|\mathbb{F}| + 1$ distinct pairs of values requires $s \geq m$ if $t = 1$.

The key step in the proof of Theorem 1 is Theorem 2 saying that every dense enough hypergraph contains a *small* dense subgraph. Some of the techniques used in the proof of Theorem 2 are well-known (for example, the proof of Claim 4 is similar to the standard upper bounds on the girth of a sparse graph [Bol98, Chapter IV, Theorem 1]). Nevertheless, we could not find results similar to Theorem 2 in the literature (possibly due to the upper bound on the size of S which may make this question less natural from the graph-theoretic point of view).

Theorem 2. *Let $G = (V, E)$ be a multigraph with $|V| = s \geq 2$ vertices and $|E| = m \geq s(1 + \varepsilon)$ edges for some $\varepsilon = \varepsilon(s) \in (0, 1]$. There exists a set of vertices $S \subseteq V$ of size $|S| \leq 8 \log(s) \cdot \lceil 1/\varepsilon \rceil$ spanning at least $|S| + 1$ edges.*

Let $t \geq 3$ be an integer, and $G = (V, E)$ be a t -hypergraph with $|V| = s \geq 2$ vertices and $|E| = m$ hyperedges. Let $k \in \mathbb{N}$ be a parameter such that $2^{t+2} \log(s) \leq k \leq s$. If

$$m \geq 3s \left(\frac{2^{t+3} \cdot s \cdot \log(s)}{k} \right)^{t-2}, \quad (1)$$

then there exists a subset $S \subseteq V$ of size $|S| \leq k$ that spans at least $|S| + \frac{k}{2^{t+1} \log(s)}$ hyperedges.

In the statement of Theorem 2 a *multigraph* is a graph that may contain parallel edges and parallel self-loops, and a *t -hypergraph* is a hypergraph where every edge has at most t vertices. If all the vertices of an edge e belong to a set of vertices S , then we say that S spans the edge e .

Comparison to the cell sampling bound. The classical cell sampling technique restricted to the case of non-adaptive data structures can be viewed as a slightly weaker version of Theorem 2. In the cell sampling argument, one picks k *random* vertices and proves that with non-zero probability they span at least $k + 1$ hyperedges. This way, each t -hyperedge is spanned with probability $\approx \left(\frac{k}{s}\right)^t$, and the expected number of spanned edges is $\approx m \cdot \left(\frac{k}{s}\right)^t$. This leads to the lower bound of $m \gtrsim \Omega\left(s\left(\frac{s}{k}\right)^{t-1}\right)$, which is weaker than the bound of $m \gtrsim \Omega\left(s\left(\frac{s}{k}\right)^{t-2}\right)$ from Theorem 2. The contribution of this work is a deterministic way to choose k vertices that span $k + 1$ hyperedges which improves on the aforementioned bound obtained by randomly sampling k vertices.³

The following proposition shows that the bound of Theorem 2 is essentially tight, which poses a barrier on further improvements using this technique.

Proposition 3. *Let $t \geq 3$, and $G = (V, E)$ be a t -uniform hypergraph with $|V| = s$ vertices and $|E| = m$ edges sampled uniformly at random. Then with positive probability G does not have a set of k vertices spanning at least k hyperedges for every $k \geq t$ satisfying $s \geq e^3 \cdot k \cdot \left(\frac{m}{k}\right)^{1/(t-1)}$.*

Proof. By the union bound over all k -subsets of vertices, and all k -subsets of edges, we have that the probability that G has k vertices spanning $\geq k$ hyperedges is at most

$$\binom{s}{k} \cdot \binom{m}{k} \cdot \left(\frac{\binom{k}{t}}{\binom{s}{t}} \right)^k.$$

³We remark that the work [PTW10] also uses a deterministic process similar to cell sampling to prove the standard cell sampling lower bound even against adaptive data structures. In this work, we give a different deterministic process for a stronger bound against (weaker) non-adaptive data structures.

Using the inequalities $\left(\frac{a}{b}\right)^b \leq \binom{a}{b} \leq \left(\frac{ae}{b}\right)^b$, we have that this probability is bounded from above by

$$\left(\frac{se}{k}\right)^k \cdot \left(\frac{me}{k}\right)^k \cdot \left(\frac{ke}{s}\right)^{tk} = \left(\frac{e^{t+2}mk^{t-2}}{s^{t-1}}\right)^k \leq \left(\frac{1}{e^{2t-5}}\right)^k \leq e^{-k} < 1.$$

□

2 Preliminaries

All logarithms in this paper are to the base two. By P_k and C_k we denote the path and the cycle graphs on k vertices, respectively. The length of a path is the number of edges in it. By multigraphs we mean graphs that may contain parallel edges and (possibly parallel) self-loops. The degree of a vertex is the number of incident edges, and a self-loop adds two to the degree. For a multigraph $G = (V, E)$ and a subset of its vertices $S \subseteq V$, $G[S]$ denotes the subgraph of G induced on the vertices S .

By t -hypergraphs we mean hypergraphs where each edge contains at most t distinct vertices, and where parallel edges are allowed. A t -uniform hypergraph is a t -hypergraph where each edge contains exactly t vertices. We say that a set of vertices S spans a hyperedge e , if all the vertices of e belong to S .

We will need the following auxiliary claim that shows that for every vertex v of a cubic graph, there is a small tadpole graph starting at v .

Claim 4. *Let $G = (V, E)$ be a multigraph with $|V| = s \geq 2$ vertices, and let $v \in V$. Suppose that $\deg(v) \geq 1$ and $\deg(u) \geq 3$ for all $u \in V \setminus \{v\}$. Then G contains a path (p_1, p_2, \dots, p_k) and a cycle $(c_1, c_2, \dots, c_\ell)$ such that $p_1 = v$, $p_k = c_1$, $k \geq 1$, $\ell \geq 1$, and $k + \ell \leq 4 \log(s)$.*

In Claim 4 a cycle of length 1 means a self-loop, and a cycle of length 2 corresponds to a pair of parallel edges between two vertices. In particular, if v has a self-loop, then we can take the path to be (v) and the cycle to be (v) .

Proof. Recall that a self-loop and a pair of parallel edges are cycles. For $s \leq 3$, a graph where all but one vertex have degree at least three contains a path from v of length $k \leq 2$ to a cycle of length $\ell \leq 2$. Therefore, $k + \ell \leq 4 \leq 4 \log(s)$, and the statement of the claim follows. Hence, in the following we assume that $s \geq 4$.

In order to prove the claim, we run the Breadth First Search (BFS) algorithm starting at the vertex v . Let us say that v is at the first level of the BFS tree, and let t be the smallest integer such that the first t levels of the BFS tree contain a cycle (note that a self-loop or a pair of parallel edges count as cycles). Such t exists since G has at most one vertex of degree 1.

Since all vertices in $V \setminus \{v\}$ have degrees at least three, the number of vertices in the first $t - 1$ levels of the BFS tree is at least $1 + 1 + 2 + 4 + 8 + \dots + 2^{t-3} = 2^{t-2}$. On the other hand, the total number of vertices is $|V| = s$. This implies that $s > 2^{t-2}$, and hence $t < \log(s) + 2$.

Let $(c_1, c_2, \dots, c_\ell)$ be the obtained cycle. Suppose without loss of generality that the level of c_1 in the BFS tree is the minimal among the c_i 's levels, and let $(v = p_1, p_2, \dots, p_k = c_1)$ be the path from v to c_1 in the BFS tree. Observing that the length of the cycle from level k to level t has at most $\ell \leq 2(t - k) + 1$ edges (where one edge may connect two vertices in the same level), we have that $k + \ell \leq k + \ell + (k - 1) = 2k + \ell - 1 \leq 2t < 2 \log(s) + 4 \leq 4 \log(s)$, as required. □

3 Dense subgraphs in hypergraphs

We are now ready to prove the first part of Theorem 2.

Lemma 5. *For every $\varepsilon > 0$ and every multigraph $G = (V, E)$ with $|V| = s \geq 2$ vertices and $|E| = m \geq s(1 + \varepsilon)$ edges, there exists a set of $k \leq 8 \log(s) \cdot \lceil 1/\varepsilon \rceil$ vertices spanning at least $k + 1$ edges.*

Proof. We repeatedly apply the following operations to the graph G as long as at least one of them is applicable.

- If G contains an isolated vertex, then we remove this vertex.
- If G contains a vertex of degree one, then we remove this vertex and the incident edge from the graph. In this case, we remove one edge and one vertex.
- If G contains a vertex whose only incident edge is its self-loop, then we remove this vertex with the self-loop. Again, we remove one edge and one vertex.
- If G contains a path of length $\ell \geq 1/\varepsilon + 1$ consisting of vertices of degree two, then we remove all vertices of degree two (i.e., internal vertices) belonging to this path with all the incident edges. In this case, we remove $\ell - 1 \geq 1/\varepsilon$ vertices and ℓ edges.

Note that these four operations do not decrease the average degree of the graph, as the resulting graph has s' vertices and m' edges such that $m'/s' \geq m/s \geq (1 + \varepsilon)$. Each of the remaining vertices of degree two in the resulting graph belongs to a path of degree-two vertices of length less than $\lceil 1/\varepsilon \rceil$. We contract each such maximal path into an edge, and obtain a graph $G_1 = (V_1, E_1)$ of minimum degree three. (Note that such contraction may create a self-loop, in case that the endpoints of the path are the same vertex.) We will use the following observation: since the length of each contracted path is at most $\lceil 1/\varepsilon \rceil$, when we expand a contracted edge back into a path, we add at most $\lceil 1/\varepsilon - 1 \rceil$ vertices to the graph.

We apply Claim 4 to the graph G_1 and an arbitrary vertex $v_1 \in V_1$, and get a cycle $C_\ell^{(1)}$ in G_1 of length $\ell \leq 4 \log(s)$. (Here, we ignore the path guaranteed by Claim 4.) Next, we consider the following two cases.

- If $G_1[C_\ell^{(1)}]$ is a connected component in G_1 , then since each vertex of G_1 has degree at least 3, the vertices of $C_\ell^{(1)}$ must span at least $\lceil 1.5\ell \rceil \geq \ell + 1$ edges. Let S be the vertices of the subgraph $G_1[C_\ell^{(1)}]$, together with the vertices obtained by expanding $\ell + 1$ of the contracted edges back into the vertices and edges of G . Since each expanded edge adds the same number of vertices and edges, we have that S spans at least $|S| + 1$ edges. Since each of the $\ell + 1$ expanded edges introduces at most $\lceil 1/\varepsilon - 1 \rceil$ new vertices, we have that $|S| \leq \ell + (\ell + 1) \cdot \lceil 1/\varepsilon - 1 \rceil \leq 2\ell \cdot \lceil 1/\varepsilon \rceil \leq 8 \log(s) \cdot \lceil 1/\varepsilon \rceil$. Thus, the constructed set S satisfies the required property.
- Otherwise, we contract $C_\ell^{(1)}$ into a new vertex v_2 , and denote the obtained graph by G_2 . Since $G_1[C_\ell^{(1)}]$ is not a connected component in G_1 , it follows that v_2 is not an isolated vertex in G_2 , and, hence, $\deg(v_2) \geq 1$. We now apply Claim 4 to G_2 and the vertex v_2 , and get a path $P_{k'}^{(2)}$ and a cycle $C_{\ell'}^{(2)}$ in G_2 with $k' + \ell' \leq 4 \log(s)$ such that $v_2 \in P_{k'}^{(2)}$.

Recall that the vertex v_2 in G_2 corresponds to $C_\ell^{(1)}$ in G_1 . Thus, the set of vertices $S' = C_\ell^{(1)} \cup P_{k'}^{(2)} \cup C_{\ell'}^{(2)}$ forms two cycles connected by a path of length $k' - 1$ in G_1 . Then S' has $\ell + (k' - 2) + \ell' \leq \ell + 4 \log(s) - 1 \leq 8 \log(s) - 1$ vertices, and spans at least $|S'| + 1$ edges in G_1 . By expanding $|S'| + 1$ contracted edges, we again have a set S of vertices of G spanning at least $|S| + 1$ edges. It remains to note that the number of vertices in S is $|S| \leq |S'| + (|S'| + 1) \cdot \lceil 1/\varepsilon - 1 \rceil \leq (|S'| + 1) \cdot \lceil 1/\varepsilon \rceil \leq 8 \log(s) \cdot \lceil 1/\varepsilon \rceil$.

This completes the proof of Lemma 5. \square

The following lemma generalizes Lemma 5 by finding a small induced subgraph with a large gap between the number of vertices and the number of edges.

Lemma 6. *Let $G = (V, E)$ be a multigraph with $|V| = s$ and $|E| = m$. For every $g \geq 1$ satisfying $m \geq 2s + g + 1$, there is a subset of vertices $S \subseteq V$ of size at most $|S| \leq 8g \log(s)$ that spans at least $|S| + g$ edges.*

Proof. The proof is by induction on $g \geq 1$. The base case of $g = 1$ follows from Lemma 5 with $\varepsilon = 1$. Next we assume that the lemma is true for $g - 1$, and prove it for $g \geq 2$. By the induction hypothesis there is a subset of vertices $S_{g-1} \subseteq V$ of size at most $|S| \leq 8(g - 1) \log(s)$ such that S_{g-1} spans at least $|S_{g-1}| + g - 1$ edges.

If S_{g-1} spans $\geq |S_{g-1}| + g$ edges, then we are done. Otherwise, S_{g-1} must span exactly $|S| + g - 1$ edges. Consider a graph $G' = (V', E')$ obtained from G by contracting S_{g-1} into a new vertex v^S and removing the edges with both ends in S_{g-1} . Then G' has $s' = |V'| = s - |S| + 1$ vertices and $m' = |E'| = m - (|S| + g - 1)$ edges. In particular, $m' \geq (2s + g + 1) - (|S| + g - 1) \geq 2s - 2|S| + 3 = 2s' + 1$. Therefore, by Lemma 5, G' has a subset of vertices $S' \subseteq V'$ of size at most $|S'| \leq 8 \log(s)$ such that S' spans at least $|S'| + 1$ edges. We remark that S' may or may not contain the vertex v^S .

By taking $S = S_{g-1} \cup (S' \setminus \{v^S\})$ we obtain a set of $|S| \leq 8g \log(s)$ vertices spanning at least $|S| + g$ edges. \square

We now finish the proof of Theorem 2.

Theorem 2. *Let $G = (V, E)$ be a multigraph with $|V| = s \geq 2$ vertices and $|E| = m \geq s(1 + \varepsilon)$ edges for some $\varepsilon = \varepsilon(s) \in (0, 1]$. There exists a set of vertices $S \subseteq V$ of size $|S| \leq 8 \log(s) \cdot \lceil 1/\varepsilon \rceil$ spanning at least $|S| + 1$ edges.*

Let $t \geq 3$ be an integer, and $G = (V, E)$ be a t -hypergraph with $|V| = s \geq 2$ vertices and $|E| = m$ hyperedges. Let $k \in \mathbb{N}$ be a parameter such that $2^{t+2} \log(s) \leq k \leq s$. If

$$m \geq 3s \left(\frac{2^{t+3} \cdot s \cdot \log(s)}{k} \right)^{t-2}, \quad (1)$$

then there exists a subset $S \subseteq V$ of size $|S| \leq k$ that spans at least $|S| + \frac{k}{2^{t+1} \log(s)}$ hyperedges.

Proof. The first part of the theorem is proven in Lemma 5. For the second part of the theorem, without loss of generality, we assume that G is t -uniform. Indeed, if an edge of G has fewer than t vertices, then we extend this edge with arbitrary vertices, and the theorem statement for the new graph will imply the statement for G .

The proof of the second part of the theorem is by induction on $t \geq 2$. For the base case of $t = 2$ the statement follows immediately from Lemma 6. Indeed, for $t = 2$ the bound (1) implies that $m \geq 3s$, and by Lemma 6 with $g = \frac{k}{8 \log(s)}$ we get the desired conclusion.

For the induction step, let us prove the statement of the theorem for t , assuming that it holds for $t - 1$. Let ℓ be an integer such that $\frac{k}{2^{t+3} \log(s)} \leq \ell \leq \frac{k}{2^{t+2} \log(s)}$. Note that since $t \geq 3$ and $s \geq 2$, we have that $\ell \leq k/4$. First we show that there exists a subset $L \subseteq V$ of ℓ vertices such that the number of hyperedges touching them is $E_L = |\{e \in E : e \cap L \neq \emptyset\}| \geq \frac{\ell m}{s}$. Indeed, since $\sum_{v \in V} \deg(v) \geq tm$, there must exist a set $L \subseteq V$ such that $\sum_{v \in L} \deg(v) \geq \frac{\ell tm}{s}$. And since each hyperedge is counted in the sum at most t times, it follows that the number of edges adjacent to L is at least $\frac{\ell m}{s}$.

Associate each hyperedge $e \in E_L$ with some vertex $v_e \in e \cap L$. That is, if e contains a unique vertex v_e in $e \cap L$, then we associate e with this v_e , and if there is more than one such vertex, then we choose $v_e \in e \cap L$ arbitrarily.

Define the graph $G^* = (V, E^*)$, where $E^* = \{e \setminus \{v_e\} : e \in E_L\}$. Note that G^* has s vertices, and by the assumption on m in (1), the number of hyperedges (of size at most $t - 1$) is at least

$$\begin{aligned} \frac{\ell m}{s} &\geq 3s \left(\frac{2^{t+3} \cdot s \cdot \log(s)}{k} \right)^{t-2} \cdot \frac{\ell}{s} \\ &\geq 3s \left(\frac{2^{t+3} \cdot s \cdot \log(s)}{k} \right)^{t-2} \cdot \frac{k}{2^{t+3} \cdot s \cdot \log(s)} \\ &= 3s \left(\frac{2^{t+2} \cdot s \cdot \log(s)}{k/2} \right)^{t-3} \\ &\geq 3s \left(\frac{2^{t+2} \cdot s \cdot \log(s)}{k - \ell} \right)^{t-3}, \end{aligned}$$

where the last inequality uses $k - \ell \geq k/2$. Therefore, we can apply the induction hypothesis to the $(t - 1)$ -hypergraph G^* with $k - \ell$ being the bound on the size of the guaranteed set. We get that G^* has a subset $S^* \subseteq V$ of size $|S^*| \leq (k - \ell)$ that spans at least $|S^*| + \frac{k - \ell}{2^t \log(s)}$ hyperedges. Define the set $S = S^* \cup L$. Therefore, $|S| \leq |S^*| + |L| \leq (k - \ell) + \ell \leq k$, and since the number of hyperedges spanned by S in G is at least the number of hyperedges spanned by S^* in G^* , it follows that S spans at least

$$|S^*| + \frac{k - \ell}{2^t \log(s)} \geq |S| - \ell + \frac{k - \ell}{2^t \log(s)} \geq |S| - \frac{k}{2^{t+2} \log(s)} + \frac{3k}{2^{t+2} \log(s)} \geq |S| + \frac{k}{2^{t+1} \log(s)}$$

edges, as required. \square

4 Data structure lower bound

We are now ready to prove the main theorem of this paper using Theorem 2. We will prove our data structure lower bound for k -wise independent functions. A function $f: \mathbb{F}^n \rightarrow \mathbb{F}^m$ is called k -wise independent if for every k -tuple S of outputs, the uniform distribution of the n inputs induces the uniform distribution on S .

One way to construct k -wise independent functions utilizes linear error correcting codes. It is well known that the parity check matrix of a linear code with distance $k + 1$ is k -wise independent. Therefore, one can define a k -wise independent data structure problem as the problem of multiplying an input vector $x \in \mathbb{F}^n$ by a fixed parity check matrix $M \in \mathbb{F}^{n \times m}$ of a code with a large distance. In particular, for fields of size $|\mathbb{F}| > m$, one can achieve n -wise independence by taking M as

the Vandermonde matrix. For smaller fields, one can take rate-optimal linear codes [MS77] and achieve $\Omega(n)$ -wise independence for $m = O(n)$ and $\Omega(n/\log_{|\mathbb{F}|}(n))$ -wise independence for every $m = \text{poly}(n)$, which is tight [CGH⁺85].

We remark that the result of Theorem 1 applies to non-linear k -wise independent functions as well, and in fact it can be generalized to almost k -wise independent functions recovering the class of functions for which [Sie04] proved the cell sampling lower bound. For ease of exposition, in the proof below we show a lower bound for data structures computing $k = \Theta(n/\log(n))$ -wise independent functions.

Theorem 1. *Fix a finite field \mathbb{F} and a parameter $m = \text{poly}(n)$.*

1. *There exists an explicit problem with n inputs and m queries such that every non-adaptive static data structure solving it with query time $t = 2$ requires space $s \geq m - \tilde{O}(m/n)$.*
2. *For every $t \geq 3$, there exists an explicit problem with n inputs and m queries such that every non-adaptive static data structure solving it with query time t requires space*

$$s \geq \Omega \left(n \cdot \left(\frac{m}{n} \right)^{1/(t-1)} \cdot \frac{1}{2^t \log(n) \log(m)} \right).$$

Proof. Consider a data structure for a $k = \Theta(n/\log(n))$ -wise independent problem. For such a problem, in order to answer any k -tuple of queries, one needs to read at least k memory cells. Indeed, every k -tuple of outputs of a k -wise independent function must take $|F|^k$ distinct values, and if it depended on $k - 1$ memory cells it could only take at most $|F|^{k-1}$ distinct values.

For $t = 2$, we construct a multigraph with s vertices corresponding to the memory cells of the data structure, and m edges, each corresponding to the pair of memory cells read for a query. Let $k > 8 \log(s)$ and $\varepsilon = \frac{16 \log(s)}{k}$. If $m \geq s(1 + \varepsilon)$, then by the first part of Theorem 2 we have a set of k queries that depends on $k - 1$ memory cells. Therefore, any data structure where in order to answer any k -tuple of queries, one needs to read at least k memory cells, must satisfy $s \geq m/(1 + \varepsilon) \geq m(1 - \varepsilon) = m - \frac{16m \log(s)}{k}$. By plugging $k = \Theta(n/\log(n))$, we obtain the desired lower bound of $s \geq m - \tilde{O}(m/n)$.

For $t \geq 3$, we construct a t -uniform hypergraph on s vertices, where the vertices correspond to the memory cells of the data structure, and m hyperedges correspond to the t -tuples of memory cells read for each query. Let $k \geq 2^{t+2} \log(s)$. If $m \geq 3s \left(\frac{2^{t+3} \cdot s \cdot \log(s)}{k} \right)^{t-2}$, then by the second part of Theorem 2, there exists a set of $k + 1$ queries that can be answered by k memory cells. Therefore, every data structure that does not have such a $(k + 1)$ -tuple of queries must satisfy $s \geq \left(\frac{m}{3} \right)^{1/(t-1)} \cdot \left(\frac{k}{2^{t+3} \log(s)} \right)^{(t-2)/(t-1)}$. Setting $k = \Theta(n/\log(n))$ leads to the bound of $s \geq \Omega \left(n \cdot \left(\frac{m}{n} \right)^{1/(t-1)} \cdot \frac{1}{2^t \log(n) \log(m)} \right)$. \square

Acknowledgments

We would like to thank the anonymous reviewers whose detailed comments significantly helped us to improve the presentation of this result.

References

- [Bol98] Béla Bollobás. *Modern graph theory*, volume 184. Springer, 1998.
- [CGH⁺85] Benny Chor, Oded Goldreich, Johan Håstad, Joel Freidmann, Steven Rudich, and Roman Smolensky. The bit extraction problem or t -resilient functions. In *FOCS 1985*, pages 396–407, 1985.
- [Lar12] Kasper Green Larsen. Higher cell probe lower bounds for evaluating polynomials. In *FOCS 2012*, pages 293–301, 2012.
- [Mil93] Peter Bro Miltersen. The bit probe complexity measure revisited. In *STACS 1993*, pages 662–671, 1993.
- [Mil99] Peter Miltersen. Cell probe complexity - a survey. *FSTTCS 1999*. Advances in Data Structures Workshop, 1999.
- [MS77] Florence Jessie MacWilliams and Neil James Alexander Sloane. *The theory of error-correcting codes*. Elsevier, 1977.
- [Păt11] Mihai Pătraşcu. Unifying the landscape of cell-probe lower bounds. *SIAM Journal on Computing*, 40(3):827–847, 2011.
- [PTW10] Rina Panigrahy, Kunal Talwar, and Udi Wieder. Lower bounds on near neighbor search via metric expansion. In *FOCS 2010*, pages 805–814, 2010.
- [RN20] Anup Rao and Sivaramakrishnan Natarajan Ramamoorthy. Personal communication, 2020.
- [Sie04] Alan Siegel. On universal classes of extremely random constant-time hash functions. *SIAM Journal on Computing*, 33(3):505–543, 2004. Preliminary version appeared in *FOCS*, 1989, 20–25.
- [Vio19] Emanuele Viola. Lower bounds for data structures with space close to maximum imply circuit lower bounds. *Theory of Computing*, 15(1):1–9, 2019.