# Unconditional Boolean Dynamic Lower Bounds from Disjointness

Young Kun Ko*

**Abstract**

We prove an unconditional $\Omega\big((\log n/\log\log n)^2\big)$ lower bound on the maximum of the worst-case update time and expected query time for the insert-only version of Pătraşcu's multiphase Disjointness problem [STOC 2010]. The bound holds in the zero-error cell-probe model with logarithmic word size. Pătraşcu's reductions transfer it to directed reachability under edge insertions, already on directed acyclic graphs of depth 2. This improves the $\widetilde{\Omega}(\log^{3/2} n)$ bound of Larsen and Yu [SICOMP 2025] to the quadratic-logarithmic scale previously attained for dynamic range counting and multiphase Inner Product. The bound also transfers to a broad collection of dynamic graph, algebraic, geometric, and string problems.

We also obtain a separation between amortized and worst-case time for streaming regular-expression membership. For a simple family of expressions of maximum length $M$ over a polynomial-size alphabet, a deterministic algorithm uses $O(1)$ amortized time per symbol on every input stream, whereas every zero-error cell-probe algorithm with word size $w = \Theta(\log M)$ requires $\Omega\big((\log M/\log\log M)^2\big)$ time on some symbol.

The proof adapts Ko's 2.5-round multiphase communication framework [FOCS 2026] from Inner Product to Disjointness. Our main technical observation is that the protocol's verification step allows a one-sided corruption bound to replace the discrepancy property used for Inner Product.

*Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA. ykko@psu.edu

# 1 Introduction

Recent algorithmic breakthroughs have exposed the vulnerability of central hardness conjectures. At the same time, Ko [Ko26] offered *a new hope* for unconditional Boolean dynamic lower bounds: a program to unify bounds at the quadratic-logarithmic scale through 2.5-round communication games. In this work, *the unconditional lower bounds strike back*: we extend the framework from Inner Product to Disjointness and obtain unconditional bounds for a broad collection of dynamic graph, set, matrix, geometric, and string problems.

The common source is Pătraşcu's multiphase Disjointness problem [Păt10]: preprocess a family of sets, insert elements into a set $X$, and finally test whether $X$ is disjoint from a specified member of the family. For $n$ preprocessed subsets of $[n]$, we prove

$$\max(t_u, t_q) = \Omega\left(\left(\frac{\log n}{\log\log n}\right)^2\right)$$

in the zero-error cell-probe model with word size $w = \Theta(\log n)$, where $t_u$ is the worst-case insertion time and $t_q$ the expected query time on each input. Preprocessing is unrestricted. Theorem 1.1 gives the stronger update/query trade-off from which all our applications follow.

Disjointness connects this theorem to problems with very different query interfaces. Known reductions, and adaptations of published constructions, transfer the bound to incremental directed reachability and bipartite matching, planar distances, range mode, inverse-entry and determinant maintenance, geometric counting, and string matching. They use $O(1)$ target operations per source operation and change the instance size only polynomially, so they preserve the quadratic-logarithmic scale. Table 1 lists these problems with references, and Appendix A collects the constructions, attributes their sources, and checks that they meet the requirements of our model.

Directed reachability is a leading example. Our bound holds already under edge insertions on depth-2 directed acyclic graphs and improves the $\widetilde{\Omega}(\log^{3/2} n)$ bound of Larsen and Yu [LY25] to the quadratic-logarithmic scale previously attained for dynamic range counting and multiphase Inner Product [Lar12, Ko26].

We also obtain a separation between amortized and worst-case time for streaming regular-expression membership, which asks after each arriving symbol whether the prefix read so far matches a preprocessed expression. For a simple family of expressions of maximum length $M$ over a polynomial-size alphabet, a deterministic algorithm uses $O(1)$ amortized time per symbol on every input stream, whereas every zero-error cell-probe algorithm with word size $w = \Theta(\log M)$ requires $\Omega\big((\log M/\log\log M)^2\big)$ time on some symbol (Corollary 1.3). Thus the Disjointness bound also gives an unconditional obstruction to deamortizing this streaming computation.

**The vulnerability of hardness conjectures.** Many of these dynamic problems have been studied through polynomial conditional lower bounds. Such a bound shows that a faster data structure would give a faster algorithm for a problem conjectured to be hard, such as 3SUM, all-pairs shortest paths (APSP), or CNF satisfiability [Păt10, AW14]. These reductions have been remarkably successful at charting relationships among problems, but they do not by themselves establish that any of the problems is hard. An unconditional lower bound proves hardness within the stated computational model, independently of such conjectures.

Recent events have made this distinction concrete. An algorithm discovered by Claude (Anthropic), subsequently analyzed, strengthened, and presented by Alman and Vassilevska Williams [AW26], yields truly subquadratic 3SUM and truly subcubic APSP algorithms, refuting

the corresponding conjectures. Two recent manuscripts released by OpenAI[1] and Anthropic[2] claim subexponential algorithms for 3-SAT, which would refute the Exponential Time Hypothesis (ETH) and hence the Strong Exponential Time Hypothesis (SETH) [IPZ01].

The refuted conjectures no longer supply hardness for the corresponding lower bounds. The web of reductions remains intact, but its starting premise is gone, and reductions alone cannot establish the hardness of their targets. Our goal is to supply the missing foundation with **an unconditional lower bound** and put the existing reduction network to use. We do so for multiphase Disjointness: our quadratic-logarithmic lower bound is a proved source of hardness, which established reductions propagate to a broad collection of dynamic problems.

## 1.1 Previous work

### 1.1.1 Conditional lower bounds across dynamic problems

Conditional lower bounds motivated this reduction network. Pătraşcu [Păt10] introduced multiphase Disjointness, proved its conditional hardness under the 3SUM conjecture, and gave reductions to reachability, shortest paths, subgraph connectivity, and set and array problems. Subsequent work developed conditional bounds for graph problems [AW14, HKNS15], planar graphs [AD16], dynamic matrices [vdBNS19], geometric counting [DI22, JX22], and string problems [CGLS18, CKM+19]. The 3SUM and APSP refutations [AW26] have removed central foundations of this program: those conjectures no longer support the associated lower bounds. The claimed ETH refutations discussed above would likewise refute SETH. The reductions remain valid, and many can be specialized or adapted to a single evolving set and a final selected query. The refutations thus leave the affected reduction chains intact, but with no valid source of hardness.

### 1.1.2 Unconditional cell-probe lower bounds

The cell-probe model [Yao81] charges only for memory access: memory consists of $w$-bit cells, each read or write costs one probe, and computation is free. A lower bound therefore also applies to word-RAM algorithms with the same word size. We work with $w = \Theta(\log n)$ and allow arbitrary preprocessing, so the lower bound concerns the information that must be accessed while processing updates and queries. The model's generous assumptions make lower bounds especially difficult to prove, helping explain the slow progress even for basic dynamic problems.

Fredman and Saks [FS89] introduced the Chronogram method and proved an $\Omega(\log n/\log\log n)$ lower bound, which transfers to dynamic connectivity [MSVT94, HF98]. Pătraşcu and Demaine [PD06] improved the connectivity bound to $\Omega(\log n)$. For almost two decades, the strongest unconditional lower bounds for natural graph problems remained at the (near-)logarithmic scale. For problems with word-size answers, Larsen [Lar12] reached $\Omega((\log n/\log\log n)^2)$ for weighted range counting. Reaching stronger bounds for Boolean queries required a further breakthrough: Larsen, Weinstein and Yu [LWY20] proved an $\widetilde{\Omega}(\log^{3/2} n)$ bound. Building on their technique, Larsen and Yu [LY25] obtained the first super-logarithmic bound for a natural graph problem, proving

$$t_q = \Omega\left(\frac{\log^{3/2} n}{\log^2(t_u w)}\right)$$

for incremental reachability on DAGs, with worst-case update time $t_u$ and expected query time $t_q$ [LY25, Theorem 1]. Figure 1 places our reachability consequence in this progression.

[1] *The Exponential Time Hypothesis is False*, OpenAI, October 25, 2026; available from the OpenAI Math repository.
[2] *A $2^{o(n)}$ Time Algorithm for 3-SAT*, Claude (Anthropic), 2026; paper and Lean formalization.

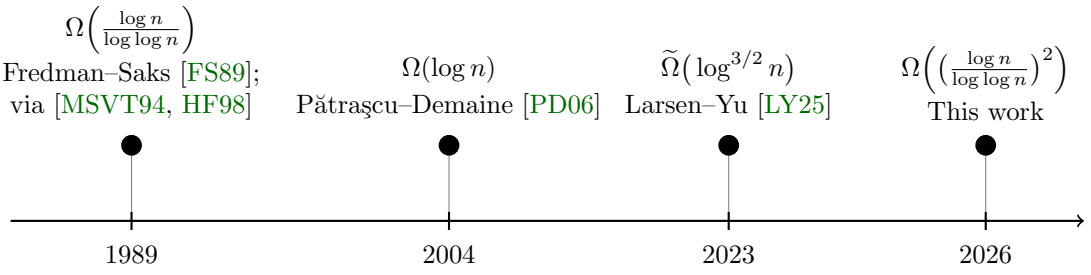


Figure 1: Unconditional cell-probe lower bounds for dynamic directed reachability, with $w = \Theta(\log n)$, stated as bounds on $\max(t_u, t_q)$. For prior results, dates refer to conference appearances. The first two bounds transfer from fully dynamic undirected connectivity to fully dynamic directed reachability; the later bounds already hold for incremental reachability. The first is Fredman and Saks's bound for prefix parity [FS89], which transfers to dynamic connectivity by a simple reduction observed independently by Miltersen, Subramanian, Vitter and Tamassia [MSVT94] and by Henzinger and Fredman [HF98].

The Boolean-answer restriction is central to this history. The Chronogram method separates information according to the epochs in which updates arrive. Larsen's cell-sampling argument forces many probes to an individual epoch, but its original encoding argument uses the many bits returned by a query. Larsen, Weinstein and Yu instead reduce a Boolean data structure to a one-way communication game in which the receiver predicts the answer with a small advantage over random guessing; a peak-to-average lemma based on Chebyshev polynomials bounds this advantage [LWY20]. This is the framework underlying Larsen and Yu's reachability bound.

For multiphase Disjointness itself, Pătraşcu's conjectured polynomial lower bound remains open. He proposed a three-party communication game whose hardness would imply the conjecture in the cell-probe model [Păt10]. Earlier progress includes bounds for non-adaptive data structures [BL15], regimes with very fast queries [CGL15], semi-adaptive data structures via two-round communication [KW20, DL20], and a 1.5-round version of the communication game [CEEP16]. Proving Pătraşcu's conjectured communication lower bound even for three-round protocols would imply major circuit lower bounds [KW20].

**The unconditional lower bounds strike back.** The new framework of [Ko26] is a common route to quadratic-logarithmic unconditional lower bounds for Boolean dynamic problems. Its 2.5-round game lets Bob verify Alice's simulation and detect its failure. Its first instance, multiphase Inner Product over $\mathbb{F}_2$, reaches the scale previously attained by Larsen for word-size answers [Lar12]. Beyond that instance, the framework gives a general recipe based on discrepancy bounds, initiating a program to unify dynamic lower bounds at this scale through communication hardness and reductions.

We take the next step in this program by extending the framework to Disjointness. This requires a further insight into the 2.5-round game: the hard distributions for Disjointness do not satisfy the discrepancy property used for Inner Product in [Ko26]. We exploit verification by conditioning on certified "disjoint" answers, allowing a one-sided corruption bound to replace discrepancy. This change matters beyond the source problem: Pătraşcu's reductions [Păt10] begin from Disjointness, and many later constructions also encode it. The extension therefore connects the 2.5-round framework to a broad collection of natural dynamic problems. In the spirit of Pătraşcu's program of reductions and communication lower bounds [Păt08, Păt10], the same constructions now transfer

a proved lower bound, independently of the conjectures that motivated them.

### 1.1.3 Streaming and amortization

Lower bounds for dynamic data structures are closely related to lower bounds on streaming processing time. A streaming membership algorithm updates its state as symbols arrive and reports after each arrival whether the current prefix belongs to a fixed language. This connection lets us apply our Disjointness bound to streaming regular-expression membership, studied by Bille and Thorup [BT10] and Dudek, Gawrychowski, Gourdel and Starikovskaya [DGGS22]. Earlier separations between amortized and worst-case time include union–find [Tar75, Blu85, FS89] and variance-optimal reservoir sampling [CDK+11]; we refer the reader to Appendix A.7 for the detailed discussion.

## 1.2 Our results

**Main result: multiphase Disjointness.** Our main technical result is an $\Omega\big((\log n/\log\log n)^2\big)$ lower bound for the multiphase problem of Pătraşcu [Păt10] with Disjointness, in its insert-only form. All applications in this paper follow from this theorem.

- *Phase I.* Sets $S_1, \ldots, S_m \subseteq [n]$ arrive and are preprocessed arbitrarily.
- *Phase II.* Elements are inserted one at a time into a set $X \subseteq [n]$, initially empty; each insertion probes at most $t_u$ cells.
- *Phase III.* An index $q \in [m]$ arrives, and the data structure outputs whether $S_q \cap X = \emptyset$, reading at most $t_q$ cells in expectation; it does not write.

**Problem 1:** Insert-only multiphase Disjointness

**Theorem 1.1** (Main technical theorem, informal)**.** *Every zero-error cell-probe data structure for insert-only multiphase Disjointness with $m = n$ query sets, word size $w = \Theta(\log n)$, worst-case update time $t_u$ per insertion and expected query time $t_q$ satisfies*

$$t_q = \Omega\left(\left(\frac{\log n}{\log(t_u w)}\right)^2\right).$$

*In particular,*

$$\max(t_u, t_q) = \Omega\left(\left(\frac{\log n}{\log\log n}\right)^2\right).$$

Theorem 3.1 gives the formal statement; Section 2.1 specifies the randomness and cost conventions.

Pătraşcu [Păt10] conjectured a polynomial lower bound for the multiphase problem when $m = n^{1+\Omega(1)}$ and showed that Disjointness serves as a common source of hardness for several natural dynamic problems. Our theorem does not resolve that conjecture, but reaches the quadratic-logarithmic scale already in the regime $m = n$. Ko [Ko26] and Koucký, Loff, Molli and Saks [KLMS26] discuss barriers to going beyond this scale with current techniques. We next give two consequences.

**Corollary I: dynamic problems.** Applying Pătraşcu's reduction gives a quadratic-logarithmic lower bound for incremental directed reachability, improving the bound of Larsen and Yu [LY25].

**Corollary 1.2** (Directed reachability, informal)**.** *Every zero-error cell-probe data structure for incremental directed reachability on $N$ vertices, starting with no edges, with word size $w = \Theta(\log N)$, worst-case insertion time $t_u$ and expected query time $t_q$, satisfies*

$$t_q = \Omega\left(\left(\frac{\log N}{\log(t_u w)}\right)^2\right);$$

*in particular,*

$$\max(t_u, t_q) = \Omega\left(\left(\frac{\log N}{\log\log N}\right)^2\right).$$

*This holds already on depth-2 DAGs whose queries ask whether a fixed sink is reachable.*

Corollary A.2 gives the formal statement. The bound trivially applies to fully dynamic directed reachability, since a fully dynamic data structure must handle insertion-only sequences.

The same $\Omega\big((\log N/\log\log N)^2\big)$ lower bound on $\max(t_u, t_q)$ follows for the dynamic problems in Table 1, which describes the tasks and gives representative references to prior work. The existing web of reductions gives these consequences their breadth, and we expect further applications beyond this list. Each consequence is an unconditional lower bound for its target problem, obtained by transferring proved hardness from Disjointness.

Table 1: Dynamic applications: brief descriptions and representative prior work.

| **Problem** | **Task** | **Prior work** |
|---|---|---|
| *Graphs (edge insertions unless stated otherwise)* | | |
| Directed reachability | Is one vertex reachable from another? Includes fixed $s$–$t$ queries and counting vertices reachable from a fixed source. | [Păt10, AW14] |
| Strong connectivity | Can two vertices reach one another along directed paths? | [HKM+12] |
| Directed cycle detection | Does the graph contain a directed cycle? | [HKM+12, CKL+24] |
| Undirected shortest paths | Return the distance between two vertices, possibly fixed vertices $s$ and $t$. | [Păt10, AW14] |
| Subgraph connectivity | Activate vertices of a fixed undirected graph; are two vertices connected in the induced subgraph? | [Păt10] |
| Triangle detection | Does an undirected graph contain a triangle? | [HKNS15] |
| Bipartite matching | Return the maximum number of vertex-disjoint edges, or test whether such edges can cover every vertex. | [AW14] |
| Exact densest subgraph | Return the maximum ratio of induced edges to vertices over nonempty vertex subsets. | [HKNS15] |
| Nearest marked vertex | In a fixed graph with changing vertex marks, return the distance from a queried vertex to the nearest marked vertex. | [HKNS15] |
| *Planar graphs (edge insertions)* | | |
| Undirected distance | Return the distance between two vertices; edge weights are unit or positive. | [AD16] |
| Directed girth | Return the length of a shortest directed cycle; edge weights are unit or positive. | [AD16] |
| Diameter | Return the largest distance between any two vertices in a graph with positive edge weights. | [AD16] |

Continued on the next page.

Table 1 (continued)

| Problem | Task | Prior work |
|---|---|---|
| Maximum-weight bipartite matching | Return the maximum total weight of vertex-disjoint edges in a bipartite graph with positive edge weights. | [AD16] |
| *Sets and arrays* | | |
| Set disjointness | Insert elements into a family of sets; are two queried sets disjoint? | [KPP15] |
| Range mode | Change individual array entries; return a most frequent value in a queried subarray. | [CDL+14] |
| Langerman's problem | Change individual entries of an integer array; does any nonempty prefix have sum zero? | [Păt10] |
| Pagh's problem | Add intersections of existing sets to a family; test whether a queried element belongs to a queried set. | [Păt10] |
| Erickson's problem | Increment an entire matrix row or column; return the largest matrix entry. | [Păt10, Ko26] |
| *Matrices over prime fields of size* $\Theta(N)$ | | |
| Inverse-entry zero testing | Maintain a nonsingular matrix under single-entry updates; is a specified inverse entry zero? | [vdBNS19] |
| Determinant | Change individual matrix entries; return the determinant or test whether it equals one. | [vdBNS19] |
| *Geometry* | | |
| 2D distinct-color counting | Insert colored points; count the distinct colors inside a queried axis-aligned rectangle. | [JX22] |
| 3D convex-hull vertex counting | Insert points; count the vertices of their convex hull. | [DI22] |
| 3D maximal-point counting | Insert points; count those for which no other point is at least as large in every coordinate. | [DI22, JX22] |
| *Strings* | | |
| Internal dictionary matching | Insert patterns specified as substrings of a fixed text; does a queried text interval contain an inserted pattern? | [CKM+19] |
| Binary Hamming-distance threshold | Change bits of a pattern; does it differ from an aligned substring of a fixed text in at most $k$ positions? The threshold $k$ is fixed and proportional to the pattern length. | [CGLS18] |
| Active-document existence | Turn on documents from a fixed collection; does an active document contain both queried patterns? | [JX22] |

These consequences follow from known reductions or simple adaptations of published constructions; for completeness and reader convenience, Appendix A gives the reductions and verifies their applicability to our precise model. The problem names in Table 1 link to their constructions, and Table 2 records the precise trade-offs. All statements use the zero-error model, worst-case update time, and per-input expected query time of Corollary 1.2, and many hold already under insertion-only updates. A reduction that answers the source query with target queries alone bounds $t_q$; one that also needs a constant number of target updates bounds $t_u + t_q$ (Lemma A.1).

**Corollary II: streaming membership, amortized versus worst case.** Our theorem also bounds the worst-case time per symbol of streaming regular-expression membership, a basic streaming task [BT10, DGGS22]. For each query set $S_i$, let $\Gamma_i$ be the alternation of the alphabet symbols

corresponding to elements outside $S_i$, and consider the regular expression

$$\bigcup_i (\#\,\Gamma_i^*\, c_i),$$

where $\cup$ denotes alternation and the final symbol $c_i$ selects the query; when $S_i = [n]$, the $i$th branch is simply $\#c_i$. Data symbols can be recorded in constant time, keeping only their first occurrence, while on the final selector one scans the resulting list of distinct symbols. Consequently, this family admits deterministic $O(1)$ amortized processing time per arriving symbol on every input stream.

A matcher with small worst-case processing time per symbol would yield a multiphase data structure with the same update and query bounds. Writes performed while processing the final selector can be simulated in temporary workspace, since no later operation follows.

**Corollary 1.3** (Streaming regular-expression membership, informal)**.** *There is a family of regular expressions of maximum length $M$ over a polynomial-size alphabet for which streaming membership, with an answer after every symbol, can be solved in deterministic $O(1)$ amortized time per arriving symbol on every input stream, but every zero-error cell-probe algorithm with word size $w = \Theta(\log M)$ requires*

$$\Omega\left(\left(\frac{\log M}{\log\log M}\right)^2\right)$$

*worst-case processing time on some symbol.*

Binary encoding gives a weaker $\Omega(\log M/(\log\log M)^2)$ bound per bit. The symbols in the original alphabet occupy $O(1)$ words each. Appendix A.7 gives both statements, their reductions, and a comparison with earlier separations.

## 1.3 Proof overview

We sketch the proof of Theorem 1.1, which follows the framework of [Ko26] and works in two layers. The outer layer is the Chronogram method of Fredman and Saks [FS89]: we split the insertions into $\Theta(\log n/\log(t_u w))$ epochs, and show that a query must probe $\Omega(\log n/\log(t_u w))$ cells written in each of a constant fraction of them. The inner layer proves this bound for a single epoch, following [Ko26]: if a query probes fewer cells of the epoch, we turn the data structure into a 2.5-round protocol whose probability of answering "disjoint" contradicts a communication lower bound. Section 3 gives the details.

### 1.3.1 The outer layer: the Chronogram method

The Chronogram method [FS89] divides the insertions into $\ell$ epochs whose sizes grow geometrically going back in time. We number the epochs $\ell, \ell-1, \ldots, 1$ in the order in which they occur, so that the larger epochs come first and epoch 1 is the most recent. The main observation of Fredman and Saks is that we can decompose the cells by the epoch of their last write: let $C_i$ be the set of cells last written in epoch $i$. Each probe goes to at most one of the disjoint sets $C_1, \ldots, C_\ell$, so for most epochs $i$, a random query probes only $O(t_q/\ell)$ cells of $C_i$ in expectation. We show that for a constant fraction of the epochs $i$, a random query must probe $\Omega(\log n/\log(t_u w))$ cells of $C_i$. With $\ell = \Theta(\log n/\log(t_u w))$, this gives $t_q = \Omega\big((\log n/\log(t_u w))^2\big)$.

**The hard distribution.** Recall that in multiphase Disjointness (Problem 1), the query sets $S_1, \ldots, S_m$ arrive first, the elements of $X$ are then inserted, and finally a query asks whether $S_q \cap X = \emptyset$. We choose disjoint blocks $B_1, \ldots, B_\ell$ in $[n]$ and leave any remaining coordinates unused. In epoch $i$, $s_i = (t_u w)^{2i}$ random elements of $B_i$ are inserted, and every query set has $s_i$ random elements in $B_i$. In other words, this is the hard distribution of Babai, Frankl and Simon [BFS86] for Disjointness, two independent random sets of size about the square root of the universe, spread across the epochs: each block carries one copy of it, made slightly sparser so that the $\ell$ copies together meet with constant probability. Two features of this distribution matter.

- *Recent epochs cannot record older ones.* The epoch sizes are geometric: going back in time, they grow by a factor $(t_u w)^2$. So the insertions after epoch $i$ number at most $s_{i-1} + s_{i-2} + \cdots + s_1 \leq 2s_{i-1} = 2s_i/(t_u w)^2$, and since each insertion writes at most $t_u$ cells of $w$ bits, they write only $O(s_i/(t_u w))$ bits. This is far too few to describe the $s_i$ elements inserted in epoch $i$, so a query that needs these elements must read cells of $C_i$. This is the key property of the Chronogram method.
- *Every epoch matters.* Block $B_i$ has $\ell s_i^2$ elements. By the birthday paradox, two random sets of size $s_i$ in a universe of size $\ell s_i^2$ meet with probability about $1/\ell$. So the answer is "disjoint" with constant probability, and each block can change it.

### 1.3.2 The inner layer: the 2.5-round game

Suppose, for contradiction, that $t_q = o\big((\log n/\log(t_u w))^2\big)$, and fix an epoch $i$ in which a random query probes only $k = O(t_q/\ell)$ cells of $C_i$ in expectation. The idea of cell sampling, introduced to the Chronogram method by Larsen [Lar12], is that a small random sample of $C_i$ then already suffices to answer a nontrivial fraction of the queries. If each cell of $C_i$ is sampled independently with probability $p = \Theta(1/(t_u w))$, then all $k$ cells of $C_i$ probed by a query are sampled with probability about $p^k$. Both $1/\ell$ and $\log(1/p)$ contribute a factor of $\log(t_u w)$, so

$$p^k \;\geq\; \exp\Big(-O\Big(\frac{t_q \log^2(t_u w)}{\log n}\Big)\Big) \;=\; n^{-o(1)}.$$

This is the source of the quadratic form of the bound. To exploit it, we use the following variant of the 2.5-round multiphase communication game of [Ko26]. Alice holds the query sets $S_1, \ldots, S_m$, Bob holds $X$, and an omniscient adviser, Merlin, holds both. Merlin follows the prescribed message rule, supplying advice determined by the inputs.

- *Round 0 (Merlin to Bob).* Merlin sends Bob a message $U$.
- *Round 0.5 (Bob to Alice).* Bob sends Alice a message $\Phi$, which may depend on $U$, $X$ and his private randomness, but not on the query.
- *Round 1 (Alice to Bob).* A random query index $Q \in [m]$ is revealed to Alice only, and she sends Bob a message $A_Q$.
- *Round 2 (Bob).* Bob outputs whether $S_Q \cap X = \emptyset$, or FAIL.

**Protocol 1:** The 2.5-round multiphase communication game (cf. [Ko26, Protocol 3])

This game is a restriction of Pătraşcu's multiphase communication game, in which Alice and Bob both know the query and may then communicate freely. The required lower bound for Pătraşcu's

game remains open, while known bounds for more restricted games [CEEP16, KW20, DL20] do not imply lower bounds for general data structures. The 2.5-round game sits in between. Bob's message $\Phi$ does not depend on the query, which makes the game amenable to information-theoretic analysis, and yet the game can still simulate a data structure.

**Simulation.** We turn the data structure into a protocol for the game as follows (Step 1, Section 3.2).

- Merlin's message $U$ is the list of all cells written in epochs $i, \dots, 1$, with their final contents.
- Bob's message $\Phi$ consists of the elements of $X$ outside $B_i$, the contents of all cells written after epoch $i$, and each cell of $C_i$ independently with probability $p = \varepsilon/(32t_u w)$, for a small constant $\varepsilon > 0$. With $\Phi$, Alice knows the whole memory except the unsampled cells of $C_i$.
- Alice's message $A_Q$ is her simulation of query $Q$ on the memory she knows.
- Bob checks $A_Q$ against $U$. Alice's simulation can only go wrong on unsampled cells of $C_i$, which are listed in $U$, so Bob detects every mistake. He outputs Alice's answer if $A_Q$ is consistent with $U$, and FAIL otherwise.

Every non-FAIL answer is therefore correct. Moreover, Alice's simulation is correct whenever all $k$ cells of $C_i$ that the query probes are sampled, which happens with probability about $p^k = n^{-o(1)}$.

Verification is what distinguishes this protocol from the one-way games behind earlier bounds [Lar12, LY25]. There, Alice answers on her own and cannot tell whether her simulation is correct, since an unsampled cell of $C_i$ looks to her like a cell that was not written in epoch $i$ at all; a failed simulation is indistinguishable from a successful one. Here, Bob knows exactly when the simulation fails.

The protocol nevertheless reveals little about block $i$. Bob's message reveals at most $\varepsilon s_i/4$ bits about $X \cap B_i$, a small fraction of its entropy, and Alice's simulation has $O(t_q w)$ bits. Merlin's list $U$ is long, but it is shared by all $m$ query sets, so it reveals little about the random query set $S_Q$. Yet Bob learns, with probability $n^{-o(1)}$, that $S_Q$ and $X$ are disjoint. The communication lower bound rules out this combination of small information loss and substantial probability of a certified "disjoint" answer.

### 1.3.3 From discrepancy to one-sided corruption

In [Ko26], Bob outputs a random guess when verification fails, and the contradiction comes from the protocol's advantage over random guessing. After conditioning on suitable message values and smoothing, the inputs have nearly full min-entropy and are nearly independent. For Inner Product the proof ends with a discrepancy bound: under every product distribution with nearly full min-entropy, the answer is nearly unbiased. That discrepancy property does not hold for Disjointness under the distributions needed here.

Verification allows us to use a weaker, one-sided property. [Ko26, Remark 3.4] already relates detected failure to the protocol's acceptance probability. We exploit the certified answer itself: Bob outputs FAIL instead of guessing, and we condition on his returning "disjoint". Every answer he returns is correct. We therefore need only that every product distribution with nearly full min-entropy puts noticeable mass on intersecting pairs. This one-sided corruption bound does hold for Disjointness: for sets of size $\sqrt{N}$ in a universe of size $N$, it is the classic rectangle bound of Babai, Frankl and Simon [BFS86], which we extend to our sparser sets in Section 4. Thus a one-sided corruption bound compatible with the epoch distribution replaces discrepancy. Extending the argument to another block function would also require a suitable product hard distribution with constant probability of the accepted global answer.

#### 1.3.4 The complete argument

Our proof proceeds in four steps. As in [Ko26], the first is a simulation, and the other three prove a communication lower bound.

1. **Simulation** (Section 3.2). A data structure violating the proposed bound yields a protocol for the game of Protocol 1 in which Bob outputs "disjoint" with probability $n^{-o(1)}$, and every answer he outputs is correct.

2. **A monochromatic near-rectangle** (Section 3.3). Among runs on which Bob returns "disjoint", choose a suitable value of the messages and auxiliary conditioning variables. Then restrict $S_Q \cap B_i$ to an event of constant conditional probability. After this smoothing step, the two block inputs have nearly full min-entropy and small mutual information. They are always disjoint because Bob's answer is certified.

3. **A nearly monochromatic rectangle** (Corollary 3.11). By Pinsker's inequality, replacing this distribution by the product of its marginals changes it little in total variation distance. The resulting rectangle therefore assigns little mass to intersecting pairs.

4. **Corruption** (Corollary 3.12). Every product of two distributions on sparse subsets of $B_i$ with nearly full min-entropy puts mass $\Omega(1/\ell)$ on intersecting pairs, a contradiction.

The data structure may also be randomized, with expected query time, as in [LY25]: we fix its coins, and Alice stops each simulation after $8t_q$ probes, which costs only a constant factor.

**Organization.** Section 2 gives the model conventions and information-theoretic preliminaries. Section 3 proves the main theorem using the sparse-set corruption bound established in Section 4. Appendix A contains the dynamic reductions and the streaming application.

## 2 Preliminaries

### 2.1 Cell-probe conventions

Cells and addresses have $w$ bits. Computation and temporary workspace within an operation are free; all input-dependent information retained between operations resides in memory and is accessed through charged probes. Preprocessing is unrestricted and uncharged. As in [LY25], a random tape is drawn independently of the input, correctness holds for every input and tape, update time is worst case over inputs and tapes, and query time is expected over the tape for each fixed valid input history. We normalize $t_u, t_q \geq 1$, take $t_u$ to be an integer, and state asymptotic bounds for sufficiently large instances.

### 2.2 Information theory

We collect the information-theoretic facts used in the proof; see [CT06] for background. Logarithms are base 2 unless written ln.

**Definition 2.1** (Entropy, mutual information, divergence)**.** The entropy of a random variable $X$ is $H(X) := \sum_x \Pr[X = x] \log \frac{1}{\Pr[X=x]}$, and the conditional entropy is $H(X|Y) := \mathbb{E}_{y\sim Y}\left[H(X|Y = y)\right]$. The mutual information between $X$ and $Y$, conditioned on $Z$, is $I(X;Y|Z) := H(X|Z) - H(X|Y,Z)$;

when we need to specify the distribution $\mu$ of $(X, Y)$, we write $I_\mu(X;Y)$. The KL divergence and the total variation distance between two distributions $P$ and $Q$ are

$$D(P \,\|\, Q) := \sum_x P(x) \log \frac{P(x)}{Q(x)} \qquad \text{and} \qquad \|P - Q\|_{TV} := \frac{1}{2}\|P - Q\|_1.$$

For a random variable $X$ and an event $E$, we write $X|_E$ for the distribution of $X$ conditioned on $E$, for instance $X|_{Y=y}$. A random variable in place of a distribution, as in $D(X \,\|\, Y)$, stands for its distribution.

**Fact 2.2** (Basic properties)**.** *For any random variables $A$, $B$, $C$ and $D$:*

*(a) If $A$ takes at most $2^\lambda$ values, then $H(A) \le \lambda$. Moreover, $0 \le H(A|B) \le H(A)$.*

*(b) $I(A;B|C) = I(B;A|C)$ and $0 \le I(A;B|C) \le H(A|C)$. If $A$ and $B$ are independent given $C$, for instance if $A$ is a function of $C$, then $I(A;B|C) = 0$.*

*(c) $I(A;B|C) = \mathbb{E}_{c\sim C}\big[I(A;B|C=c)\big]$.*

*(d) $D(P \,\|\, Q) \ge 0$ for any two distributions $P$ and $Q$.*

*(e) If $B'$ is a function of $B$, then $I(A;B'|C) \le I(A;B|C)$.*

*(f) If $D$ is independent of $(A,B,C)$, then $I(A;B|C,D) = I(A;B|C)$.*

**Fact 2.3** (Chain rules)**.** *(a) For any random variables $A_1, \dots, A_k$, $B$ and $C$,*

$$I(A_1 \cdots A_k; B|C) \;=\; \sum_{j=1}^{k} I\big(A_j; B \;\big|\; CA_1 \cdots A_{j-1}\big).$$

*(b) For any two pairs of random variables $(X,Y)$ and $(X',Y')$,*

$$D\big(XY \,\|\, X'Y'\big) \;=\; D\big(X \,\|\, X'\big) + \mathop{\mathbb{E}}_{x\sim X}\Big[D\big(Y|_{X=x} \,\big\|\, Y'|_{X'=x}\big)\Big].$$

**Fact 2.4.** *For any random variables $A$, $B$, $C$ and $D$, if $I(B;D|AC) = 0$, then $I(A;B|CD) \le I(A;B|C)$.*

*Proof.* By the chain rule (Fact 2.3(a)) and Fact 2.2(b),

$$I(A;B|CD) \;\le\; I(AD;B|C) \;=\; I(A;B|C) + I(B;D|AC) \;=\; I(A;B|C). \qquad \square$$

**Fact 2.5** (KL divergence and mutual information)**.** *For any random variables $A$ and $B$, $I(A;B) = \mathbb{E}_{b\sim B}\big[D\big(A|_{B=b} \,\|\, A\big)\big]$.*

**Fact 2.6** (Pinsker's inequality)**.** *For any two distributions $P$ and $Q$,*

$$\|P - Q\|_{TV} \;\le\; \sqrt{(\ln 2/2)\, D(P \,\|\, Q)}.$$

### 2.3 Min-entropy

The min-entropy of a random variable $A$ is $H_\infty(A) := -\log \max_a \Pr[A = a]$, and for a distribution $\nu$ we write $H_\infty(\nu)$ likewise. We use the average min-entropy of [DORS08],

$$\widetilde{H}_\infty(A|B) := -\log \mathop{\mathbb{E}}_{b\sim B}\Big[\max_a \Pr[A = a|B = b]\Big].$$

**Fact 2.7** ([DORS08])**.** *Let $A$, $B$, $C$ be random variables.*

*(a) If $B$ has at most $2^\lambda$ possible values, then $\widetilde{H}_\infty(A|B, C) \geq \widetilde{H}_\infty(A|C) - \lambda$.*

*(b) For every $r > 0$, $\Pr_{b\sim B}\left[H_\infty(A|B = b) \geq \widetilde{H}_\infty(A|B) - r\right] \geq 1 - 2^{-r}$.*

*(c) If $B'$ is a function of $B$, then $\widetilde{H}_\infty(A|B') \geq \widetilde{H}_\infty(A|B)$.*

**Fact 2.8.** *Let $E \in \{0, 1\}$ be a random variable with $\Pr[E = 1] > 0$. Then $\widetilde{H}_\infty(X|Y, E{=}1) \geq \widetilde{H}_\infty(X|Y) - \log(1/\Pr[E = 1])$, where the left side is with respect to the distribution conditioned on $E = 1$.*

*Proof.* For every $y$ with $\Pr[E = 1|Y = y] > 0$, we have $\Pr[X = x|Y = y, E = 1] \leq \Pr[X = x|Y = y]/\Pr[E = 1|Y = y]$ and, by Bayes' rule, $\Pr[Y = y|E = 1]/\Pr[E = 1|Y = y] = \Pr[Y = y]/\Pr[E = 1]$. Summing over such $y$, and then over all $y$,

$$\mathop{\mathbb{E}}_{y\sim Y|E=1} \max_x \Pr[X = x|Y = y, E = 1] \;\leq\; \sum_y \frac{\Pr[Y = y]\max_x \Pr[X = x|Y = y]}{\Pr[E = 1]} \;=\; \frac{2^{-\widetilde{H}_\infty(X|Y)}}{\Pr[E = 1]}.$$

□

**Fact 2.9** (Conditioning on one side)**.** *Let $A, B$ be random variables, and let $E \in \{0, 1\}$ be a function of $A$ with $\pi := \Pr[E = 1] > 0$. Then*

$$I(A; B|E{=}1) \leq \frac{I(A; B)}{\pi} \qquad \text{and} \qquad H_\infty(B|E{=}1) \geq H_\infty(B) - \log\frac{1}{\pi},$$

*where the left sides are with respect to the distribution conditioned on $E = 1$.*

*Proof.* Since $E$ is a function of $A$, $I(B; E|A) = 0$, so Fact 2.4 and Fact 2.2(b) and (c) give

$$I(A; B) \;\geq\; I(A; B|E) \;\geq\; \pi \cdot I(A; B|E{=}1).$$

For the second bound, $\Pr[B = b|E = 1] \leq \Pr[B = b]/\pi$ for every $b$. □

The next two facts relate min-entropy to flat distributions and Shannon entropy. For a finite set $\mathcal{S}$, $\mathcal{U}_\mathcal{S}$ denotes the uniform distribution on $\mathcal{S}$; thus $D(\mu \,\|\, \mathcal{U}_\mathcal{S}) = \log|\mathcal{S}| - H(\mu)$.

**Fact 2.10** (Flat decomposition; folklore, see e.g. [Vad12, Lemma 6.10])**.** *If $H_\infty(\mu) \geq \log K$ for an integer $K \geq 1$, then $\mu$ is a convex combination of uniform distributions $\mathcal{U}_T$ on $K$-element sets $T$.*

**Fact 2.11** (Smoothing [JRS09]; see [Ko26, Claim 2.16])**.** *If $D(\mu \,\|\, \mathcal{U}_\mathcal{S}) \leq t$ with $t > 1$, then for every $\alpha \geq 4$ there exists an event $E$ such that $\mu(E) \geq 1 - 1/\alpha$ and $H_\infty(\mu|_E) \geq \log|\mathcal{S}| - 2\alpha t$.*

## 2.4 Rectangles

We express the size of a combinatorial rectangle $\mathcal{A} \times \mathcal{B}$ through min-entropy. This lets us treat product distributions and distributions that are close to products in a common notation.

**Definition 2.12** (Deficit)**.** Let $\mathcal{S}$ be a finite family of sets. The *deficit* of a distribution $\nu$ on $\mathcal{S}$ is $\log|\mathcal{S}| - H_\infty(\nu)$.

The deficit is zero exactly for the uniform distribution on $\mathcal{S}$, and the uniform distribution on a subfamily $\mathcal{A} \subseteq \mathcal{S}$ has deficit $\log(|\mathcal{S}| / |\mathcal{A}|)$. A deficit bound of $d$ therefore limits the mass of every point to $2^d/|\mathcal{S}|$. It measures how much the distribution can concentrate relative to uniform. Below, $(A, B) \sim \mu$ for a distribution $\mu$ on $\mathcal{S} \times \mathcal{S}$ with marginals $\mu^A, \mu^B$.

**Definition 2.13** (Rectangles, near-rectangles, monochromatic)**.** Let $d, \kappa, \delta \geq 0$.

(a) $\mu$ is a *$d$-rectangle* if $\mu = \mu^A \otimes \mu^B$ and both marginals have deficit at most $d$. That is,

$$H_\infty(\mu^A) \;\geq\; \log|\mathcal{S}| - d \qquad \text{and} \qquad H_\infty(\mu^B) \;\geq\; \log|\mathcal{S}| - d.$$

(b) $\mu$ is a *$(d, \kappa)$-near-rectangle* if both marginals have deficit at most $d$ and $I_\mu(A; B) \leq \kappa$.

(c) $\mu$ is *$\delta$-monochromatic* if $\Pr_\mu[A \cap B \neq \emptyset] \leq \delta$, and *monochromatic* if it is 0-monochromatic, i.e. supported on disjoint pairs.

A $d$-rectangle is the distributional analog of a large combinatorial rectangle. For $\mathcal{A}, \mathcal{B} \subseteq \mathcal{S}$, the uniform distribution $\mathcal{U}_{\mathcal{A}\times\mathcal{B}} = \mathcal{U}_\mathcal{A} \otimes \mathcal{U}_\mathcal{B}$ is a $d$-rectangle if and only if $|\mathcal{A}|, |\mathcal{B}| \geq 2^{-d}|\mathcal{S}|$, since $\mathcal{U}_\mathcal{A}$ has deficit $\log(|\mathcal{S}| / |\mathcal{A}|)$. Conversely, every $d$-rectangle is a convex combination of uniform distributions on combinatorial rectangles of almost this size.

**Corollary 2.14** (Flat decomposition of rectangles)**.** *Let $\mu$ be a $d$-rectangle on $\mathcal{S} \times \mathcal{S}$, and let $K$ be an integer with $1 \leq K \leq 2^{-d}|\mathcal{S}|$. Then there are subfamilies $\mathcal{A}_j, \mathcal{B}_k \subseteq \mathcal{S}$ of size $K$ and weights $p_j, q_k \geq 0$ with $\sum_j p_j = \sum_k q_k = 1$ such that*

$$\mu \;=\; \sum_{j,k} p_j q_k \, \mathcal{U}_{\mathcal{A}_j \times \mathcal{B}_k}.$$

*Proof.* Both marginals of $\mu$ have min-entropy at least $\log|\mathcal{S}| - d \geq \log K$. So by Fact 2.10, $\mu^A = \sum_j p_j \mathcal{U}_{\mathcal{A}_j}$ and $\mu^B = \sum_k q_k \mathcal{U}_{\mathcal{B}_k}$ for such subfamilies and weights. Therefore

$$\mu \;=\; \mu^A \otimes \mu^B \;=\; \sum_{j,k} p_j q_k \, \mathcal{U}_{\mathcal{A}_j} \otimes \mathcal{U}_{\mathcal{B}_k}. \qquad \square$$

A near-rectangle relaxes the product condition: its two sides may be correlated, and $\kappa$ bounds their mutual information. In particular, a $d$-rectangle is a $(d, 0)$-near-rectangle. Replacing a near-rectangle by the product of its marginals gives a rectangle that is close in total variation distance (see, e.g., [Ko26]).

**Lemma 2.15** (From near-rectangles to rectangles)**.** *If $\mu$ is a $\delta$-monochromatic $(d, \kappa)$-near-rectangle, then $\mu^A \otimes \mu^B$ is a $\left(\delta + \sqrt{(\ln 2/2)\,\kappa}\right)$-monochromatic $d$-rectangle.*

*Proof.* Let $R := \mu^A \otimes \mu^B$. It has the same marginals as $\mu$, so it is a $d$-rectangle. By the chain rule for KL divergence (Fact 2.3(b)) and Fact 2.5, $D(\mu \,\|\, R) = \mathbb{E}_{b\sim\mu^B}\left[D(A|_{B=b} \,\|\, \mu^A)\right] = I_\mu(A; B) \leq \kappa$, so $\|\mu - R\|_{TV} \leq \sqrt{(\ln 2/2)\,\kappa}$ by Fact 2.6. Therefore

$$\Pr_{(A,B)\sim R}\left[A \cap B \neq \emptyset\right] \;\leq\; \Pr_{(A,B)\sim\mu}\left[A \cap B \neq \emptyset\right] + \|\mu - R\|_{TV} \;\leq\; \delta + \sqrt{(\ln 2/2)\,\kappa}. \qquad \square$$

# 3 Main proof

In this section, we prove our main theorem, the formal version of Theorem 1.1.

**Theorem 3.1** (Main theorem)**.** *Every zero-error cell-probe data structure for insert-only multiphase Disjointness with $m = n$ query sets, word size $w = \Theta(\log n)$, worst-case update time $t_u$ and expected query time $t_q$ has*

$$t_q = \Omega\left(\left(\frac{\log n}{\log(t_u w)}\right)^2\right).$$

*In particular,* $\max(t_u, t_q) = \Omega\big((\log n/\log\log n)^2\big)$.

We follow the outline in Section 1.3. After defining the hard distribution, we turn a data structure violating the bound into a 2.5-round protocol (Section 3.2). The communication lower bound in Section 3.3 rules out this protocol, using the corruption bound proved in Section 4.

## 3.1 The hard distribution

Let $\beta := (t_u w)^4$, and choose the largest integer $\ell \geq 1$ such that $\ell \sum_{i=1}^{\ell} \beta^i \leq n$. The bound is trivial when $t_u w > n^{1/95}$, so we work in the complementary regime; in particular, such an $\ell$ exists. Choose pairwise disjoint blocks $B_1, \ldots, B_\ell \subseteq [n]$ with

$$|B_i| = n_i := \ell\beta^i, \qquad \text{and let} \qquad s_i := \sqrt{n_i/\ell} = \beta^{i/2}.$$

The remaining coordinates occur in neither the query sets nor $X$. We keep $m = n$ query sets. Since $t_u$ and $w$ are integers, all block and set sizes are integers. Epochs are indexed backward in time: epoch $\ell$ comes first and epoch 1 comes last. Under the distribution $\mathcal{D}$ of $(\vec{S}, X)$, independently:

- each query set $S_j$, $j \in [m]$, consists of $s_i$ uniformly random elements of each block $B_i$; write $S_j^{(i)} := S_j \cap B_i$ and $\vec{S} := (S_1, \ldots, S_m)$;
- in epoch $i$, $s_i$ uniformly random elements of $B_i$ are inserted into $X$, in increasing order. We write $X^{(i)} := X \cap B_i$ for the set inserted in epoch $i$, $X^{(>i)} := X \cap (B_{i+1} \cup \cdots \cup B_\ell)$, and $X^{(\neq i)} := X \setminus B_i$.

The query index $Q$ is uniform in $[m]$ and independent of $(\vec{S}, X)$. We write $\Pr_{\mathcal{D},Q}$ and $\mathbb{E}_{\mathcal{D},Q}$ for probability and expectation over $(\vec{S}, X) \sim \mathcal{D}$ and $Q$. In each block, $S_Q$ and $X$ share $s_i^2/n_i = 1/\ell$ elements in expectation over $\mathcal{D}$ and $Q$; since the blocks are independent, $S_Q$ and $X$ are disjoint with probability at least $1/4$ (Claim 3.2(d) and Fig. 2). The *Chronogram epochs* are $\mathcal{I} := \{i \in [\ell] : i \geq 2(\ell+2)/3\}$; any range of $\Omega(\ell)$ epochs with $s_i \geq n^{\Omega(1)}$ would do.

The following bounds summarize the parameters of the hard distribution.

**Claim 3.2** (Parameters)**.** *The hard distribution has the following properties.*

*(a)* $\ell = \Theta(\log n/\log(t_u w))$, $\ell \geq 20$, *and* $|\mathcal{I}| \geq (\ell - 4)/3$.

*(b) For* $i \in \mathcal{I}$: $s_i \geq n^{1/3}$.

*(c)* $s_i/s_{i-1} = \sqrt{\beta} = (t_u w)^2$, $\sum_{j<i} s_j \leq 2s_{i-1}$, *and* $s_\ell \leq \sqrt{n}$.

*(d)* $\mathbb{E}_{\mathcal{D},Q}\left[\left|S_Q^{(i)} \cap X^{(i)}\right|\right] = s_i^2/n_i = 1/\ell$ *for every* $i$, *and* $\Pr_{\mathcal{D},Q}[S_Q \cap X = \emptyset] \geq (1 - 1/\ell)^\ell \geq 1/4$ *for* $\ell \geq 2$.

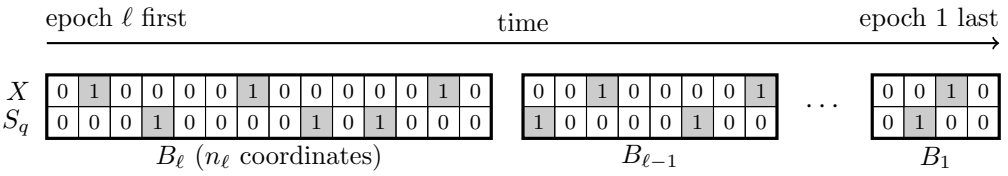


Figure 2: The hard distribution (not to scale), with $X$ and $S_q$ written as characteristic vectors. The ones of $X$ in block $B_i$ are inserted during epoch $i$. Block sizes grow by a factor $\beta = (t_u w)^4$ going back in time. In each block, the expected number of common ones is $s_i^2/n_i = 1/\ell$, so $S_Q$ and $X$ are disjoint with constant probability.

*Proof.* We verify the parameter bounds, including the unused coordinates.

(a) By maximality of $\ell$ and $\beta \geq 2$,

$$\ell\beta^\ell \leq n < (\ell+1)\sum_{j=1}^{\ell+1}\beta^j \leq 2(\ell+1)\beta^{\ell+1}.$$

Thus $\ell \leq \log n$, and $\ell = \Theta(\log n/\log\beta)$. Also $|\mathcal{I}| \geq \ell - 2(\ell+2)/3 = (\ell-4)/3$. Since $t_u w \leq n^{1/95}$, we have $\log\beta \leq (4/95)\log n$ and

$$\ell > \frac{\log n - \log(2(\ell+1))}{\log\beta} - 1 \geq \frac{95}{4}\left(1 - \frac{\log(2(\log n+1))}{\log n}\right) - 1 \geq 20.$$

(b) Since $w = \Theta(\log n)$ and $\ell \leq \log n$, we have $\beta \geq 2(\ell+1)$. For $i \geq 2(\ell+2)/3$, $s_i = \beta^{i/2} \geq (\beta^{\ell+2})^{1/3} > (\beta n/(2(\ell+1)))^{1/3} \geq n^{1/3}$.

(c) The ratio is immediate; $\sum_{j<i} s_j \leq s_{i-1}\sum_{r\geq 0}\beta^{-r/2} \leq 2s_{i-1}$ since $\beta \geq 4$; and $s_\ell^2 = n_\ell/\ell \leq n$.

(d) Each element of $B_i$ lies in $S_Q^{(i)}$ with probability $s_i/n_i$ and in $X^{(i)}$ with probability $s_i/n_i$, independently, so the expectation is $n_i(s_i/n_i)^2 = 1/\ell$. By Markov, block $i$ is disjoint with probability at least $1 - 1/\ell$, and the blocks are independent. □

**Setup.** Fix the random tape by averaging over the hard distribution. We may assume that $t_u w \leq n^{1/95}$ and $t_q \leq \log^2 n$, since otherwise the bound is trivial. So the data structure is deterministic, with zero error and worst-case update time $t_u$, and the number $T(q)$ of cells read by a query $q \in [m]$ is a function of $(\vec{S}, X)$ with $\mathbb{E}_{\mathcal{D},Q}[T(Q)] \leq t_q$. By Claim 3.2, $\ell \geq 20$ and $s_i \geq n^{1/3}$ for $i \in \mathcal{I}$. Since $w = \Theta(\log n)$, also $t_u w \geq 64/\varepsilon$, where $\varepsilon = 1/40$ is the constant of Lemma 4.1.

**Epochs.** The updates are processed in decreasing order of the epoch index, so the larger epochs come first. We follow the notation and the setup of [LY25]. For $i \in \mathcal{I}$, let $M_{>i}$ be the memory after Phase I and epochs $\ell, \ldots, i+1$; it is a function of $(\vec{S}, X^{(>i)})$. For $j \in [\ell]$, let $C_j$ be the set of cells last written in epoch $j$, and let $C_{<j} := \bigcup_{j'<j} C_{j'}$ be the set of cells written after epoch $j$. The sets $C_1, \ldots, C_\ell$ are pairwise disjoint. Let $t_j(q)$ be the number of distinct cells of $C_j$ read by query $q$, so that

$$\sum_{j\in[\ell]} t_j(q) \;\leq\; T(q), \qquad \text{and let} \qquad \tau_j \;:=\; \mathop{\mathbb{E}}_{\mathcal{D},Q}\big[t_j(Q)\big].$$

### 3.2 Simulation theorem

The following theorem converts a data structure into a protocol for the 2.5-round game of Protocol 1 (cf. [Ko26, Protocol 3]), in which Bob verifies Alice's simulation of the query against the actual memory. We adapt the protocol of [Ko26, Protocol 4] and the analysis of [Ko26, Theorem 3.2]; numbered references to that work use arXiv version 2. The main difference is Bob's output when the simulation fails: in [Ko26], he outputs FAIL together with an independent random guess, while here he outputs only FAIL. In addition, the epoch $i$ is fixed in advance rather than chosen by Merlin, Merlin's message lists only the cells written in epochs $i, \dots, 1$ rather than all updates, Bob sends Alice all elements of $X$ outside $B_i$, and, since $t_q$ bounds the query time only on average, Alice stops her simulation after $8t_q$ probes.

Let $U$ consist of two lists, of the cells of $C_i$ and of the cells of $C_{<i}$, each cell with its final content, that is, its content after Phase II. Let $p := \varepsilon/(32\, t_u w)$. Bob's private randomness $R_{\mathrm{Bob}}$ consists of an independent bit $R_a$ with $\Pr[R_a = 1] = p$ for every cell $a$, and $(\vec{S}, X)$, $Q$ and $R_{\mathrm{Bob}}$ are mutually independent. The length $|\cdot|$ of a message is measured in bits, padded to a fixed length, so that a message of length $\lambda$ takes at most $2^\lambda$ values.

**Theorem 3.3** (Simulation theorem)**.** *For every $i \in \mathcal{I}$ there is a 2.5-round protocol $G_i$ for the game of Protocol 1, in which Merlin sends $U$ and Bob's message has the form $\Phi = (X^{(\neq i)}, \Psi)$, with the following properties, where $B_Q$ is the indicator that Bob outputs "disjoint" and $\rho_1 := \Pr_{\mathcal{D}, R_{\mathrm{Bob}}, Q}[B_Q = 1]$.*

*(a)* (Zero error) *If Bob does not output* FAIL*, his output is the correct answer. In particular, $B_Q = 1$ implies $S_Q \cap X = \emptyset$.*

*(b)* (Lengths of $U$ and $\Psi$) *$|U| \le 5\, t_u w\, s_i$ and $|\Psi| \le \varepsilon s_i/4$.*

*(c)* (Length of Alice's messages) *$|A_q| \le 17\, t_q w$ for every $q \in [m]$.*

*(d)* (Probability of answering disjoint) *$\rho_1 \ge \frac{1}{8}\, p^{8\tau_i} - e^{-n^{1/4}/3}$.*

*(e)* (Structure) *$\Psi$ is a function of $(U, R_{\mathrm{Bob}})$, each $A_q$ is a function of $(\vec{S}, \Phi)$, and $B_Q$ is a function of $(\Psi, A_Q, U)$.*

*Proof.* We simulate the data structure by the 2.5-round protocol $G_i$ of Protocol 2. A verified transcript may have either answer; $B_Q = 1$ denotes only a verified "disjoint" answer. Part (e) is immediate from Protocol 2: Bob computes $\Psi$ from $U$ and $R_{\mathrm{Bob}}$, Alice computes $A_q$ from $\vec{S}$ and $\Phi$, and Bob's decision reads only $\Psi$, which records whether he aborted, $A_Q$, which records whether the run finished and its output, and the final contents listed in $U$.

**Cost of the protocol.** Each cell is encoded by its $w$-bit address and $w$-bit content. We encode list lengths and the abort flag using $O(w)$ additional bits, and pad messages to fixed lengths. Merlin's message $U$ lists the cells of $C_i$ and of $C_{<i}$. Here $|C_i| \le t_u s_i$, since each of the $s_i$ insertions of epoch $i$ writes at most $t_u$ cells, and similarly $|C_{<i}| \le t_u \sum_{j<i} s_j \le 2t_u s_{i-1}$ by Claim 3.2(c). Bob's list $\Psi$ has at most $|C_{<i}| + 2p\, t_u s_i$ cells. Since $s_{i-1} = s_i/(t_u w)^2$ and $t_u w \ge 64/\varepsilon$,

$$
\begin{aligned}
|U| &\le 2w\big(\,|C_i| + |C_{<i}|\,\big) + O(w) \le 2w\big(t_u s_i + 2t_u s_{i-1}\big) + O(w) \le 5\, t_u w\, s_i, \\
|\Psi| &\le 2w\big(2t_u s_{i-1} + 2p\, t_u s_i\big) + O(w) = \frac{4 s_i}{t_u w} + \frac{\varepsilon s_i}{8} + O(w) \le \frac{\varepsilon s_i}{4}.
\end{aligned}
$$

- Alice holds $\vec{S}$. Bob holds $X$ and $R_{\text{Bob}}$. Merlin holds $\vec{S}$ and $X$.
- Merlin sends $U$ to Bob.
- As the 0.5-round message, Bob lets $\widetilde{C}_i := \{a \in C_i : R_a = 1\}$. If $\left|\widetilde{C}_i\right| > 2p\, t_u s_i$, Bob aborts: he sets $\Psi := \perp$, and outputs FAIL at the end. Otherwise $\Psi$ lists the cells of $C_{<i} \cup \widetilde{C}_i$ with their final contents. Bob sends $\Phi := (X^{(\neq i)}, \Psi)$ to Alice. The message depends on $U$, $X$ and $R_{\text{Bob}}$, but not on the query $Q$.
- The query $Q \in [m]$ is then revealed to Alice. Alice computes $M_{>i}$ from $\vec{S}$ and $X^{(>i)}$, and runs the query algorithm on $Q$. She answers a probe to a cell listed in $\Psi$ with its listed content, and a probe to any other cell with its content in $M_{>i}$. She stops the run if it has not finished after $\lfloor 8t_q \rfloor$ probes. She sends the transcript

  $$A_Q \;:=\; \big((a_1, c_1), \ldots, (a_r, c_r), z_{out}\big), \qquad r \le \lfloor 8t_q \rfloor,$$

  where $a_k$ is the $k$-th probed cell, $c_k$ the content she used, and $z_{out}$ the output, or a mark that the run was stopped.
- Bob *verifies* $A_Q$ if (i) he did not abort, (ii) the run recorded in $A_Q$ finished, and (iii) $c_k$ is the final content of $a_k$ for every $k$ with $a_k \in C_i \cup C_{<i}$; he reads these final contents off $U$. If verification succeeds, he outputs $z_{out}$; otherwise he outputs FAIL.

Protocol 2: Simulation protocol $G_i$ (cf. [Ko26, Protocol 4])

Alice's transcript records at most $8t_q$ probes, together with the number of probes and the output or the stop mark, so

$$|A_q| \;\le\; 16\, t_q w + \lceil \log(\lfloor 8t_q \rfloor + 1) \rceil + 2 \;\le\; 17\, t_q w.$$

The header costs are absorbed since $s_i \ge n^{1/3}$ and $w = \Theta(\log n)$. This proves (b) and (c).

**Verification and disjoint answers.** We first prove that verification certifies the true execution and succeeds when the sampled cells cover that execution, provided Bob does not abort and the query finishes within the probe limit. We then restrict to disjoint inputs to prove (d).

**Claim 3.4** (Verification)**.** *Fix $(\vec{S}, X)$ and $q \in [m]$. In (b) and (c), the probability is over $R_{\text{Bob}}$.*

*(a) If Bob verifies $A_q$, then $A_q$ is the true execution of query $q$. In particular, its output is* $\text{DISJ}(S_q, X)$.

*(b)* $\Pr_{R_{\text{Bob}}}[\textit{Bob aborts}] \le e^{-n^{1/4}/3}$.

*(c) If $T(q) \le 8t_q$, then* $\Pr_{R_{\text{Bob}}}[\textit{Bob verifies } A_q] \ge p^{t_i(q)} - e^{-n^{1/4}/3}$.

*Proof.* (a) If Bob verifies $A_q$, then every content Alice used is the final content. For a cell of $C_i \cup C_{<i}$, Bob checked this. Any other cell is not listed in $\Psi$ and is not written during epochs $i, \ldots, 1$, so Alice used its content in $M_{>i}$, which is final. Hence her run is the true execution of query $q$, and its output is $\text{DISJ}(S_q, X)$.

(b) $\left|\widetilde{C}_i\right|$ is a sum of $|C_i| \le t_u s_i$ independent indicators with mean $p$ each. By the Chernoff bound, applied with the upper bound $p\, t_u s_i$ on the mean,

$$\Pr_{R_{\text{Bob}}}\left[\left|\widetilde{C}_i\right| > 2p\, t_u s_i\right] \;\le\; e^{-p\, t_u s_i/3},$$

and $p\, t_u s_i = \varepsilon s_i/(32w) \geq n^{1/4}$.

(c) Let $P_q \subseteq C_i$ be the set of the $t_i(q)$ cells of $C_i$ probed by the true execution of query $q$. If Bob does not abort and $P_q \subseteq \widetilde{C}_i$, then every cell that the true execution probes is either listed in $\Psi$ or outside $C_i \cup C_{<i}$, so Alice reproduces the true execution, which finishes within $T(q) \leq 8t_q$ probes, and verification succeeds. Since $\Pr_{R_{\mathrm{Bob}}}[P_q \subseteq \widetilde{C}_i] = p^{t_i(q)}$, a union bound with (b) gives (c). □

Part (a) of the theorem follows from Claim 3.4(a), since Bob outputs FAIL unless he verifies $A_Q$. It remains to show (d). Let $F$ be the event that $S_Q \cap X = \emptyset$ and $T(Q) \leq 8t_q$; it is determined by $(\vec{S}, X, Q)$. By Markov's inequality, $\Pr_{\mathcal{D},Q}[T(Q) > 8t_q] \leq 1/8$, and by Claim 3.2(d), $\Pr_{\mathcal{D},Q}[S_Q \cap X = \emptyset] \geq 1/4$, so $\Pr_{\mathcal{D},Q}[F] \geq 1/8$. If $F$ holds and Bob verifies $A_Q$, then the recorded output is "disjoint" by Claim 3.4(a), so $B_Q = 1$. Since $R_{\mathrm{Bob}}$ is independent of $(\vec{S}, X, Q)$, Claim 3.4(c), applied for each fixed $(\vec{S}, X, Q)$, and then the convexity of $x \mapsto p^x$ (with Jensen's inequality) give

$$\rho_1 \;\geq\; \mathop{\mathbb{E}}_{\mathcal{D},Q}\Big[\mathbf{1}_F \cdot \big(p^{t_i(Q)} - e^{-n^{1/4}/3}\big)\Big] \;\geq\; \Pr_{\mathcal{D},Q}[F] \cdot p^{\mathbb{E}_{\mathcal{D},Q}[t_i(Q)|F]} - e^{-n^{1/4}/3}.$$

Since $t_i \geq 0$, $\mathbb{E}_{\mathcal{D},Q}[t_i(Q)|F] \leq \tau_i / \Pr_{\mathcal{D},Q}[F] \leq 8\tau_i$, which gives (d). □

As in [Ko26, Remark 3.4], the protocol detects simulation failure: whenever Bob does not output FAIL, he has verified Alice's transcript against $U$, and his output is correct. The remaining runs play no role in the argument: Steps 2–4 use only the event $B_Q = 1$, on which the answer is "disjoint" with certainty.

The next corollary quantifies the probability of answering "disjoint" when a query reads few cells from epoch $i$.

**Corollary 3.5** (Choice of parameters)**.** *If $\tau_i = o\big(\log n / \log(t_u w)\big)$, then $\rho_1 \geq n^{-o(1)}$.*

*Proof.* Since $t_u w \geq 64/\varepsilon$, $\log(1/p) = \log(32\, t_u w/\varepsilon) \leq 2\log(t_u w)$, so $p^{8\tau_i} \geq 2^{-16\tau_i \log(t_u w)} = n^{-o(1)}$. By Theorem 3.3(d), $\rho_1 \geq \frac{1}{8} n^{-o(1)} - e^{-n^{1/4}/3} \geq n^{-o(1)}$. □

### 3.3 Communication lower bound

We prove that $\rho_1 < n^{-1/6}$ for every $i \in \mathcal{I}$ (Lemma 3.13). Following [Ko26, KW20], we collect the relevant messages and auxiliary variables into $Z$, including Bob's output $B_Q$. If disjoint answers occur too often, we can choose a value $Z = (z, 1)$ and then restrict the query-side input $S_Q^{(i)}$ to an event $E$ of constant conditional probability. Under this joint conditioning, both block inputs have nearly full min-entropy and their mutual information is small. The smoothing event $E$ depends only on the query-side input once $z$ is fixed.

Every pair in the resulting distribution is disjoint, by Theorem 3.3(a). Thus it is a monochromatic near-rectangle (Definition 2.13). The one-sided corruption bound rules out this distribution after we pass to the product of its marginals; this replaces the discrepancy step for Inner Product in [Ko26].

Fix $i \in \mathcal{I}$ with $\rho_1 > 0$. Following [Ko26], we take

$$Z \;:=\; \big(Z_{\mathrm{alice}}, B_Q\big), \qquad Z_{\mathrm{alice}} \;:=\; \big(\Phi, A_{\mathrm{all}}, Q, S_{<Q}\big),$$

where $S_{<Q} := (S_1, \ldots, S_{Q-1})$, and $A_{\mathrm{all}} := (A_1, \ldots, A_m)$ with $A_q$ the transcript that Alice sends when the query is $q$. The vector $A_{\mathrm{all}}$ is never transmitted; it is an analysis variable that permits direct sum over the query sets. The prefix $S_{<Q}$ supplies the conditioning required by the chain rule. As in [Ko26], $Z$ includes these transcripts but not Merlin's message $U$. By Theorem 3.3(e), $A_{\mathrm{all}}$ is a function of $(\vec{S}, \Phi)$, and $B_Q$ is a function of $(Z_{\mathrm{alice}}, U)$, since $\Psi$ is part of $\Phi$ and $A_Q$ is determined by

$A_{\text{all}}$ and $Q$. We only consider values of $Z$ with $B_Q = 1$, and write them as $(z, 1)$, where $z$ is a value of $Z_{\text{alice}}$.

We use two facts throughout. First, Merlin's message is short compared to the number $m = n$ of query sets: by Theorem 3.3(b) and Fact 2.2(a),

$$H(U) \;\le\; 5\, t_u w\, s_i \;\le\; n^{3/5}, \tag{1}$$

since $s_i \le s_\ell \le \sqrt{n}$ (Claim 3.2(c)) and $t_u w \le n^{1/95}$. Second, the following standard direct-sum identity holds.

**Claim 3.6** (Direct sum)**.** *For any random variables $V$ and $W$ that are functions of $(\vec{S}, X, R_{\text{Bob}})$,*

$$I\big(S_Q; W \;\big|\; V, Q, S_{<Q}\big) \;=\; \frac{1}{m}\, I\big(\vec{S}; W \;\big|\; V\big).$$

*Proof.* Since $Q$ is uniform and independent of $(\vec{S}, X, R_{\text{Bob}})$, Fact 2.2(c) and the chain rule (Fact 2.3(a)) give

$$I\big(S_Q; W \;\big|\; V, Q, S_{<Q}\big) \;=\; \frac{1}{m} \sum_{q \in [m]} I\big(S_q; W \;\big|\; V, S_{<q}\big) \;=\; \frac{1}{m}\, I\big(\vec{S}; W \;\big|\; V\big). \qquad \square$$

### 3.3.1 Information about $S_Q$

We first bound the divergence of the conditional query-side distribution from uniform, and then obtain high min-entropy by smoothing. Although $Z_{\text{alice}}$ contains all $m$ Alice transcripts, they carry only $O(t_q w)$ bits about a random $S_Q$ on average. As in [Ko26], this yields a high-min-entropy distribution after restricting $S_Q^{(i)}$ to a likely event $E$.

**Proposition 3.7** (The query side)**.** *If $\rho_1 \ge n^{-1/6}$, then*

$$\Pr_{z \sim Z_{\text{alice}} | B_Q = 1} \left[ \exists E \subseteq \binom{B_i}{s_i} : \begin{array}{l} \Pr\big[S_Q^{(i)} \in E \;\big|\; Z = (z,1)\big] \;\ge\; \dfrac{3}{4} \quad \textit{and} \\[2mm] H_\infty\big(S_Q^{(i)} \;\big|\; Z = (z,1),\, S_Q^{(i)} \in E\big) \;\ge\; \log \dbinom{n_i}{s_i} - 32\, n^{1/5} \end{array} \right] \;\ge\; \frac{3}{4}.$$

*Proof.* First we upper bound $I\big(S_Q^{(i)}; Z_{\text{alice}}, B_Q\big)$. Since $S_Q^{(i)}$ is part of $S_Q$, and $S_Q$ is independent of $(Q, S_{<Q})$, the chain rule (Fact 2.3) gives

$$I\big(S_Q^{(i)}; Z_{\text{alice}}, B_Q\big) \;\le\; I\big(S_Q; \Phi, A_{\text{all}}, Q, S_{<Q}, B_Q\big) \;=\; I\big(S_Q; \Phi, A_{\text{all}}, B_Q \;\big|\; Q, S_{<Q}\big).$$

Since $B_Q$ is a function of $(\Phi, A_{\text{all}}, Q, U)$, the chain rule and Claim 3.6 (with $V$ constant) give

$$I\big(S_Q; \Phi, A_{\text{all}}, B_Q \;\big|\; Q, S_{<Q}\big) \;\le\; I\big(S_Q; \Phi, A_{\text{all}}, U \;\big|\; Q, S_{<Q}\big) \;=\; \frac{1}{m}\, I\big(\vec{S}; \Phi, A_{\text{all}}, U\big).$$

By the chain rule and Fact 2.2,

$$I\big(\vec{S}; \Phi, A_{\text{all}}, U\big) \;=\; I\big(\vec{S}; \Phi, U\big) + I\big(\vec{S}; A_{\text{all}} \;\big|\; \Phi, U\big) \;\le\; H(U) + H(A_{\text{all}}),$$

where $I(\vec{S}; \Phi, U) \le I(\vec{S}; X^{(\neq i)}, R_{\text{Bob}}, U) = I(\vec{S}; U | X^{(\neq i)}, R_{\text{Bob}}) \le H(U)$ by Fact 2.2(e) and the chain rule, since $\Psi$ is a function of $(U, R_{\text{Bob}})$ (Theorem 3.3(e)) and $\vec{S}$ is independent of $(X^{(\neq i)}, R_{\text{Bob}})$. By

(1) and $m = n$, $H(U)/m \le n^{-2/5}$, and by Theorem 3.3(c) and Fact 2.2(a), $H(A_{\text{all}})/m \le 17 t_q w$. Since $t_q \le \log^2 n$ and $w = \Theta(\log n)$,

$$I\big(S_Q^{(i)}; Z_{\text{alice}}, B_Q\big) \;\le\; n^{-2/5} + 17 t_q w \;=\; O(\log^3 n).$$

Next we condition on $B_Q = 1$, an event of probability $\rho_1$. Let $\mathcal{U}$ be the uniform distribution on $\binom{B_i}{s_i}$, which is the distribution of $S_Q^{(i)}$. By Fact 2.5, and since divergence is nonnegative (Fact 2.2(d)),

$$\begin{aligned} I\big(S_Q^{(i)}; Z_{\text{alice}}, B_Q\big) \;&=\; \mathop{\mathbb{E}}_{(z,b)\sim(Z_{\text{alice}},B_Q)} \Big[D\big(S_Q^{(i)}|_{Z=(z,b)} \,\big\|\, \mathcal{U}\big)\Big] \\ &\ge\; \rho_1 \cdot \mathop{\mathbb{E}}_{z\sim Z_{\text{alice}}|_{B_Q=1}} \Big[D\big(S_Q^{(i)}|_{Z=(z,1)} \,\big\|\, \mathcal{U}\big)\Big]. \end{aligned}$$

Hence, since $\rho_1 \ge n^{-1/6}$,

$$\mathop{\mathbb{E}}_{z\sim Z_{\text{alice}}|_{B_Q=1}} \Big[D\big(S_Q^{(i)}|_{Z=(z,1)} \,\big\|\, \mathcal{U}\big)\Big] \;\le\; \frac{O(\log^3 n)}{\rho_1} \;\le\; n^{1/5}.$$

Finally, by Markov's inequality, $D\big(S_Q^{(i)}|_{Z=(z,1)} \,\|\, \mathcal{U}\big) \le 4n^{1/5}$ with probability at least $3/4$ over $z \sim Z_{\text{alice}}|_{B_Q=1}$. For such $z$, Fact 2.11, applied to the distribution of $S_Q^{(i)}$ given $Z = (z,1)$ with $\mathcal{S} = \binom{B_i}{s_i}$, $t = 4n^{1/5}$ and $\alpha = 4$, gives the desired set $E$. □

### 3.3.2 Information about $X$

Next, we show that conditioning on $Z = (z,1)$ costs little min-entropy of $X^{(i)}$. The only information about $X^{(i)}$ in $Z_{\text{alice}}$ is Bob's message $\Psi$, which is short, and conditioning on $B_Q = 1$ costs $\log(1/\rho_1)$ more bits.

**Proposition 3.8** (The update side)**.**

$$\Pr_{z\sim Z_{\text{alice}}|_{B_Q=1}} \left[ H_\infty\big(X^{(i)} \;\big|\; Z = (z,1)\big) \;\ge\; \log\binom{n_i}{s_i} - \frac{\varepsilon s_i}{4} - \log\frac{1}{\rho_1} - 2 \right] \;\ge\; \frac{3}{4}.$$

*Proof.* First we lower bound $\widetilde{H}_\infty\big(X^{(i)} \;\big|\; Z_{\text{alice}}\big)$. The set $X^{(i)}$ is uniform on $\binom{B_i}{s_i}$ and independent of $(\vec{S}, X^{(\neq i)}, Q)$, and $Z_{\text{alice}}$ is a function of $(\Psi, \vec{S}, X^{(\neq i)}, Q)$ by Theorem 3.3(e). By Fact 2.7(c), and then (a) with $|\Psi| \le \varepsilon s_i/4$ (Theorem 3.3(b)),

$$\widetilde{H}_\infty\big(X^{(i)} \;\big|\; Z_{\text{alice}}\big) \;\ge\; \widetilde{H}_\infty\big(X^{(i)} \;\big|\; \Psi, \vec{S}, X^{(\neq i)}, Q\big) \;\ge\; \log\binom{n_i}{s_i} - \frac{\varepsilon s_i}{4}.$$

Next we condition on $B_Q = 1$, an event of probability $\rho_1$. By Fact 2.8,

$$\widetilde{H}_\infty\big(X^{(i)} \;\big|\; Z_{\text{alice}}, B_Q{=}1\big) \;\ge\; \widetilde{H}_\infty\big(X^{(i)} \;\big|\; Z_{\text{alice}}\big) - \log\frac{1}{\rho_1} \;\ge\; \log\binom{n_i}{s_i} - \frac{\varepsilon s_i}{4} - \log\frac{1}{\rho_1},$$

where the left side is the average min-entropy with respect to the distribution conditioned on $B_Q = 1$:

$$\widetilde{H}_\infty\big(X^{(i)} \;\big|\; Z_{\text{alice}}, B_Q{=}1\big) \;=\; -\log \mathop{\mathbb{E}}_{z\sim Z_{\text{alice}}|_{B_Q=1}} \Big[2^{-H_\infty(X^{(i)} \,|\, Z=(z,1))}\Big].$$

Finally, by Fact 2.7(b) with $r = 2$, applied to this distribution, $H_\infty(X^{(i)}|Z=(z,1))$ is at least the left side minus 2 with probability at least $1 - 2^{-2} = 3/4$ over $z \sim Z_{\text{alice}}|_{B_Q=1}$. □

### 3.3.3 Correlation between $S_Q$ and $X$

Finally, we show that $S_Q^{(i)}$ and $X^{(i)}$ are weakly correlated after conditioning on $Z = (z, 1)$. Merlin's message $U$ is long, but it is shared by all $m$ query sets, so it reveals little about the random query set $S_Q$. Since $B_Q$ is a function of $(Z_{\text{alice}}, U)$, conditioning on it adds no correlation beyond what $U$ already carries. This is again a standard technique from [KW20] and [Ko26].

**Proposition 3.9** (Correlation on disjoint answers)**.**

$$\mathop{\mathbb{E}}_{z\sim Z_{\text{alice}}|_{B_Q=1}} \Big[ I\big(S_Q^{(i)}; X^{(i)} \;\big|\; Z = (z,1)\big) \Big] \;\le\; \frac{n^{-2/5}}{\rho_1}.$$

*Proof.* We first bound the correlation before conditioning on $B_Q$. Since $S_Q^{(i)}$ is part of $S_Q$, the chain rule (Fact 2.3) and Claim 3.6 give

$$I\big(S_Q^{(i)}; X^{(i)}, U \;\big|\; Z_{\text{alice}}\big) \;\le\; I\big(S_Q; X^{(i)}, U \;\big|\; \Phi, A_{\text{all}}, Q, S_{<Q}\big) \;=\; \frac{1}{m}\, I\big(\vec{S}; X^{(i)}, U \;\big|\; \Phi, A_{\text{all}}\big).$$

Since $A_{\text{all}}$ is a function of $(\vec{S}, \Phi)$, Fact 2.4 lets us drop it from the conditioning:

$$I\big(\vec{S}; X^{(i)}, U \;\big|\; \Phi, A_{\text{all}}\big) \;\le\; I\big(\vec{S}; X^{(i)}, U \;\big|\; \Phi\big).$$

Recall that $\Phi = (X^{(\neq i)}, \Psi)$, where $\Psi$ is a function of $(U, R_{\text{Bob}})$ (Theorem 3.3(e)). Since $\vec{S}$ is independent of $X$, and $R_{\text{Bob}}$ is independent of $(\vec{S}, X)$, the chain rule and Fact 2.2(e) and (f) give

$$I\big(\vec{S}; X^{(i)}, U \;\big|\; X^{(\neq i)}, \Psi\big) \;\le\; I\big(\vec{S}; X^{(i)}, U, R_{\text{Bob}} \;\big|\; X^{(\neq i)}\big) \;=\; I\big(\vec{S}; U \;\big|\; X^{(i)}, X^{(\neq i)}\big) \;\le\; H(U).$$

By (1), we conclude that $I(S_Q^{(i)}; X^{(i)}, U | Z_{\text{alice}}) \le H(U)/m \le n^{-2/5}$, since $m = n$. Next we condition on $B_Q$. Since $B_Q$ is a function of $(Z_{\text{alice}}, U)$, the chain rule gives

$$I\big(S_Q^{(i)}; X^{(i)} \;\big|\; Z_{\text{alice}}, B_Q\big) \;\le\; I\big(S_Q^{(i)}; X^{(i)}, U, B_Q \;\big|\; Z_{\text{alice}}\big) \;=\; I\big(S_Q^{(i)}; X^{(i)}, U \;\big|\; Z_{\text{alice}}\big) \;\le\; n^{-2/5}.$$

Finally, by Fact 2.2(b) and (c),

$$\begin{aligned} I\big(S_Q^{(i)}; X^{(i)} \;\big|\; Z_{\text{alice}}, B_Q\big) \;&=\; \mathop{\mathbb{E}}_{(z,b)\sim(Z_{\text{alice}}, B_Q)} \Big[ I\big(S_Q^{(i)}; X^{(i)} \;\big|\; Z = (z,b)\big) \Big] \\ &\ge\; \rho_1 \cdot \mathop{\mathbb{E}}_{z\sim Z_{\text{alice}}|_{B_Q=1}} \Big[ I\big(S_Q^{(i)}; X^{(i)} \;\big|\; Z = (z,1)\big) \Big]. \end{aligned}$$

□

### 3.3.4 The main contradiction

Combining the three bounds gives a monochromatic near-rectangle after conditioning $S_Q^{(i)}$ on an event of constant probability.

**Lemma 3.10** (A monochromatic near-rectangle)**.** *If $\rho_1 \ge n^{-1/6}$, then there are a value $(z, 1)$ of $Z$ and a set $E \subseteq \binom{B_i}{s_i}$ with $\Pr\big[S_Q^{(i)} \in E \;\big|\; Z = (z,1)\big] \ge 3/4$ such that, writing $\mathcal{E}$ for the event that $Z = (z, 1)$ and $S_Q^{(i)} \in E$,*

*(i)* $H_\infty\big(S_Q^{(i)} \;\big|\; \mathcal{E}\big) \;\ge\; \log \binom{n_i}{s_i} - \frac{\varepsilon s_i}{2}$ *and* $H_\infty\big(X^{(i)} \;\big|\; \mathcal{E}\big) \;\ge\; \log \binom{n_i}{s_i} - \frac{\varepsilon s_i}{2}$, *and*

*(ii)* $I\big(S_Q^{(i)}; X^{(i)} \mid \mathcal{E}\big) \;\le\; n^{-1/5}$.

*In particular, the distribution of* $(S_Q^{(i)}, X^{(i)})$ *given* $\mathcal{E}$ *is a monochromatic* $(\varepsilon s_i/2,\, n^{-1/5})$*-near-rectangle on* $\binom{B_i}{s_i} \times \binom{B_i}{s_i}$.

*Proof.* Draw $z \sim Z_{\text{alice}}|_{B_Q=1}$. Each of the following fails with probability at most $1/4$:

(a) there is a set $E$ as in Proposition 3.7;

(b) $H_\infty(X^{(i)}|Z=(z,1)) \ge \log\binom{n_i}{s_i} - \varepsilon s_i/4 - \log(1/\rho_1) - 2$, by Proposition 3.8;

(c) $I(S_Q^{(i)}; X^{(i)}|Z=(z,1)) \le 4\,n^{-2/5}/\rho_1 \le 4\,n^{-7/30}$, by Proposition 3.9 and Markov's inequality.

So some $z$ satisfies all three. Fix such $z$ and $E$, and let $\mathcal{E}$ be the event that $Z=(z,1)$ and $S_Q^{(i)} \in E$. Given $Z=(z,1)$, the event $S_Q^{(i)} \in E$ is a function of $S_Q^{(i)}$ and has probability at least $3/4$, so Fact 2.9 applies to it.

- *Min-entropy of* $S_Q^{(i)}$. By (a), $H_\infty(S_Q^{(i)}|\mathcal{E}) \ge \log\binom{n_i}{s_i} - 32\,n^{1/5} \ge \log\binom{n_i}{s_i} - \varepsilon s_i/2$, since $s_i \ge n^{1/3}$.
- *Min-entropy of* $X^{(i)}$. By (b) and Fact 2.9, $H_\infty(X^{(i)}|\mathcal{E}) \ge \log\binom{n_i}{s_i} - \varepsilon s_i/4 - \log(1/\rho_1) - 3 \ge \log\binom{n_i}{s_i} - \varepsilon s_i/2$, since $\log(1/\rho_1) \le (\log n)/6$.
- *Correlation.* By (c) and Fact 2.9, $I(S_Q^{(i)}; X^{(i)}|\mathcal{E}) \le \frac{4}{3}\cdot 4\,n^{-7/30} \le n^{-1/5}$.

Finally, by Theorem 3.3(a), $S_Q \cap X = \emptyset$ whenever $B_Q = 1$, so the distribution of $(S_Q^{(i)}, X^{(i)})$ given $\mathcal{E}$ is supported on disjoint pairs. □

**Passing to a product distribution.** As in [Ko26], we replace this distribution by the product of its two marginals and apply Lemma 2.15.

**Corollary 3.11** (A nearly monochromatic rectangle)**.** *If* $\rho_1 \ge n^{-1/6}$*, let* $\mathcal{E}$ *be the event of Lemma 3.10. Then*

$$R \;:=\; S_Q^{(i)}|_{\mathcal{E}} \otimes X^{(i)}|_{\mathcal{E}}$$

*is an* $(\varepsilon s_i/2)$*-rectangle on* $\binom{B_i}{s_i} \times \binom{B_i}{s_i}$*, and*

$$\Pr_{(S,X)\sim R}\big[S \cap X \neq \emptyset\big] \;\le\; n^{-1/10}.$$

*Proof.* By Lemma 3.10, the distribution of $(S_Q^{(i)}, X^{(i)})$ given $\mathcal{E}$ is a monochromatic $(\varepsilon s_i/2,\, n^{-1/5})$-near-rectangle, and its marginals are $S_Q^{(i)}|_{\mathcal{E}}$ and $X^{(i)}|_{\mathcal{E}}$. By Lemma 2.15 with $\delta = 0$, $R$ is an $(\varepsilon s_i/2)$-rectangle with

$$\Pr_{(S,X)\sim R}\big[S \cap X \neq \emptyset\big] \;\le\; \sqrt{\frac{\ln 2}{2}\,n^{-1/5}} \;\le\; n^{-1/10}.$$ □

**Applying the corruption bound.** A corruption bound shows that no such rectangle exists: two independent sparse sets with high min-entropy intersect with noticeable probability. For sets of size $\sqrt{N}$ in a universe of size $N$, this is the rectangle bound of Babai, Frankl and Simon [BFS86]. We need it for sparser sets, and prove it in Section 4.

**Corollary 3.12** (High min-entropy rectangles are not monochromatic)**.** *Let $\varepsilon := 1/40$ and $c := 1/100$. Let $s \geq 200$, let $N$ be a multiple of $s^2$, let $\mathcal{S} := \binom{[N]}{s}$, and let $\nu_A, \nu_B$ be distributions on $\mathcal{S}$ with*

$$H_\infty(\nu_A),\, H_\infty(\nu_B) \;\geq\; \log|\mathcal{S}| - (\varepsilon s - 1).$$

*Then* $\Pr_{a\sim\nu_A,\, b\sim\nu_B}[a \cap b \neq \emptyset] \geq cs^2/N$.

The following lemma completes the proof of the communication lower bound.

**Lemma 3.13** (Communication lower bound)**.** *For every $i \in \mathcal{I}$, the protocol $G_i$ of Theorem 3.3 has $\rho_1 < n^{-1/6}$.*

*Proof.* Suppose, for contradiction, that $\rho_1 \geq n^{-1/6}$. By Corollary 3.11, $R$ is an $(\varepsilon s_i/2)$-rectangle on $\binom{B_i}{s_i} \times \binom{B_i}{s_i}$ that puts mass at most $n^{-1/10}$ on intersecting pairs. On the other hand, apply Corollary 3.12 with $N := n_i$ and $s := s_i$, identifying $B_i$ with $[n_i]$; this is allowed since $n_i = \ell s_i^2$, $s_i \geq n^{1/3} \geq 200$ and $\varepsilon s_i/2 \leq \varepsilon s_i - 1$. So $R$ puts mass at least $cs_i^2/n_i = c/\ell \geq c/\log n > n^{-1/10}$ on intersecting pairs, a contradiction. □

**Proof of Theorem 3.1.** Recall that $\ell = \Theta(\log n/\log(t_u w))$ by Claim 3.2(a). Let $c_0 := 1/192$, and suppose, for contradiction, that $t_q \leq c_0\ell^2$. Since $t_u w \leq n^{1/95}$, Claim 3.2(a) gives $\ell \geq 20$, and hence $|\mathcal{I}| \geq (\ell-4)/3 \geq \ell/4$. Hence, using $\sum_{i\in\mathcal{I}} \tau_i \leq \mathbb{E}_{\mathcal{D},Q}[T(Q)] \leq t_q$, some $i \in \mathcal{I}$ satisfies $\tau_i \leq 4c_0\ell$. As in Corollary 3.5, $\log(1/p) \leq 2\log(t_u w)$, and therefore

$$p^{8\tau_i} \;\geq\; 2^{-16\tau_i \log(t_u w)} \;\geq\; 2^{-64c_0\ell\log(t_u w)} \;\geq\; n^{-16c_0} = n^{-1/12},$$

where the last inequality uses $4\ell\log(t_u w) = \ell\log\beta \leq \log n$. By Theorem 3.3(d),

$$\rho_1 \;\geq\; \frac{1}{8}n^{-1/12} - e^{-n^{1/4}/3} \;>\; n^{-1/6},$$

contradicting Lemma 3.13. Thus $t_q > c_0\ell^2$, and Claim 3.2(a) gives the claimed $\Omega((\log n/\log(t_u w))^2)$ bound. For the bound on $\max(t_u, t_q)$, if $t_u \geq \log^2 n$ there is nothing to prove; otherwise $\log(t_u w) = O(\log\log n)$. □

## 4 A corruption bound for sparse sets

We prove a combinatorial bound: two independent sparse sets, each drawn from a sufficiently large family, intersect with noticeable probability. The case $N = s^2$ is the rectangle bound of Babai, Frankl and Simon [BFS86]. We extend their argument to $s$-sets in a universe of size $N$, where $N$ is a multiple of $s^2$.

Write $\mathcal{S} := \binom{[N]}{s}$, let $\mathcal{A}, \mathcal{B} \subseteq \mathcal{S}$, and let $\mathcal{U}_\mathcal{A}$ and $\mathcal{U}_\mathcal{B}$ be the uniform distributions on these families. We write $\mathcal{U}$ for the uniform distribution on $\mathcal{S}$.

**Lemma 4.1** (Corruption for sparse sets)**.** *Let $\varepsilon := 1/40$ and $c := 1/100$. Let $s \geq 200$, and let $N$ be a multiple of $s^2$. If $\mathcal{A}, \mathcal{B} \subseteq \mathcal{S}$ satisfy $|\mathcal{A}|, |\mathcal{B}| \geq 2^{-\varepsilon s}|\mathcal{S}|$, then*

$$\Pr_{\substack{a\sim\mathcal{U}_\mathcal{A}\\ b\sim\mathcal{U}_\mathcal{B}}}[a \cap b \neq \emptyset] \;\geq\; c \cdot \frac{s^2}{N}$$

First we derive the min-entropy form used in Step 4 (Corollary 3.12) by applying the flat decomposition in Corollary 2.14. We restate the corollary for convenience.

*Corollary 3.12.* Let $s \geq 200$, let $N$ be a multiple of $s^2$, and let $\nu_A, \nu_B$ be distributions on $\mathcal{S}$ with

$$H_\infty(\nu_A),\, H_\infty(\nu_B) \;\geq\; \log|\mathcal{S}| - (\varepsilon s - 1).$$

Then $\Pr_{a\sim\nu_A,\, b\sim\nu_B}[a \cap b \neq \emptyset] \geq cs^2/N$.

*Proof.* The product $\nu_A \otimes \nu_B$ is an $(\varepsilon s - 1)$-rectangle. Since $|\mathcal{S}| \geq 2^{\varepsilon s}$, some integer $K$ satisfies $2^{-\varepsilon s}|\mathcal{S}| \leq K \leq 2^{1-\varepsilon s}|\mathcal{S}|$. By Corollary 2.14 with this $K$, the product is a convex combination of uniform distributions on combinatorial rectangles $\mathcal{A} \times \mathcal{B}$ with $|\mathcal{A}| = |\mathcal{B}| = K \geq 2^{-\varepsilon s}|\mathcal{S}|$. By Lemma 4.1, each of them puts mass at least $cs^2/N$ on intersecting pairs. Linearity gives the same bound for $\nu_A \otimes \nu_B$. □

The one-sided formulation is essential: an analogous lower bound on the probability of disjointness is false. If $\log(N/s) \leq \varepsilon s$, as in our application, then the family $\mathcal{A}$ of sets that contain a fixed element has $|\mathcal{A}| = (s/N)|\mathcal{S}| \geq 2^{-\varepsilon s}|\mathcal{S}|$, but all pairs in $\mathcal{A} \times \mathcal{A}$ intersect. This is why the main proof uses only runs whose answer is "disjoint".

## 4.1 Counting facts

We use the following counting estimates, as in [BFS86]. Write $H_{\mathrm{b}}(\alpha) := -\alpha\log\alpha - (1-\alpha)\log(1-\alpha)$ for the binary entropy function.

**Fact 4.2** (Binomial estimates)**.** *For all integers $u \geq 0$ and $k \geq 1$, $\binom{u}{k} \leq \left(\frac{eu}{k}\right)^k$. For every integer $g \geq 1$ and every $0 < \alpha \leq 1/2$,*

$$\sum_{0\leq h\leq \alpha g} \binom{g}{h} \;\leq\; 2^{H_{\mathrm{b}}(\alpha) g}.$$

For $a \sim \mathcal{U}$, the probability that $a$ contains a fixed set or avoids a fixed set is expressed by binomial coefficients. For $T, V \subseteq [N]$ with $|T| = k \leq s$,

$$\Pr_{a\sim\mathcal{U}}[T \subseteq a] \;=\; \binom{N-k}{s-k}\Big/\binom{N}{s} \;=\; \prod_{h=0}^{k-1}\frac{s-h}{N-h} \;\leq\; \left(\frac{s}{N}\right)^k,$$

$$\Pr_{a\sim\mathcal{U}}[a \cap V = \emptyset] \;=\; \binom{N-|V|}{s}\Big/\binom{N}{s} \;=\; \prod_{h=0}^{s-1}\frac{N-|V|-h}{N-h} \;\leq\; \left(1-\frac{|V|}{N}\right)^s,$$

and $\Pr[T \subseteq a] = 0$ if $|T| > s$. We use the following two consequences.

**Claim 4.3** (Two counting facts)**.** *Let $1 \leq s \leq N$.*

*(a) For every $P \subseteq [N]$ and every integer $k \geq 1$, $\Pr_{a\sim\mathcal{U}}\left[\,|a \cap P| \geq k\right] \leq \binom{|P|}{k}\left(\frac{s}{N}\right)^k \leq \left(\frac{e|P|s}{kN}\right)^k$.*

*(b) For every $V \subseteq [N]$, $\Pr_{a\sim\mathcal{U}}[a \cap V = \emptyset] \leq \left(1-\frac{|V|}{N}\right)^s \leq e^{-s|V|/N}$.*

*Proof.* (a) If $|a \cap P| \geq k$, then $a$ contains some $T \subseteq P$ with $|T| = k$. By a union bound over the $\binom{|P|}{k}$ choices of $T$, the bound $\Pr[T \subseteq a] \leq (s/N)^k$, and Fact 4.2,

$$\Pr_{a\sim\mathcal{U}}\left[\,|a \cap P| \geq k\right] \;\leq\; \Pr_{a\sim\mathcal{U}}\left[\exists\, T \subseteq P \text{ with } |T| = k \text{ and } T \subseteq a\right] \;\leq\; \sum_{\substack{T\subseteq P\\ |T|=k}} \Pr_{a\sim\mathcal{U}}[T \subseteq a]$$

$$\leq\; \binom{|P|}{k}\left(\frac{s}{N}\right)^k \;\leq\; \left(\frac{e\,|P|\,s}{kN}\right)^k.$$

(b) If $N - |V| < s$, the avoidance probability is zero and the bound follows. Otherwise, all factors below are nonnegative and satisfy $\frac{N-|V|-h}{N-h} \le 1 - \frac{|V|}{N}$. Using $1 - x \le e^{-x}$,

$$\Pr_{a\sim\mathcal{U}}[a \cap V = \emptyset] \;=\; \binom{N-|V|}{s} \Big/ \binom{N}{s} \;=\; \prod_{h=0}^{s-1} \frac{N-|V|-h}{N-h}$$
$$\le \left(1 - \frac{|V|}{N}\right)^s \;\le\; e^{-s|V|/N}.$$

□

## 4.2 Proof of the corruption bound

For $N = s^2$, Lemma 4.1 is the rectangle bound behind the $\Omega(\sqrt{N})$ lower bound for Disjointness of [BFS86] (see also [CP10, Section 3.2, proof of Theorem 6]). We first recall their argument before explaining the changes needed for sparser sets. Suppose that $\mathcal{A} \times \mathcal{B}$ is large and nearly disjoint.

- *A disjoint chain in $\mathcal{A}$:* Since $\mathcal{A}$ is large, it contains rows $a_1, \ldots, a_t$ with $t = \Theta(s)$, each of which has at least $\lceil s/2 \rceil$ elements outside the earlier rows. These pieces of new elements are pairwise disjoint and together cover a constant fraction of $[N]$.
- *Columns of $\mathcal{B}$ avoid the chain:* A typical row meets few columns of $\mathcal{B}$, so a typical column of $\mathcal{B}$ meets few of the pieces. That is, it lies in the complement of all but a few of the pieces.
- *Few sets avoid the chain:* A uniform $s$-set meets each piece with constant probability. A union bound over which pieces it meets shows that only a $2^{-\Omega(s)}$ fraction of all $s$-sets meet so few pieces. Hence $\mathcal{B}$ is small, a contradiction.

For sparser sets, i.e. when $N$ is a multiple of $s^2$, we make two changes.

- *A longer chain:* We take $t = (N/s^2)\lfloor s/6 \rfloor = \Theta(N/s)$ rows. The $t$ rows cover at most $N/6$ elements, so another row with many new elements can still be found, while the disjoint pieces cover a constant fraction of the universe.
- *Grouping the pieces:* We group the $t$ pieces into $g = \lfloor s/6 \rfloor$ groups of $N/s^2$ pieces each, and apply the union bound to groups instead of pieces. Each group has $\Theta(N/s)$ elements, so a uniform $s$-set meets it with constant probability, just as it meets a single piece when $N = s^2$. For $N = s^2$, every group is a single piece.

**Proof of Lemma 4.1.** For brevity set $\delta := cs^2/N$. Recall that $|\mathcal{A}|, |\mathcal{B}| \ge 2^{-\varepsilon s} |\mathcal{S}|$. Suppose, for contradiction, that

$$\Pr_{a\sim\mathcal{U}_\mathcal{A},\, b\sim\mathcal{U}_\mathcal{B}}[a \cap b \neq \emptyset] \;<\; \delta.$$

We show that $|\mathcal{B}| < 2^{-\varepsilon s} |\mathcal{S}|$.

*Good rows.* Let $\mathcal{A}' := \{a \in \mathcal{A} : \Pr_{b\sim\mathcal{U}_\mathcal{B}}[a \cap b \neq \emptyset] \le 2\delta\}$. By Markov's inequality, $|\mathcal{A}'| \ge |\mathcal{A}|/2 \ge 2^{-\varepsilon s - 1} |\mathcal{S}|$.

*A chain of rows with new elements.* Put $r := N/s^2$, $k := \lceil s/2 \rceil$, $g := \lfloor s/6 \rfloor$, and $t := rg$. These are positive integers. We select $t$ rows in $\mathcal{A}'$, each of which brings at least $k$ new elements.

**Claim 4.4** (A chain of rows with new elements)**.** *There are $a_1, \ldots, a_t \in \mathcal{A}'$ and sets $D_1, \ldots, D_t$ such that, for every $j \in [t]$,*

$$D_j \;\subseteq\; a_j \setminus (a_1 \cup \cdots \cup a_{j-1}) \qquad \text{and} \qquad |D_j| = k.$$

*In particular, $D_1, \ldots, D_t$ are pairwise disjoint.*

*Proof.* We choose $a_1, a_2, \dots$ greedily. Let $1 \le j \le t$, suppose that $a_1, \dots, a_{j-1}$ have been chosen, and let $P_j := a_1 \cup \cdots \cup a_{j-1}$ (so $P_1 = \emptyset$). It suffices to find $a_j \in \mathcal{A}'$ with $|a_j \cap P_j| < k$. Then $|a_j \setminus P_j| \ge s - k + 1 \ge k$, so we can choose $D_j$ to contain $k$ new elements.

Each earlier row has $s$ elements, so $|P_j| \le (j-1)s < ts \le N/6$. By Claim 4.3(a), using $k \ge s/2$,

$$\Pr_{a \sim \mathcal{U}}[|a \cap P_j| \ge k] \le \left(\frac{e\,|P_j|\,s}{kN}\right)^k \le (e/3)^k \le (e/3)^{s/2} \le 2^{-0.071s}.$$

Since $|\mathcal{A}'| \ge 2^{-\varepsilon s - 1}\,|\mathcal{S}| > 2^{-0.071s}\,|\mathcal{S}|$, some $a_j \in \mathcal{A}'$ has $|a_j \cap P_j| < k$. Finally, $D_j \subseteq a_j \setminus P_j$ is disjoint from $D_1, \dots, D_{j-1} \subseteq P_j$. □

*Grouping the pieces.* Partition $[t]$ into $g$ sets $\Gamma_1, \dots, \Gamma_g$, each of size $r = N/s^2$. For $u \in [g]$, let

$$E_u := \bigcup_{j \in \Gamma_u} D_j.$$

The sets $E_1, \dots, E_g$ are pairwise disjoint, and each has $rk \ge N/(2s)$ elements.

*Most columns miss most of the groups.* Since every $a_j$ lies in $\mathcal{A}'$,

$$\mathop{\mathbb{E}}_{b \sim \mathcal{U}_{\mathcal{B}}}\left[\ |\{j \in [t] : a_j \cap b \ne \emptyset\}|\ \right] = \sum_{j=1}^{t} \Pr_{b \sim \mathcal{U}_{\mathcal{B}}}[a_j \cap b \ne \emptyset] \le 2\delta t,$$

so by Markov's inequality the set $\mathcal{B}'$ of $b \in \mathcal{B}$ that meet at most $4\delta t$ of the rows $a_j$ satisfies $|\mathcal{B}'| \ge |\mathcal{B}|/2$. If $b$ meets $E_u$, then it meets $a_j \supseteq D_j$ for some $j \in \Gamma_u$, and the sets $\Gamma_u$ are disjoint. So every $b \in \mathcal{B}'$ meets at most $4\delta t$ of the sets $E_u$, and by our choice of parameters

$$4\delta t = 4 \cdot \frac{cs^2}{N} \cdot \frac{N}{s^2} g = 4cg = \frac{g}{25}.$$

*Counting $\mathcal{B}'$.* If $b \in \mathcal{B}'$ meets $h \le g/25$ groups, it avoids the union $V$ of the other $g - h$ groups. Since $|V| = (g-h)rk \ge (g-h)N/(2s)$,

$$\frac{s\,|V|}{N} \ge \frac{g-h}{2} \ge \frac{12g}{25}.$$

For each choice of the groups met, Claim 4.3(b) bounds the number of such sets by $e^{-12g/25}\,|\mathcal{S}|$. A union bound and Fact 4.2 give

$$\begin{aligned} |\mathcal{B}'| &\le \sum_{0 \le h \le g/25} \binom{g}{h} e^{-12g/25}\,|\mathcal{S}| \\ &\le 2^{H_{\mathrm{b}}(1/25)g} e^{-12g/25}\,|\mathcal{S}| \le 2^{-0.45g}\,|\mathcal{S}|\,, \end{aligned}$$

because $H_{\mathrm{b}}(1/25) \le 0.2423$ and $12/(25 \ln 2) > 0.69249$. As $g \ge s/6 - 1$ and $s \ge 200$,

$$|\mathcal{B}| \le 2\,|\mathcal{B}'| \le 2^{-0.075s + 1.45}\,|\mathcal{S}| < 2^{-s/40}\,|\mathcal{S}|\,,$$

contradicting the assumed size of $\mathcal{B}$. □

## AI methodology

Claude Opus 5.5 assisted in developing the proof of Lemma 4.1, following the author's proposal to adapt the argument of Babai, Frankl and Simon [BFS86] to sparser sets. The modifications are a longer chain of rows and a union bound over groups of pieces. AI tools also assisted with the adaptations and exposition of the published reductions in Appendix A. The applications other than those originally in [Păt10] were found with the assistance of ChatGPT (GPT-6.0 Astra). For applications based on published reductions, the tool served as an aid to searching the literature. AI tools assisted with application writeups, independent logical checks, and editorial revisions. The author is responsible for the final manuscript.

## References


[AD16] Amir Abboud and Søren Dahlgaard. Popular Conjectures as a Barrier for Dynamic Planar Graph Algorithms. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 477–486, New Brunswick, NJ, USA, October 2016. IEEE. Full version: arXiv:1605.03797.

[AW14] Amir Abboud and Virginia Vassilevska Williams. Popular Conjectures Imply Strong Lower Bounds for Dynamic Problems. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 434–443, October 2014. Full version: arXiv:1402.0054.

[AW26] Josh Alman and Virginia Vassilevska Williams. Truly Subquadratic 3SUM and Truly Subcubic APSP via Triangles in Sparse Lopsided Graphs, October 2026. arXiv:2610.06783 [cs.DS].

[BFS86] László Babai, Péter Frankl, and János Simon. Complexity classes in communication complexity theory (preliminary version). In *27th Annual Symposium on Foundations of Computer Science*, pages 337–347, 1986.

[BL15] Joshua Brody and Kasper Green Larsen. Adapt or Die: Polynomial Lower Bounds for Non-Adaptive Dynamic Data Structures. *Theory of Computing*, 11(19):471–489, 2015.

[Blu85] Norbert Blum. On the Single-Operation Worst-Case Time Complexity on the Disjoint Set Union Problem. In *STACS 85, 2nd Symposium of Theoretical Aspects of Computer Science*, pages 32–38, 1985.

[BT10] Philip Bille and Mikkel Thorup. Regular Expression Matching with Multi-Strings and Intervals. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1297–1308. Society for Industrial and Applied Mathematics, January 2010.

[CDK+11] Edith Cohen, Nick Duffield, Haim Kaplan, Carsten Lund, and Mikkel Thorup. Efficient Stream Sampling for Variance-Optimal Estimation of Subset Sums. *SIAM Journal on Computing*, 40(5):1402–1431, January 2011.

[CDL+14] Timothy M. Chan, Stephane Durocher, Kasper Green Larsen, Jason Morrison, and Bryan T. Wilkinson. Linear-Space Data Structures for Range Mode Query in Arrays. *Theory of Computing Systems*, 55(4):719–741, November 2014.

[CEEP16] Arkadev Chattopadhyay, Jeff Edmonds, Faith Ellen, and Toniann Pitassi. Upper and Lower Bounds on the Power of Advice. *SIAM Journal on Computing*, 45(4):1412–1432, January 2016.

[CGL15] Raphaël Clifford, Allan Grønlund, and Kasper Green Larsen. New Unconditional Hardness Results for Dynamic and Online Problems. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 1089–1107, October 2015.

[CGLS18] Raphaël Clifford, Allan Grønlund, Kasper Green Larsen, and Tatiana Starikovskaya. Upper and Lower Bounds for Dynamic Data Structures on Strings. In *35th Symposium on Theoretical Aspects of Computer Science (STACS 2018)*, volume 96 of *LIPIcs*, pages 22:1–22:14, 2018.

[CKL+24] Li Chen, Rasmus Kyng, Yang P. Liu, Simon Meierhans, and Maximilian Probst Gutenberg. Almost-Linear Time Algorithms for Incremental Graphs: Cycle Detection, SCCs, s-t Shortest Path, and Minimum-Cost Flow. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, pages 1165–1173, Vancouver BC Canada, June 2024. ACM.

[CKM+19] Panagiotis Charalampopoulos, Tomasz Kociumaka, Manal Mohamed, Jakub Radoszewski, Wojciech Rytter, and Tomasz Waleń. Internal Dictionary Matching. In *30th International Symposium on Algorithms and Computation (ISAAC 2019)*, volume 149 of *LIPIcs*, pages 22:1–22:17, 2019.

[CP10] Arkadev Chattopadhyay and Toniann Pitassi. The story of set disjointness. *ACM SIGACT News*, 41(3):59–85, September 2010.

[CT06] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, New York, NY, USA, 2006.

[DGGS22] Bartłomiej Dudek, Paweł Gawrychowski, Garance Gourdel, and Tatiana Starikovskaya. Streaming Regular Expression Membership and Pattern Matching. In Joseph (Seffi) Naor and Niv Buchbinder, editors, *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 670–694. Society for Industrial and Applied Mathematics, Philadelphia, PA, January 2022.

[DI22] Justin Dallant and John Iacono. Conditional Lower Bounds for Dynamic Geometric Measure Problems. In *30th Annual European Symposium on Algorithms (ESA 2022)*, volume 244 of *LIPIcs*, pages 39:1–39:17, 2022. Full version: arXiv:2112.10095.

[DL20] Pavel Dvořák and Bruno Loff. Lower Bounds for Semi-adaptive Data Structures via Corruption. In *40th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2020)*, volume 182 of *LIPIcs*, pages 20:1–20:15, 2020.

[DORS08] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data. *SIAM Journal on Computing*, 38(1):97–139, January 2008.

[FS89] Michael L. Fredman and Michael E. Saks. The Cell Probe Complexity of Dynamic Data Structures. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing*, pages 345–354, 1989.

[GKKP20] Shay Golan, Tomasz Kociumaka, Tsvi Kopelowitz, and Ely Porat. The Streaming $k$-Mismatch Problem: Tradeoffs Between Space and Total Time. In *31st Annual Symposium on Combinatorial Pattern Matching (CPM 2020)*, volume 161 of *LIPIcs*, pages 15:1–15:15, 2020.

[GU18] Paweł Gawrychowski and Przemysław Uznański. Towards Unified Approximate Pattern Matching for Hamming and $L_1$ Distance. In *45th International Colloquium on Automata, Languages, and Programming (ICALP 2018)*, volume 107 of *LIPIcs*, pages 62:1–62:13, 2018.

[HF98] Monika R. Henzinger and Michael L. Fredman. Lower Bounds for Fully Dynamic Connectivity Problems in Graphs. *Algorithmica*, 22(3):351–362, November 1998.

[HKM+12] Bernhard Haeupler, Telikepalli Kavitha, Rogers Mathew, Siddhartha Sen, and Robert E. Tarjan. Incremental Cycle Detection, Topological Ordering, and Strong Component Maintenance. *ACM Transactions on Algorithms*, 8(1):1–33, January 2012.

[HKNS15] Monika Henzinger, Sebastian Krinninger, Danupon Nanongkai, and Thatchaphol Saranurak. Unifying and Strengthening Hardness for Dynamic Problems via the Online Matrix-Vector Multiplication Conjecture. In *Proceedings of the Forty-seventh Annual ACM Symposium on Theory of Computing*, STOC '15, pages 21–30, New York, NY, USA, 2015. ACM. Full version: arXiv:1511.06773.

[IPZ01] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which Problems Have Strongly Exponential Complexity? *Journal of Computer and System Sciences*, 63(4):512–530, December 2001.

[JRS09] Rahul Jain, Jaikumar Radhakrishnan, and Pranab Sen. A property of quantum relative entropy with an application to privacy in quantum communication. *Journal of the ACM*, 56(6):1–32, September 2009.

[JX22] Ce Jin and Yinzhan Xu. Tight dynamic problem lower bounds from generalized BMM and OMv. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1515–1528, Rome Italy, June 2022. ACM. Full version: arXiv:2202.11250.

[KLMS26] Michal Koucký, Bruno Loff, Tulasimohan Molli, and Michael E. Saks. The Natural Proofs Barrier against Data-Structure Lower-Bounds. In *Proceedings of the 58th Annual ACM Symposium on Theory of Computing*, pages 1338–1343, Salt Lake City UT USA, June 2026. ACM.

[Ko26] Young Kun Ko. An $\Omega((\log n/\log\log n)^2)$ Cell-Probe Lower Bound for Dynamic Boolean Data Structures. In *Proceedings of the 67th IEEE Symposium on Foundations of Computer Science (FOCS) (to appear)*. IEEE, 2026. Full version: arXiv:2603.25914v2.

[KPP15] Tsvi Kopelowitz, Seth Pettie, and Ely Porat. Dynamic set intersection. In *Workshop on Algorithms and Data Structures*, pages 470–481. Springer, 2015.

[KW20] Young Kun Ko and Omri Weinstein. An Adaptive Step Toward the Multiphase Conjecture. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 752–761, Durham, NC, USA, November 2020. IEEE.

[Lar12] Kasper Green Larsen. The Cell Probe Complexity of Dynamic Range Counting. In *Proceedings of the Forty-fourth Annual ACM Symposium on Theory of Computing*, STOC '12, pages 85–94, New York, NY, USA, 2012. ACM.

[LWY20] Kasper Green Larsen, Omri Weinstein, and Huacheng Yu. Crossing the Logarithmic Barrier for Dynamic Boolean Data Structure Lower Bounds. *SIAM Journal on Computing*, 49(5):STOC18–323–STOC18–367, January 2020.

[LY25] Kasper Green Larsen and Huacheng Yu. Super-Logarithmic Lower Bounds for Dynamic Graph Problems. *SIAM Journal on Computing*, pages FOCS23–42–FOCS23–69, February 2025.

[MSVT94] Peter Bro Miltersen, Sairam Subramanian, Jeffrey Scott Vitter, and Roberto Tamassia. Complexity models for incremental computation. *Theoretical Computer Science*, 130(1):203–236, August 1994.

[Păt08] Mihai Pătraşcu. Unifying the Landscape of Cell-Probe Lower Bounds. In *FOCS 2008*, pages 434–443, 2008.

[Păt10] Mihai Pătraşcu. Towards polynomial lower bounds for dynamic problems. In Leonard J. Schulman, editor, *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 603–610. ACM, 2010.

[PD06] Mihai Pătraşcu and Erik D. Demaine. Logarithmic Lower Bounds in the Cell-Probe Model. *SIAM J. Comput.*, 35(4):932–963, April 2006.

[Tar75] Robert Endre Tarjan. Efficiency of a Good But Not Linear Set Union Algorithm. *J. ACM*, 22(2):215–225, 1975.

[Vad12] Salil P. Vadhan. Pseudorandomness. *Foundations and Trends® in Theoretical Computer Science*, 7(1–3):1–336, 2012.

[vdBNS19] Jan van den Brand, Danupon Nanongkai, and Thatchaphol Saranurak. Dynamic Matrix Inverse: Improved Algorithms and Matching Conditional Lower Bounds. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 456–480, Baltimore, MD, USA, November 2019. IEEE. Full version: arXiv:1905.05067.

[Yao81] Andrew Chi-Chih Yao. Should Tables Be Sorted? *J. ACM*, 28(3):615–628, July 1981.

# A Reductions to other problems

For completeness and reader convenience, we collect the reductions yielding the applications of Theorem 3.1. Each construction identifies its published source and any adaptations. Lemma A.1 gives the common simulation, Table 2 indexes the bounds, and Appendix A.7 proves the streaming separation.

**Conventions.** Let $S_1, \dots, S_m \subseteq [n]$ be the preprocessed sets, $X \subseteq [n]$ the evolving set, and $q \in [m]$ the final query. Sets are indexed by $i$ and elements by $j$. We write $M_{ij} := \mathbf{1}_{j \in S_i}$, $x_j := \mathbf{1}_{j \in X}$ and $c_i := |S_i \cap X|$, so that $S_q \cap X = \emptyset$ exactly when $c_q = 0$, and we take $m = n$. The data structures we build keep the characteristic vector of $X$ and ignore repeated insertions, so every inserted element is new. The initial target instance is built during Phase I, which is not charged. Every target instance has size $N = n^{\Theta(1)}$; padding with parts that no query sees, such as isolated vertices, extends it to every larger size with $\log N = \Theta(\log n)$, and we discuss the padding only when it needs care. Fixed thresholds may depend on $N$ but are chosen before updates. Numbered references to [AW14, HKNS15, AD16, vdBNS19, DI22, JX22] are to the full versions listed in the bibliography, unless stated otherwise.

For a target data structure on instances of size $N$, in the model of Section 2.1, with word size $w = \Theta(\log N)$, worst-case update time $t_u$ and expected query time $t_q$, let

$$\mathcal{L}_N := \left( \frac{\log N}{\log(t_u w)} \right)^2 .$$

**Lemma A.1** (Transfer)**.** *Consider a reduction from insert-only multiphase Disjointness with $m = n$ to a dynamic problem on instances of size $N = n^{\Theta(1)}$. The initial target instance depends only on $S_1, \dots, S_m$, and the reduction has the following properties.*

*(i) Each insertion into $X$ is simulated by $O(1)$ target updates, and the query $q$ by $O(1)$ target updates and target queries, in some order. Each source operation also probes $O(1)$ further cells, and the answer to $q$ is determined by the answers to the target queries and the contents of these cells.*

*(ii) The target operations form a valid operation sequence for the target problem, determined by $S_1, \dots, S_m$, the sequence of insertions and $q$, and each of their arguments occupies $O(1)$ words.*

*Then every zero-error target data structure satisfies $t_u + t_q = \Omega(\mathcal{L}_N)$, and even $t_q = \Omega(\mathcal{L}_N)$ if the query $q$ is simulated without target updates. In either case, $\max(t_u, t_q) = \Omega\big((\log N / \log\log N)^2\big)$.*

*Proof.* We build a data structure for insert-only multiphase Disjointness. It stores the memory of the target data structure and $O(1)$ auxiliary tables, such as the characteristic vector of $X$, in disjoint address ranges, which changes the word size by at most a constant factor; so each insertion probes $O(t_u)$ cells. During the query, it keeps every write of the simulated target operations, including writes by target queries, in temporary workspace, from which it answers later reads of the same cells. This reproduces the target execution while leaving the memory unchanged, as Problem 1 requires and Section 2.1 allows. By (ii), each target query runs on an operation sequence that does not depend on the random tape, so it probes at most $t_q$ cells in expectation. Hence the query probes $O(t_u + t_q)$ cells in expectation, and $O(t_q)$ if it uses no target updates. Since $\log N = \Theta(\log n)$, Theorem 3.1 gives the first two bounds. For the last one, if $t_u \geq \log^2 N$ there is nothing to prove; otherwise $\log(t_u w) = O(\log\log N)$. □

Table 2: The bounds of this appendix. Each listed quantity is $\Omega(\mathcal{L}_N)$, so in every case $\max(t_u, t_q) = \Omega\big((\log N/\log\log N)^2\big)$.

| Problem | Bounded quantity | Section |
|---|---|---|
| Directed reachability; undirected shortest paths, exact or within a factor $\alpha < 2$; subgraph connectivity | $t_q$ | A.1 |
| $s$-$t$ reachability; #SSR; undirected $s$-$t$ shortest paths, exact or within a factor $\alpha < 5/3$; strong connectivity; cycle detection; triangle detection | $t_u + t_q$ | A.1 |
| Bipartite perfect matching; maximum matching size | $t_u + t_q$ | A.1.1 |
| Exact densest subgraph, or a fixed density threshold | $t_u + t_q$ | A.1.2 |
| Nearest marked vertex, exact or within a one-sided factor $\alpha < 3$ | $t_q$ | A.1.3 |
| Planar distance, unit or positive weights | $t_q$ | A.2.1 |
| Planar directed girth, unit or positive weights | $t_u + t_q$ | A.2.2 |
| Planar diameter, positive weights | $t_u + t_q$ | A.2.3 |
| Planar bipartite maximum-weight matching | $t_u + t_q$ | A.2.4 |
| Set disjointness; Pagh's problem; range mode | $t_q$ | A.3 |
| Langerman's problem; Erickson's problem | $t_u + t_q$ | A.3 |
| Inverse-entry zero testing over prime fields | $t_q$ | A.4.1 |
| Determinant over prime fields, or whether it equals one | $t_u + t_q$ | A.4.2 |
| Three-dimensional convex-hull vertex counting | $t_u + t_q$ | A.5.1 |
| Three-dimensional maximal-point counting | $t_u + t_q$ | A.5.2 |
| Two-dimensional distinct-color counting | $t_q$ | A.5.3 |
| Internal dictionary matching | $t_q$ | A.6.1 |
| Binary Hamming-distance threshold | $t_q$ | A.6.2 |
| Active-document existence | $t_q$ | A.6.3 |

## A.1 Graph problems

Here $N$ is the number of vertices. Except in subgraph connectivity and in Appendix A.1.3, the graph initially has no edges, and an update inserts an edge.

In *directed reachability*, queries test whether one vertex reaches another; *$s$-$t$ reachability* fixes both vertices, and *#SSR* counts vertices other than a fixed source reachable from it. *Strong-connectivity* queries test a pair, count strongly connected components, or report the largest component's size [AW14, HKM+12]. *Cycle detection* tests for a directed cycle [HKM+12, CKL+24], and *triangle detection* for an undirected triangle. *Undirected shortest-path* queries return a distance $d$, or an approximation in $[d, \alpha d]$, preserving $d = \infty$; the $s$-$t$ version fixes both vertices. In *subgraph connectivity*, updates turn on vertices of a fixed undirected graph, and queries test connectivity in the induced subgraph of vertices that are on.

**The reduction.** The reductions for reachability, shortest paths and subgraph connectivity are those of Pătraşcu [Păt10, Appendix A]. Let $G$ be the graph with vertices $u_1, \ldots, u_m$, $v_1, \ldots, v_n$ and $t$, and with the edges $\{u_i, v_j\}$ for $j \in S_i$, inserted in Phase I, and $\{v_j, t\}$ for $j \in X$: inserting $j$ into $X$ inserts $\{v_j, t\}$. In the directed problems, these edges are directed from $u_i$ to $v_j$ and from $v_j$ to $t$, so $G$ is a DAG of depth 2. For subgraph connectivity, the fixed graph has the edges $\{u_i, v_j\}$ for $j \in S_i$ and $\{v_j, t\}$ for all $j$; initially exactly the $u_i$ and $t$ are on, and inserting $j$ into $X$ turns $v_j$ on. (Pătraşcu instead turns on only the queried vertex $u_q$, in the final phase.)

**Queries without an update.** The out-neighbors of $u_q$ are the $v_j$ with $j \in S_q$, and $v_j$ has an edge to $t$ exactly when $j \in X$, so $t$ is reachable from $u_q$ exactly when $S_q \cap X \neq \emptyset$. Reversing all edges gives the version with the fixed source $t$. In the undirected graph, the distance between $u_q$ and $t$ is 2 in this case, and at least 4 otherwise, since $G$ is bipartite with $u_q$ and $t$ on the same side; so an approximation within any factor $\alpha < 2$ tells the two cases apart. In subgraph connectivity, the neighbors of $u_q$ that are on are the $v_j$ with $j \in S_q \cap X$, and each of them is adjacent to $t$, so $u_q$ and $t$ are connected exactly when $S_q \cap X \neq \emptyset$.

**Queries with one update.**

- *s-t reachability, #SSR and s-t shortest paths.* The directed cases are selector variants of Pătraşcu's reduction, and the undirected case specializes Henzinger, Krinninger, Nanongkai and Saranurak [HKNS15, Lemma 3.2] to a single set. Add a vertex $r$, isolated in Phases I and II, as the fixed source, with the fixed target $t$. The query $q$ inserts the edge $(r, u_q)$, or $\{r, u_q\}$ in the undirected case. In the directed case, $t$ is reachable from $r$ exactly when $S_q \cap X \neq \emptyset$, and the number of vertices other than $r$ that are reachable from $r$ is $|S_q| + 1 + \mathbf{1}_{S_q \cap X \neq \emptyset}$, which determines the answer since we store $|S_q|$. In the undirected case, the distance between $r$ and $t$ is 3 if $S_q \cap X \neq \emptyset$, and at least 5 otherwise, so an approximation within any factor $\alpha < 5/3$ tells the two cases apart.

- *Strong connectivity and cycle detection.* The query $q$ inserts the edge $(t, u_q)$. Since $G$ is acyclic, every cycle uses this edge, so a cycle exists exactly when $t$ is reachable from $u_q$ in $G$, that is, when $S_q \cap X \neq \emptyset$. In that case $u_q$, $t$ and some $v_j$ lie in one strongly connected component; otherwise every strongly connected component is a single vertex. So each of the three strong-connectivity queries tells the two cases apart.

- *Triangle detection.* This specializes [HKNS15, Lemma 3.3] to a single set. The query $q$ inserts the edge $\{u_q, t\}$. Since $G$ is bipartite with $u_q$ and $t$ on the same side, every triangle consists of $u_q$, $t$ and some $v_j$ with $j \in S_q \cap X$.

**Corollary A.2** (Dynamic graph problems)**.** *Consider a zero-error cell-probe data structure on $N$ vertices with word size $w = \Theta(\log N)$, worst-case update time $t_u$ and expected query time $t_q$.*

*(a) For incremental directed reachability, incremental undirected shortest paths, exact or within any factor $\alpha < 2$, and incremental subgraph connectivity, $t_q = \Omega(\mathcal{L}_N)$. For directed reachability, this holds already on depth-2 DAGs whose queries ask whether a fixed sink is reachable, or whether a vertex is reachable from a fixed source. For undirected shortest paths, it holds already for distances from a fixed source.*

*(b) For incremental s-t reachability, #SSR, undirected s-t shortest paths, exact or within any factor $\alpha < 5/3$, strong connectivity in each of the three forms above, cycle detection, and triangle detection, $t_u + t_q = \Omega(\mathcal{L}_N)$.*

*In particular,* $\max(t_u, t_q) = \Omega\big((\log N / \log\log N)^2\big)$ *for all of these problems.*

*Proof.* The graph has at most $2n + 2$ vertices. Each insertion into $X$ is simulated by one update, and the query by one query, preceded by one update in (b). So Lemma A.1 gives the bounds. □

### A.1.1 Bipartite matching

The graph is bipartite and unweighted, an update inserts an edge, and a query asks for the size of a maximum matching, or whether a perfect matching exists. We specialize the reduction from

reachability to bipartite matching of Abboud and Vassilevska Williams [AW14, Lemma 6.2]. The two sides are

$$L = \{s_L\} \cup \{u_{i,L} : i \in [n]\} \cup \{v_{j,L} : j \in [n]\}, \qquad R = \{t_R\} \cup \{u_{i,R} : i \in [n]\} \cup \{v_{j,R} : j \in [n]\}.$$

In Phase I, insert the edges $\{u_{i,L}, u_{i,R}\}$ and $\{v_{j,L}, v_{j,R}\}$, and the edges $\{u_{i,L}, v_{j,R}\}$ for $j \in S_i$. Inserting $j$ into $X$ inserts $\{v_{j,L}, t_R\}$. The query $q$ inserts $\{s_L, u_{q,R}\}$, and asks whether there is a perfect matching.

The edges $\{u_{i,L}, u_{i,R}\}$ and $\{v_{j,L}, v_{j,R}\}$ form a matching of size $2n$, which leaves exactly $s_L$ and $t_R$ unmatched, and its augmenting paths are exactly

$$s_L,\ u_{q,R},\ u_{q,L},\ v_{j,R},\ v_{j,L},\ t_R \qquad (j \in S_q \cap X).$$

So a perfect matching, of size $2n+1$, exists exactly when $S_q \cap X \neq \emptyset$, and otherwise the maximum matching has size $2n$. The graph has $4n+2$ vertices. For perfect matching, take even $N$ and pad with disjoint edges; for maximum matching size, an isolated vertex also permits odd $N$. Each insertion into $X$ is one update, and the query is one update and one query, so Lemma A.1 gives $t_u + t_q = \Omega(\mathcal{L}_N)$.

### A.1.2 Exact densest subgraph

The density of a nonempty vertex set $U$ is $|E(G[U])| \,/\, |U|$. The graph is simple and unweighted, an update inserts an edge, and a query asks for the maximum density $\rho^*(G)$ over all nonempty $U$, or whether $\rho^*(G) \geq \theta$ for a threshold $\theta$ fixed in advance. We specialize the reduction of Henzinger, Krinninger, Nanongkai and Saranurak [HKNS15, Section 3.6, Theorems 3.23 and 3.25], whose vector-matrix-vector formulation we use with the row vector $e_q$, supplied in the final phase.

Let $k := 6n$. For each $i \in [n]$, the graph has a triple $R_i$ of vertices with a designated vertex $r_i \in R_i$, and for each $j \in [n]$, a triple $Y_j$ with a designated vertex $y_j \in Y_j$. For each $(i,j) \in [n]^2$, it has a set $P_{ij}$ of $k$ further vertices with two distinct endpoints $a_{ij}$ and $b_{ij}$. In Phase I, insert the edges $r_i a_{ij}$ and $b_{ij} y_j$ for all $i, j$, and, if $M_{ij} = 1$, a path from $a_{ij}$ to $b_{ij}$ through all vertices of $P_{ij}$. A triple is *active* if its vertices form a triangle. Inserting $j$ into $X$ inserts the three edges of a triangle on $Y_j$. The query $q$ inserts the three edges of a triangle on $R_q$, and asks whether $\rho^*(G) \geq \theta$, where

$$\theta := \frac{k+7}{k+6}.$$

The graph has $kn^2 + 6n = 6n^3 + 6n$ vertices, and isolated vertices pad it without changing $\rho^*(G)$.

**Lemma A.3.** *After the query inserts the triangle on $R_q$, we have $\rho^*(G) \geq \theta$ if and only if $S_q \cap X \neq \emptyset$.*

*Proof.* If $j \in S_q \cap X$, then $R_q \cup P_{qj} \cup Y_j$ has $k+6$ vertices and induces $(k-1) + 2 + 3 + 3 = k+7$ edges, so its density is $\theta$.

Conversely, suppose that $S_q \cap X = \emptyset$, and that some nonempty $U$ has $\Phi(U) := |E(G[U])| - \theta\,|U| \geq 0$. We modify $U$ in the following steps, each of which strictly increases $\Phi(U)$; in particular, $U$ never becomes empty, since $\Phi(\emptyset) = 0$.

- For each $(i,j)$, remove every component of $G[U \cap P_{ij}]$, unless it is all of $P_{ij}$, $M_{ij} = 1$, and $r_i, y_j \in U$. A removed component with $v$ vertices spans at most $v-1$ edges and has at most one edge to $r_i$ or $y_j$, so its removal deletes at most $v$ edges and increases $\Phi(U)$ by at least $(\theta - 1)v > 0$.

- For each triple, if it is not active, remove its vertices other than its designated vertex; they are isolated. If it is active and its designated vertex is not in $U$, remove its remaining vertices, which span at most one edge. If it is active and its designated vertex is in $U$, add its missing vertices: adding one vertex adds two edges, and adding two vertices adds three edges, so $\Phi(U)$ increases since $\theta < 3/2$. These steps do not create new edges to the sets $P_{ij}$, since only designated vertices have such edges.

Now let $\pi$ be the number of sets $P_{ij} \subseteq U$, $\tau$ the number of active triples contained in $U$, and $\delta$ the number of designated vertices of inactive triples in $U$. Then $G[U]$ has $v = k\pi + 3\tau + \delta$ vertices and $e = (k+1)\pi + 3\tau$ edges. The only active triple $R_i$ is $R_q$, and the triples $Y_j$ with $j \in S_q$ are inactive, since $S_q \cap X = \emptyset$; so each $P_{ij} \subseteq U$ has an endpoint adjacent to the designated vertex of an inactive triple in $U$. Every designated vertex is adjacent to at most $n$ of the sets $P_{ij}$, so $\pi \le n\delta$. Moreover $\tau + \delta > 0$, since $U$ is nonempty and every $P_{ij} \subseteq U$ comes with $r_i, y_j \in U$. Using $k = 6n$,

$$(k+7)v - (k+6)e = 3\tau + (k+7)\delta - 6\pi \ge 3\tau + 7\delta > 0,$$

so $e/v < \theta$, which contradicts $\Phi(U) \ge 0$. □

Each insertion into $X$ is three updates, and the query is three updates and one query, so Lemma A.1 gives $t_u + t_q = \Omega(\mathcal{L}_N)$. The numerator and denominator of $\theta$ have $O(\log n)$ bits. The bound concerns exact densities: $\theta$ tends to 1, so it does not extend to constant-factor approximation.

### A.1.3 Nearest marked vertex

The graph is fixed, undirected and unweighted, and every vertex is marked or unmarked. An update marks a vertex, and a query asks for the distance $d(v)$ from a given vertex $v$ to the nearest marked vertex, or for a one-sided approximation $\widehat{d}(v)$ with $d(v) \le \widehat{d}(v) \le \alpha\, d(v)$, where $1 \le \alpha < 3$. We adapt the incidence-graph construction of Henzinger, Krinninger, Nanongkai and Saranurak [HKNS15, Lemma 3.7] to marking only, adding a sentinel and hubs so that the graph is connected and all distances are finite.

The graph has vertices $r_1, \dots, r_n$, $v_1, \dots, v_n$, $a$, $b$ and $z$, and the edges $r_i v_j$ for $j \in S_i$, and $r_i a$, $ab$, $bz$ and $bv_j$ for all $i$ and $j$. It is connected and bipartite, with sides $\{r_1, \dots, r_n, b\}$ and $\{v_1, \dots, v_n, a, z\}$. Initially only $z$ is marked. Inserting $j$ into $X$ marks $v_j$, and the query $q$ asks for $d(r_q)$.

The marked vertices are $z$ and the $v_j$ with $j \in X$. So $r_q$ has a marked neighbor, and $d(r_q) = 1$, exactly when $S_q \cap X \neq \emptyset$. Otherwise all marked vertices lie on the other side of the bipartition and are not adjacent to $r_q$, so their distance from $r_q$ is odd and at least 3, and the path $r_q, a, b, z$ shows that $d(r_q) = 3$. So the test $\widehat{d}(r_q) < 3$ tells the two cases apart. The graph has $2n + 3$ vertices, and unmarked leaves attached to $b$ pad it without changing these distances. The costs are one update per insertion and one query, so Lemma A.1 gives $t_q = \Omega(\mathcal{L}_N)$.

## A.2 Planar graph problems

Here $N$ is the number of vertices, an update inserts an edge, and every graph that arises is planar. The reductions adapt the two-grid construction of Abboud and Dahlgaard [AD16, Definition 1, Corollary 1, and Sections 3 and 6] and their selectors for diameter and girth. We add a permanent dummy connection and choose positive connector lengths, so that all updates are insertions. A local shortcut first encodes one matrix entry in a grid distance; a second grid and the connectors turn these entry tests into an OR over the inserted elements.

**The grid.** For a Boolean $p \times p$ matrix $B$ with $p \geq 2$, the grid $G(B)$ has vertices $u_{rc}$, $v_{rc}$, $w_{rc}$ and $x_{rc}$ for $r, c \in [p]$, top terminals $a_c$, right terminals $b_r$, and the following edges.

| Edge | Length |
|---|---|
| $a_c v_{1c}$, and $u_{r-1,c} v_{rc}$ for $r > 1$ | $2c - 1$ |
| $v_{rc} u_{rc}$ | 1 |
| $u_{rc} w_{rc}$ | 2 |
| $w_{rc} u_{r,c+1}$ for $c < p$, and $w_{rp} b_r$ | $2p - 2$ |
| $v_{rc} x_{rc}$ | 1 |
| $x_{rc} w_{rc}$, present exactly when $B_{rc} = 1$ | 1 |

When $B_{rc} = 0$, the vertex $x_{rc}$ is a leaf, which does not affect distances between terminals; so $G(B)$ has $4p^2 + 2p$ vertices for every $B$.

**Lemma A.4** (Grid distances)**.** *For every Boolean $p \times p$ matrix $B$ with $p \geq 2$ and all $q, j \in [p]$,*

$$d_{G(B)}(a_q, b_j) = 2p(p - q + 1) + 2qj - B_{jq},$$

*which is attained by the path down column $q$ and then right along row $j$, through $x_{jq}$ if $B_{jq} = 1$.*

*Proof. Compressing each cell.* Identify each vertex $u_{rc}$ with the corner $(r, c)$, the top terminal $a_c$ with $(0, c)$, and the right terminal $b_r$ with $(r, p + 1)$. The cell $\{v_{rc}, w_{rc}, x_{rc}\}$ meets the rest of the graph only at the three corners $(r - 1, c)$, $(r, c)$ and $(r, c + 1)$. Within the cell, the distances from $(r - 1, c)$ to $(r, c)$, from $(r, c)$ to $(r, c + 1)$, and from $(r - 1, c)$ to $(r, c + 1)$ are $2c$, $2p$ and $2c + 2p - B_{rc}$: the edge $x_{rc} w_{rc}$ saves one unit only between the first and third corners. Replace the cell by the triangle on its three corners with these edge lengths. Since the cells are disjoint, this preserves all distances between corners. The compressed graph has a vertical, a horizontal and a diagonal edge for every $r, c \in [p]$, and no horizontal edges on row 0.

*A lower-bound potential.* Fix $q, j$, and let $\beta := B_{jq}$ and $z_r := \max\{j - r, 0\}$. Define

$$F(r, c) := 2qj + 2p(c - q) - 2\min\{c, q\} z_r + \eta(r, c), \qquad \eta(r, c) := \begin{cases} 1, & r < j,\ c < q, \\ -1, & r < j,\ c > q, \\ -\beta, & r \geq j,\ c > q, \\ 0, & \text{otherwise.} \end{cases}$$

Let $V_{rc} := F(r, c) - F(r - 1, c)$, $H_{rc} := F(r, c + 1) - F(r, c)$ and $D_{rc} := V_{rc} + H_{rc}$ be the increments along the vertical, horizontal and diagonal edges. Direct subtraction gives $0 \leq V_{rc} \leq 2c$ and $0 \leq H_{rc} \leq 2p$; for the lower bound on $H_{rc}$, use $r \geq 1$, so that $z_r \leq p - 1$ and the smallest possible value is $2p - 2z_r - 1 \geq 1$. The diagonal increments are as follows, with an upper bound in the top-left entry and equality elsewhere:

| | $c < q$ | $c = q$ | $c > q$ |
|---|---|---|---|
| $r < j$ | $\leq 2p + 2c - 2$ | $2p + 2q - 1$ | $2p + 2q$ |
| $r = j$ | $2p + 2c - 1$ | $2p + 2q - \beta$ | $2p + 2q + 1 - \beta$ |
| $r > j$ | $2p$ | $2p - \beta$ | $2p$ |

So $0 \leq D_{rc} \leq 2p + 2c - 1$ for $(r, c) \neq (j, q)$, and $D_{jq} = 2p + 2q - \beta$. Since $B$ is Boolean, every diagonal increment is at most the length $2p + 2c - B_{rc}$ of its edge. Hence the potential changes by at most the length of each compressed edge, in either direction.

*Attaining the bound.* Since $F(0,q)=0$ and $F(j,p+1)=2qj+2p(p-q+1)-\beta$, every path from $a_q$ to $b_j$ has at least the claimed length. The path consisting of $j-1$ vertical edges in column $q$, the diagonal edge of cell $(j,q)$, and $p-q$ horizontal edges in row $j$ has exactly this length, and expanding its edges gives the stated path. □

**The core.** Let $p := n+1$, pad $M$ with a zero row and a zero column, and let $B := M^{\mathsf{T}}$; so the rows of $B$ are indexed by elements and its columns by sets, and row $p$ of $B$ is zero. To the right of $G(B)$, place a horizontally reflected copy of $G(0)$, with terminals $b'_1,\dots,b'_p$ facing $b_1,\dots,b_p$ and top terminals $a'_1,\dots,a'_p$, and let $r_q := a'_{p-q+1}$. By Lemma A.4, applied to $G(0)$, the distance from $b'_j$ to $r_q$ in the copy is $2pq+2(p-q+1)j$. Let

$$H := 4p(p+1), \qquad \lambda_j := H + 2(p+1)(p-j), \qquad D := 2H.$$

For $j \in [p]$, the *connector* of $j$ is an edge $b_jb'_j$ of length $\lambda_j$; the connectors join facing terminals, so they do not cross. The connector of $p$ is present from the start, and the connector of $j \in [n]$ is inserted when $j$ is inserted into $X$. The core has $8p^2+4p$ vertices, and every edge length is at most $\lambda_1 = 6p^2+4p-2 < D$.

A path from $a_q$ to $r_q$ that uses the connector of $j$ once, and shortest paths in the two grids (Fig. 3), has length

$$\big(2p(p-q+1)+2qj-B_{jq}\big)+\lambda_j+\big(2pq+2(p-q+1)j\big) = D-B_{jq}.$$

A path from one grid to the other uses an odd number of connectors, and a path that uses at least three of them has length at least $3H > D$, while the connector of $p$ gives a path of length $D$. Hence

$$d(a_q,r_q) = D - f_q, \qquad \text{where} \qquad f_q := \max_{j\in X\cup\{p\}} B_{jq} = \mathbf{1}_{S_q\cap X\neq\emptyset}. \tag{2}$$

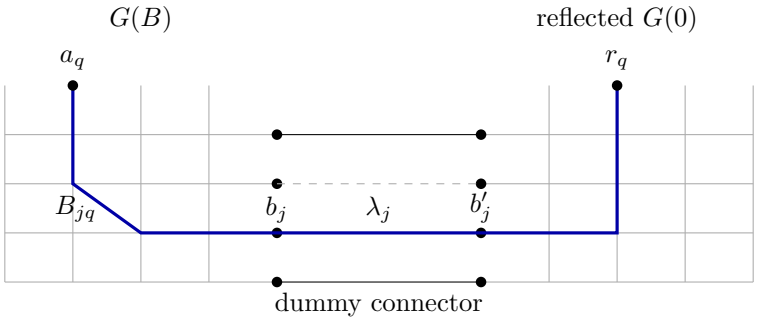


Figure 3: The planar core, with cell interiors suppressed ($p=4$, $q=2$, $X=\{1,3\}$). The highlighted path uses the cell $(j,q)$ and the connector of $j=3$; its length is $D-B_{jq}$. The second grid and connector length cancel the dependence on $j$. Solid connectors are present; the dashed connector is absent. The dummy connector guarantees distance at most $D$.

### A.2.1 Distance

A query asks for the distance between two vertices. The query $q$ asks for $d(a_q,r_q)$, which by (2) is at most $D-1$ exactly when $S_q\cap X\neq\emptyset$; so the bound holds already for this threshold query.

For unit weights, subdivide every edge of the two grids into edges of length 1, and replace the connector of $j$ by a path of length $\lambda_j$ from $b_j$ to $b'_j$. The path of $p$ is present from the start, and the

path of $j \in [n]$ is present from the start except for its edge at $b_j$, which is inserted when $j$ is inserted into $X$. The incomplete paths hang from the right grid, so the graph stays connected, and (2) still holds. Subdividing a grid adds its total edge length $3p^3 + 2p^2 + \sum_{r,c} B_{rc}$, minus its number of edges $5p^2 + \sum_{r,c} B_{rc}$, to its $4p^2 + 2p$ vertices, giving $3p^3 + p^2 + 2p$ vertices; the connector paths add $\sum_j (\lambda_j - 1) = 5p^3 + 4p^2 - 2p$ vertices. So the graph has $11p^3 + 6p^2 + 2p$ vertices, for every $B$.

In both versions, pendant leaves pad the graph, each insertion into $X$ is one update, and the query is one query, so Lemma A.1 gives $t_q = \Omega(\mathcal{L}_N)$.

### A.2.2 Directed girth

A query asks for the length of a shortest directed cycle. As in [AD16, Section 6], direct the horizontal grid edges to the right, the vertical edges of the left grid downwards and those of the right grid upwards, each path $v_{rc}x_{rc}w_{rc}$ from $v_{rc}$ to $w_{rc}$, the remaining leaf edges away from the grids, and each connector from $b_j$ to $b'_j$. Then both grids are acyclic, and all connectors go from the left grid to the right one, so the core is acyclic. The paths that attain the distances of Lemma A.4 in the two grids respect these directions, and directing the edges cannot shorten paths, so the directed distance from $a_q$ to $r_q$ is still $D - f_q$.

The query $q$ inserts an edge of length 1 from $r_q$ to $a_q$, which can be drawn in the outer face, and asks for the girth $g$. Every cycle uses this edge, so $g = D + 1 - f_q$, and $g \le D$ exactly when $S_q \cap X \neq \emptyset$. This holds for positive weights and, after the subdivision of Appendix A.2.1, with each path directed like the edge it replaces, for unit weights. Directed leaves pad the graph. Each insertion into $X$ is one update, and the query is one update and one query, so Lemma A.1 gives $t_u + t_q = \Omega(\mathcal{L}_N)$.

### A.2.3 Diameter

The graph is connected and has positive integer weights, and a query asks for the largest distance between two vertices. We add to the undirected core the selectors of [AD16, Section 6], with permanent fallback edges that keep the graph connected. Let $L := 64p^3$. For each $c \in [p]$, add vertices $z_c$ and $z'_c$ with edges $z_c a_c$ and $z'_c r_c$ of length 1, and add vertices $s$ and $t$ with edges $sz_c$ and $tz'_c$ of length $3L$. These two fans lie in the outer face, next to the two grids, and the edges $sa_c$ and $tr_c$ that a query may insert can be drawn next to the paths $sz_c a_c$ and $tz'_c r_c$; so the graph stays planar. It has $8p^2 + 6p + 2$ vertices. Let $C$ be the graph without $s$ and $t$. Even with all connectors present, the total edge length of $C$ is at most $11p^3 + 9p^2 + p < 16p^3$. To pad, attach leaves to $C$; fewer than $16p + 14$ are needed between consecutive values of $p$, so the total edge length of $C$ stays below $L/2$, and any two vertices of $C$ are at distance less than $L/2$.

The query $q$ inserts the edges $sa_q$ and $tr_q$, of length $L$ each, and asks for the diameter. A path between two vertices of $C$ through $s$ or $t$ has length at least $2L$, so it does not shorten their distance in $C$. A path from $s$ to $t$ that uses an edge $sz_c$ or $tz'_c$ has length at least $4L$, while the two new edges give a path of length at most $2L + D < 3L$; so

$$d(s,t) = 2L + d_C(a_q, r_q) = 2L + D - f_q.$$

Every other distance is less than $2L$: it is less than $L/2$ within $C$, and less than $L + L/2$ between $s$ or $t$ and a vertex of $C$. So the diameter is $2L + D - f_q$, which is at most $2L + D - 1$ exactly when $S_q \cap X \neq \emptyset$. The largest weight is $3L = O(n^3)$; we do not subdivide the selector edges, which would change the diameter. Each insertion into $X$ is one update, and the query is two updates and one query, so Lemma A.1 gives $t_u + t_q = \Omega(\mathcal{L}_N)$.

### A.2.4 Maximum-weight bipartite matching

The graph is bipartite and has positive integer weights, and a query asks for the maximum weight of a matching. We adapt the vertex-splitting and weight-complement construction of Abboud and Dahlgaard [AD16, Section 4] to the directed core of Appendix A.2.2, without the edge from $r_q$ to $a_q$. Let $N_{\text{core}} := 8p^2 + 4p$ be the number of vertices of the core, and let $K := D + 1$. Replace each vertex $v$ of the core by two vertices $v^{\text{in}}$ and $v^{\text{out}}$, joined by an edge of weight $K$, and each directed edge from $u$ to $v$ of length $\ell$ by an edge $u^{\text{out}}v^{\text{in}}$ of weight $K - \ell$, which is positive since $\ell < D$. The graph is bipartite, with the vertices $v^{\text{out}}$ on one side and the vertices $v^{\text{in}}$ on the other. At each vertex of the core, the incoming and the outgoing edges occupy two contiguous sectors: in the left grid, the incoming edges come from the north and west and the outgoing edges leave to the south and east, and analogously in the right grid. So splitting the vertices preserves planarity, as in [AD16]. Add a vertex $\sigma$ on the side of the vertices $v^{\text{out}}$ and a vertex $\tau$ on the other side, isolated in Phases I and II. Inserting $j$ into $X$ inserts the edge that replaces the connector of $j$. The query $q$ inserts the edges $\sigma a_q^{\text{in}}$ and $r_q^{\text{out}}\tau$, of weight $K$ each, which can be drawn in the outer face, and asks for the maximum weight of a matching.

Compare a perfect matching with the matching that consists of the $N_{\text{core}}$ edges $v^{\text{in}}v^{\text{out}}$. Their symmetric difference consists of an alternating path from $\sigma$ to $\tau$, which corresponds to a directed path from $a_q$ to $r_q$ in the core, and of alternating cycles, which would correspond to directed cycles and hence do not exist. Conversely, every directed path from $a_q$ to $r_q$ gives a perfect matching in this way, and if the path has length $d$, then the perfect matching has weight $(N_{\text{core}} + 1)K - d$. Since $d(a_q, r_q) \le D$, some perfect matching has weight at least $(N_{\text{core}} + 1)K - D = N_{\text{core}}K + 1$, while a matching that is not perfect has at most $N_{\text{core}}$ edges, each of weight at most $K$. So the maximum weight of a matching is

$$(N_{\text{core}} + 1)K - (D - f_q) = N_{\text{core}}K + 1 + f_q,$$

which is at least $N_{\text{core}}K + 2$ exactly when $S_q \cap X \neq \emptyset$. The graph has $2N_{\text{core}} + 2$ vertices and weights $O(n^2)$, and isolated vertices pad it. Each insertion into $X$ is one update, and the query is two updates and one query, so Lemma A.1 gives $t_u + t_q = \Omega(\mathcal{L}_N)$.

## A.3 Set and array problems

Pătraşcu [Păt10] reduced the multiphase problem to Langerman's, Pagh's and Erickson's problems, and Chan, Durocher, Larsen, Morrison and Wilkinson [CDL+14] reduced it to range mode. For set disjointness and Pagh's problem, $N$ is the size of the universe, and the instances have $O(N)$ sets; for the other problems, $N$ bounds the number of array or matrix entries, and the constructions use $\Theta(n^2)$ entries.

### A.3.1 Set disjointness

A family of subsets of $[N]$ is maintained under insertions of elements into its sets, and a query asks whether two sets of the family are disjoint [KPP15]. This problem contains multiphase Disjointness directly: the family consists of $S_1, \ldots, S_m$ and $X$, inserting an element into $X$ is an update, and the query $q$ asks whether $S_q$ and $X$ are disjoint.

### A.3.2 Langerman's problem

An array of integers is maintained under updates that assign a value to an entry, and a query asks whether some nonempty prefix has sum zero. The reduction is Pătraşcu's [Păt10, Appendix A].

The array consists of $A[1]$, initially 1, followed by a block for each $j \in [n]$: a head $h_j$, then a pair $(a_{j,i}, b_{j,i})$ for each $i \in [m]$, and then a tail $t_j$. In Phase I, set $(a_{j,i}, b_{j,i}) := (1,1)$ if $j \in S_i$ and $(2,0)$ otherwise, and $(h_j, t_j) := (0, -2m)$. Inserting $j$ into $X$ sets $(h_j, t_j) := (-2m, 0)$. The query $q$ sets $A[1] := 2(m-q)+1$, and asks whether some prefix sum is zero.

Every block sums to zero, so a prefix that ends at $A[1]$ or at a tail has sum $A[1] > 0$. A prefix that ends at the head or at a pair entry of block $j$ has sum $A[1] + h_j + y$, where $y \in [0, 2m]$ is the sum of the pair entries of block $j$ in the prefix; $y$ is even, or $y = 2i-1$ for some $i$ with $j \in S_i$. If $j \notin X$, then $h_j = 0$, and this sum is positive. If $j \in X$, then it equals $y - (2q-1)$, which is zero exactly when $y = 2q-1$, and block $j$ has such a prefix exactly when $j \in S_q$. So some prefix sum is zero exactly when $S_q \cap X \neq \emptyset$.

### A.3.3 Pagh's problem

A family of subsets of a universe is maintained under updates that add the intersection of two sets of the family to the family, and a query asks whether a given element belongs to a given set of the family. The reduction is Pătraşcu's [Păt10, Appendix A], with the universe added to the initial family to handle $X = \emptyset$. The universe is $[m]$, and the initial family consists of $Y_0 := [m]$ and $Y_j := \{i \in [m] : j \notin S_i\}$ for $j \in [n]$. We keep the index of a current set $C$, initially $Y_0$. Inserting $j$ into $X$ adds $C \cap Y_j$ to the family and makes it the current set. Then $C = \{i \in [m] : S_i \cap X = \emptyset\}$, so the query $q$ asks whether $q \in C$.

### A.3.4 Erickson's problem

A matrix of integers is maintained under updates that increment all entries of a row or of a column, and a query asks for the largest entry. The reduction is Pătraşcu's [Păt10, Section 1.2, proof of Theorem 2]. Initially the matrix is $(M_{ij})_{i \in [m], j \in [n]}$. Inserting $j$ into $X$ increments column $j$, and the query $q$ increments row $q$ and asks for the largest entry. The entry in row $i$ and column $j$ is then $M_{ij} + x_j + \mathbf{1}_{i=q}$, so the largest entry is 3 exactly when $S_q \cap X \neq \emptyset$. A quadratic-logarithmic lower bound for a counting version of this problem, whose query asks for the parity of the number of largest entries, is proved in [Ko26, Corollary 1.2].

### A.3.5 Range mode

An array is maintained under updates that assign a value to an entry, and a query asks for a mode, that is, a most frequent value, of a given subarray. We adapt the reduction in [CDL+14, Section 9.2], using a returned mode instead of its frequency. The array consists of $m+1$ chunks of $n$ entries each. In Phase I, chunk $i \in [m]$ lists the elements of $[n] \setminus S_i$ followed by those of $S_i$, and the last chunk consists of zeros; we also store the sizes $|S_i|$ and the entries $M_{ij}$. When $j$ is inserted into $X$ as its $k$-th element, set $A[mn+k] := j$; we also keep $|X|$. If $S_q = \emptyset$, the query $q$ answers "disjoint". Otherwise it asks for a mode $z$ of $A[qn - |S_q| + 1 : mn + |X|]$, and answers "disjoint" exactly when $z \notin S_q \cap X$.

The queried subarray consists of the elements of $S_q$, then $m-q$ chunks that are each a permutation of $[n]$, and then the elements of $X$. So each $j \in [n]$ occurs $(m-q) + M_{qj} + x_j$ times in it, and no other value occurs. If $S_q \cap X \neq \emptyset$, the modes are exactly the elements of $S_q \cap X$, each with $m-q+2$ occurrences. So $z \in S_q \cap X$ exactly when $S_q \cap X \neq \emptyset$.

**Costs.** Each insertion into $X$ is at most two updates. The query is one query for set disjointness, Pagh's problem and range mode, and one update and one query for Langerman's and Erickson's

problems. So Lemma A.1 gives $t_q = \Omega(\mathcal{L}_N)$ for the former three problems, and $t_u + t_q = \Omega(\mathcal{L}_N)$ for the latter two.

## A.4 Matrix problems

Here $N$ is the dimension of a matrix over a prime field, which is maintained under updates that change a single entry. The reductions specialize van den Brand, Nanongkai and Saranurak [vdBNS19, Theorems C.1 and C.3]; our choice of the field prevents cancellation and keeps the matrices nonsingular. We work over $\mathbb{F}_p$ for a prime $p > n + 1$ with $p = \Theta(n)$, so that field elements occupy $\Theta(\log n)$ bits. Let $d := 2n + 1$, and for $x = (x_j)_{j \in [n]}$, let

$$A(x) := \begin{pmatrix} I_n & M & 0 \\ 0 & I_n & x \\ 0 & 0 & 1 \end{pmatrix}, \qquad \text{so that} \qquad A(x)^{-1} = \begin{pmatrix} I_n & -M & Mx \\ 0 & I_n & -x \\ 0 & 0 & 1 \end{pmatrix}.$$

In Phase I, the matrix is $A(0)$, and inserting $j$ into $X$ changes the entry $(n + j, d)$ from 0 to 1. So the matrix is always $A(x)$, which has entries in $\{0, 1\}$ and is unit upper triangular.

**Field size.** The bounds below hold for every family of prime fields with $p = \Theta(N)$: take the largest $n$ with $2n + 1 \le N$ and $n + 2 \le p$, which is $\Theta(N)$, and pad the matrix to dimension $N$ by a direct sum with an identity matrix. This preserves the queried entries of the inverse, the determinant, and nonsingularity.

### A.4.1 Inverse-entry zero testing

The matrix is always nonsingular, and a query asks whether a given entry of its inverse is zero. The query $q$ asks whether the entry $(q, d)$ of $A(x)^{-1}$ is zero. This entry is $c_q \in [0, n]$, which is zero in $\mathbb{F}_p$ exactly when $c_q = 0$, since $p > n$. The costs are one update per insertion and one query, so Lemma A.1 gives $t_q = \Omega(\mathcal{L}_N)$, also for queries that return the entry itself.

### A.4.2 Determinant

A query asks for the determinant, or whether it equals one. As in [vdBNS19, Equation (11)], the query $q$ changes the entry $(d, q)$ from 0 to 1, and asks for the determinant of the resulting matrix $A(x) + e_d e_q^{\mathsf{T}}$. By the matrix determinant lemma $\det(A + uv^{\mathsf{T}}) = \det(A)\,(1 + v^{\mathsf{T}} A^{-1} u)$, which follows by factoring out $A$ and expanding $\det(I + A^{-1} uv^{\mathsf{T}})$,

$$\det\left(A(x) + e_d e_q^{\mathsf{T}}\right) = 1 + \left(A(x)^{-1}\right)_{q,d} = 1 + c_q.$$

Since $1 \le 1 + c_q \le n + 1 < p$, the matrix is nonsingular, and its determinant is one exactly when $c_q = 0$. Each insertion into $X$ is one update, and the query is one update and one query, so Lemma A.1 gives $t_u + t_q = \Omega(\mathcal{L}_N)$. Testing whether the determinant is zero does not tell the two cases apart.

## A.5 Geometric problems

Here $N$ bounds the number of points, which are inserted one at a time, and all coordinates are integers of $O(\log N)$ bits. Each construction has $O(n^2)$ points, so we take $n = \Theta(\sqrt{N})$.

### A.5.1 Convex-hull vertex counting

A query asks for the number of vertices of the convex hull of the current points in $\mathbb{R}^3$. The construction is that of Dallant and Iacono [DI22, ESA version, Section 3, Theorem 10 and Lemma 11]. Their fully dynamic version deletes the blockers of a supplied set $J$; as in their incremental version, we start without blockers and insert the blockers of the elements of $X$, so that $J = [n] \setminus X$. The points lie on $n$ columns, one for each element. A blocker hides its column, and a selector hides all levels above a cutoff, so two consecutive selectors isolate the surviving points of row $q$. Let $n \geq 5$, and let

$$H := 2n + 1, \qquad R := 4n^4, \qquad e_j := (\cos(2\pi j/n), \sin(2\pi j/n)), \qquad v_j(h) := ((R - h^2)e_j, h).$$

In Phase I, insert the points

$$g_{i,j} := v_j(2i + 1) \quad (0 \leq i \leq n,\ j \in [n]), \qquad p_{i,j} := v_j(2i) \quad (i \in [n],\ j \in S_i).$$

The points that may be inserted later are the blockers and the selectors

$$b_j := ((R-1)e_j, H+1) \quad (j \in [n]), \qquad t_a := (0, 0, T_{2a-1}) \quad (a \in [n+1]), \qquad \text{where} \quad T_s := \frac{R + s^2}{2s}.$$

**Lemma A.5.** *Add any set of blockers $b_j$ and any set of selectors $t_a$ to the points inserted in Phase I, and let $a$ be the least index of an added selector, or $n + 2$ if there is none. Then the vertices of the convex hull are exactly*

- *the points $g_{0,j}$ and the added blockers;*
- *the selector $t_a$, if a selector was added;*
- *the points $g_{i,j}$ and the present points $p_{i,j}$ with $1 \leq i < a$ whose blocker $b_j$ was not added.*

*This classification is unchanged if every point is moved by Euclidean distance at most $1/R^2$. Consequently, all the points can be replaced by integer points with coordinates of absolute value $O(n^{12})$.*

*Proof.* Write points as $(x, z)$ with $x \in \mathbb{R}^2$ and $z \in \mathbb{R}$. All radii $R - h^2$ with $1 \leq h \leq H$ are positive, $T_s$ decreases for $s \in [1, H]$, and $T_H > H + 1$.

*Exposed vertices.* The functional $e_j \cdot x - z$ uniquely exposes $g_{0,j}$, and $e_j \cdot x + z/2$ uniquely exposes an added blocker $b_j$. The highest added selector uniquely maximizes $z$. For a point $v_j(h)$ with $h > 1$ below the cutoff whose blocker was not added, use

$$L_{j,h}(x, z) := e_j \cdot x + (2h - \tfrac{1}{2})z.$$

The deficit of another point $v_\ell(h')$ inserted in Phase I, relative to $v_j(h)$, is

$$(h' - h)^2 + \frac{h' - h}{2} + (R - h'^2)(1 - e_j \cdot e_\ell) > 0;$$

here $h$ and $h'$ are integers, and equality is possible only for the same point. A blocker in another column also has smaller value: since

$$1 - \cos(2\pi/n) \geq \frac{8}{n^2} \qquad \text{and} \qquad R - H^2 \geq R/2,$$

its deficit is at least $(R-1)(1-\cos(2\pi/n)) - (H+1)^2 + 1 > 1$. Every added selector has a parameter $s \geq h$, and its deficit is

$$(s - h)(R/s - h) + \tfrac{1}{2}(T_s - h) > 0.$$

These calculations also show that every other point is at Euclidean distance at least $1/(10n)$ from the exposing plane of each claimed vertex. For the highest selector, consecutive possible heights differ by more than one, since

$$T_s - T_{s'} = \frac{(s'-s)(R-ss')}{2ss'} > 1 \qquad (1 \le s < s' \le H,\ s, s' \text{ odd}).$$

*Points hidden by a selector.* We show next that every remaining point is strictly interior. If a selector with parameter $s = 2a-1$ is present, the cone with apex $t_a$ and base $\operatorname{conv}\{g_{a-1,j} : j \in [n]\}$ contains every point $v_j(h)$ inserted in Phase I with $h > s$ in its interior. Indeed, its cross-section at height $h$ is the regular polygon with vertices

$$\big((R + s^2 - 2sh)e_j, h\big) = \big((R - h^2 + (h-s)^2)e_j, h\big),$$

and $v_j(h)$ lies strictly between its center and the corresponding vertex. More quantitatively, if $u$ is the outward unit normal of an edge of the base polygon and $c := \cos(\pi/n)$, the corresponding side face has the equation

$$u \cdot x + 2scz = c(R + s^2).$$

At $v_j(h)$, the slack is at least $c(h-s)^2$, so the distance of $v_j(h)$ from every side face is at least $1/(10n)$, and its distance from the base is at least one. A lower selector is likewise strictly interior to the cone with apex $t_a$ and base $\operatorname{conv}\{g_{0,j} : j \in [n]\}$: its distance from the base is greater than one, and its distance from each side face is greater than $1/2$.

*Points hidden by a blocker.* Suppose that $b_j$ is present. Rotate the coordinates so that $e_j = (1, 0)$, and let

$$c' := \cos(2\pi/n), \qquad d := (R - H^2)\sin(2\pi/n), \qquad D := R - 1 - c'(R - H^2).$$

The tetrahedron with vertices $g_{0,j}$, $b_j$, $g_{n,j-1}$ and $g_{n,j+1}$, with column indices modulo $n$, contains every $v_j(h)$ with $2 \le h \le H$ in its interior. To see this, let $\lambda := (h^2-1)/D$. The barycentric weights of $v_j(h)$ are $\lambda/2$ for each of $g_{n,j-1}$ and $g_{n,j+1}$, and

$$\frac{h - 1 - \lambda(H-1)}{H} \qquad \text{and} \qquad \frac{H + 1 - h - \lambda}{H}$$

for $b_j$ and $g_{0,j}$, respectively. All of them are positive, since $D > H^2 - 1$. Quantitatively, $D \le 2R$ and

$$1 - \frac{H^2-1}{D} = \frac{(1-c')(R-H^2)}{D} \ge \frac{2}{n^2},$$

so every weight is at least $3/(4R)$. The absolute value of the determinant of the tetrahedron is $2HDd \ge 32R^2/n^2$, while the cross product of the edge vectors of every face has norm at most $12R^2$. So every altitude is at least $8/(3n^2)$, and $v_j(h)$ is at distance at least $2/(Rn^2) > 2/R^2$ from every face.

*Rounding.* All the separation and interior margins above exceed $2/R^2$. Moving a point and all vertices of a witnessing hull by at most $1/R^2$ preserves strict separation and strict interior containment, since the support function of the hull changes by at most $1/R^2$ in every unit direction. This proves the perturbation statement. Round every coordinate to the nearest multiple of $1/(2R^2)$, which moves every point by at most $\sqrt{3}/(4R^2) < 1/R^2$, and multiply all coordinates by $2R^2$. The resulting coordinates are integers of absolute value at most $2R^3 + 1 = O(n^{12})$, and scaling preserves the vertex counts. All these coordinates are fixed in Phase I. □

Inserting $j$ into $X$ inserts the blocker $b_j$. The query $q$ inserts $t_{q+1}$ and asks for the number $\kappa$ of hull vertices, and then inserts $t_q$ and asks for the number $\kappa'$. By Lemma A.5,

$$\kappa = n + |X| + 1 + \sum_{i=1}^{q} \big(|J| + |S_i \cap J|\big), \qquad \kappa' = n + |X| + 1 + \sum_{i=1}^{q-1} \big(|J| + |S_i \cap J|\big),$$

so $|S_q \cap X| = |S_q| + |J| - (\kappa - \kappa')$, which we compute from the stored $|S_q|$ and $|X|$. There are at most $2n^2 + 2n + 2$ points. Each insertion into $X$ is one update, and the query is two updates and two queries, which do not depend on the answer to the first query; so Lemma A.1 gives $t_u + t_q = \Omega(\mathcal{L}_N)$.

### A.5.2 Maximal-point counting

A point is *maximal* if no other point is at least as large in every coordinate, and a query asks for the number of maximal points among the current points in $\mathbb{R}^3$. Up to the order of the coordinates and of the columns, the construction is the incremental version of the reduction of Dallant and Iacono [DI22, Theorem 14]; Jin and Xu [JX22, Section 5.1, proof of Theorem 1.10] use the same structure, with $k = 2$ and infinitesimal coordinates, for a stronger conditional bound. In Phase I, insert the points

$$p_{i,j} := (4nj - i,\ n - j,\ i) \qquad (i \in [n],\ j \in S_i).$$

Inserting $j$ into $X$ inserts the blocker $b_j := (4nj,\ n - j,\ n + 1)$, and for $0 \le r \le n$, let $t_r := (4n(n+1),\ n + 1,\ r)$. The query $q$ inserts $t_{q-1}$ and asks for the number $\kappa$ of maximal points, and then inserts $t_q$ and asks for the number $\kappa'$.

The points $p_{i,j}$ are pairwise incomparable: if $p_{i',j'}$ dominates $p_{i,j}$, then the second coordinate forces $j' \le j$, and the first forces $j' \ge j$, since $|i - i'| < 4n$; so $j' = j$, and then the first and third coordinates force $i' = i$. The blocker $b_j$ dominates exactly the points $p_{i,j}$ in column $j$: the second coordinate forces their column to be at least $j$, and the first forces it to be at most $j$. The blockers are pairwise incomparable; their third coordinate prevents the selectors and the points $p_{i,j}$ from dominating them, and their first coordinate prevents them from dominating the selectors. Finally, $t_r$ dominates exactly the points $p_{i,j}$ with $i \le r$, and $t_q$ dominates $t_{q-1}$. Hence

$$\kappa = |X| + 1 + \sum_{i=q}^{n} |S_i \setminus X|, \qquad \kappa' = |X| + 1 + \sum_{i=q+1}^{n} |S_i \setminus X|,$$

so $\kappa - \kappa' = |S_q \setminus X|$, and $S_q \cap X = \emptyset$ exactly when $\kappa - \kappa' = |S_q|$. There are at most $n^2 + n + 2$ points, with coordinates of absolute value $O(n^2)$. Each insertion into $X$ is one update, and the query is two updates and two queries, which do not depend on the answer to the first query; so Lemma A.1 gives $t_u + t_q = \Omega(\mathcal{L}_N)$.

### A.5.3 Distinct-color counting

Colored points in the plane are inserted, and a query asks for the number of distinct colors among the points in a given axis-parallel rectangle. The reduction specializes the construction of Jin and Xu [JX22, Section 3.4, proof of Theorem 1.8] to one family of sets and one inserted set; we use dominance queries and recover the intersection by inclusion–exclusion. Let $W := 4n + 4$. In Phase I, insert the point $(Wi + j,\ W(n + 1 - i))$ with color $j$ for each $i \in [n]$ and $j \in S_i$. Inserting $j$ into $X$ inserts the point $(j, 0)$ with color $j$; we keep $|X|$. The query $q$ asks for the number $C_q$ of colors in $[0, Wq + n] \times [0, W(n + 1 - q)]$.

The bound on the second coordinate excludes the points of rows $i < q$, and the bound on the first coordinate excludes those of rows $i > q$, since $W(q+1)+1 > Wq+n$. So the rectangle contains exactly the points of row $q$ and the inserted points, whose colors form $S_q \cup X$, and $|S_q \cap X| = |S_q| + |X| - C_q$. All coordinates are nonnegative, so the rectangle is a dominance range. There are at most $n^2+n$ points, with coordinates $O(n^2)$; to keep $\Theta(n^2)$ points at all times, add points of a new color to the right of all queried rectangles. The costs are one update per insertion and one query, so Lemma A.1 gives $t_q = \Omega(\mathcal{L}_N)$.

## A.6 String problems

Here $N$ bounds the total length of the texts and patterns. All constructions have size $\Theta(n^2)$, so we take $n = \Theta(\sqrt{N})$.

### A.6.1 Internal dictionary matching

A text $T$ is fixed. Patterns, each given by the endpoints of an occurrence in $T$, are inserted into a dictionary that is initially empty, and a query $\mathsf{Exists}(\ell, r)$ asks whether some pattern of the dictionary occurs in $T[\ell : r]$. The reduction is from [CKM+19, Theorem 24], with a suffix added to $T$ so that every pattern occurs in $T$. Over the alphabet $\{0, 1, \ldots, n\}$, the text consists of the blocks $T[(i-1)n+1 : in]$ for $i \in [n]$, where $T[(i-1)n+j] := j$ if $j \in S_i$ and $T[(i-1)n+j] := 0$ otherwise, followed by the suffix $1\,2 \cdots n$; so $|T| = n^2+n$. Inserting $j$ into $X$ inserts the one-symbol pattern $j$, given by its occurrence at position $n^2+j$. The query $q$ asks $\mathsf{Exists}((q-1)n+1, qn)$. The block of $q$ contains the symbol $j \in [n]$ exactly when $j \in S_q$, the dictionary contains it exactly when $j \in X$, and the symbol 0 is never in the dictionary; so the answer is yes exactly when $S_q \cap X \neq \emptyset$. The costs are one update per insertion and one query, so Lemma A.1 gives $t_q = \Omega(\mathcal{L}_N)$.

### A.6.2 Hamming-distance threshold

A binary text $T$ is fixed, and a binary pattern $P$ is maintained under updates that change one of its bits. A query asks whether the Hamming distance between $P$ and the substring of $T$ of length $|P|$ that starts at a given position is at most $k$, for a threshold $k$ fixed in advance. The reduction combines the construction of Clifford, Grønlund, Larsen and Starikovskaya [CGLS18, Theorem 15], which concatenates the rows of the matrix, with their binary encoding [CGLS18, Lemma 11]. Let $L := n^2$. Before the encoding, let $T_0$ be the concatenation of the rows $M_{i1} \cdots M_{in}$ for $i \in [n]$, followed by $0^L$, and let $P_0 := x_1 \cdots x_n 0^{L-n}$; so $|T_0| = 2L$ and $|P_0| = L$. Encode the symbols of the pattern and of the text by

$$f(0) := 010, \qquad f(1) := 111, \qquad \text{and} \qquad g(0) := 100, \qquad g(1) := 111,$$

respectively, and let $P := f(P_0)$ and $T := g(T_0)$. For all $a, b \in \{0, 1\}$, the Hamming distance between $f(a)$ and $g(b)$ is $2 - 2ab$. Initially $x = 0^n$, and inserting $j$ into $X$ changes the codeword of $x_j$ from 010 to 111, by two updates. The query $q$ compares $P$ with the substring of $T$ that starts at position $3(q-1)n+1$, which fits in $T$ since $3(q-1)n + 3L \leq 6L = |T|$. There, the first $n$ codewords of $P$ are compared with row $q$, and each of the other $L-n$ codewords of $P$ encodes 0 and contributes exactly two mismatches, whatever the aligned text symbol. So the Hamming distance is

$$2L - 2\sum_{j=1}^{n} M_{qj} x_j = 2L - 2c_q,$$

and the threshold $k := 2L - 1 = 2\,|P|\,/3 - 1$ detects whether $c_q > 0$. Each insertion into $X$ is two updates and the query is one query, so Lemma A.1 gives $t_q = \Omega(\mathcal{L}_N)$, also for queries that return the Hamming distance. The threshold grows with the length of the pattern; we do not claim a bound for a constant threshold or for constant-factor approximation.

### A.6.3 Active-document existence

A collection of documents is fixed, and every document is active or inactive. An update activates a document, and a query asks whether some active document contains two given patterns. The reduction specializes the document construction of Jin and Xu [JX22, Section 3.3, proof of Theorem 1.7]: one of the two patterns is a marker that every document contains, and all updates are activations. Over the alphabet $[n] \cup \{\#, \$\}$, the document of $j \in [n]$ is

$$D_j := \# \, d_{1j} \cdots d_{nj}, \qquad \text{where} \qquad d_{ij} := \begin{cases} i, & j \in S_i, \\ \$, & j \notin S_i. \end{cases}$$

All documents are initially inactive, and their total length is $n(n+1)$. Inserting $j$ into $X$ activates $D_j$, and the query $q$ asks whether some active document contains both $q$ and $\#$. Every document contains $\#$, and $D_j$ contains $q$ exactly when $j \in S_q$; so the answer is yes exactly when $S_q \cap X \neq \emptyset$. The same holds for the query with the single pattern $q$. The costs are one update per insertion and one query, so Lemma A.1 gives $t_q = \Omega(\mathcal{L}_N)$.

## A.7 Streaming regular-expression membership: amortized versus worst case

In streaming regular-expression membership, an expression $R$ is preprocessed, and a text then arrives one symbol at a time; after each symbol, the algorithm reports whether the text read so far belongs to $L(R)$, without knowing which symbol is the last one. Bille and Thorup [BT10] and Dudek, Gawrychowski, Gourdel and Starikovskaya [DGGS22] give algorithms for this problem. Here we give a direct encoding of Disjointness with constant amortized cost. We count the cell probes made while processing each symbol, including its report; preprocessing is not charged, the space is polynomial, and the conventions of Section 2.1 apply.

**The expression family.** Fix sets $S_1, \ldots, S_m \subseteq [n]$. The alphabet consists of the symbol $\#$, the data symbols $a_1, \ldots, a_n$ and the selectors $c_1, \ldots, c_m$, and we write $\cup$ for the union (alternation) operator of regular expressions. For each $i \in [m]$ with $S_i \neq [n]$, let

$$\Gamma_i := \bigcup_{j \in [n] \setminus S_i} a_j$$

be the alternation of the data symbols of the elements outside $S_i$, and let the $i$-th branch be $\# \, \Gamma_i^* \, c_i$; if $S_i = [n]$, let the $i$-th branch be $\# \, c_i$. Let $R_{\vec{S}}$ be the union of the $m$ branches, an ordinary regular expression that uses only union, concatenation and Kleene star. For every $q \in [m]$ and every sequence $j_1, \ldots, j_k \in [n]$, in any order and possibly with repetitions,

$$\# \, a_{j_1} \cdots a_{j_k} \, c_q \in L(R_{\vec{S}}) \qquad \Longleftrightarrow \qquad S_q \cap \{j_1, \ldots, j_k\} = \emptyset. \tag{3}$$

Indeed, the final selector $c_q$ forces the $q$-th branch, which accepts exactly when every preceding data symbol is $a_j$ for some $j \notin S_q$.

**Constant amortized time.** In preprocessing, store the incidence table of the sets $S_i$, a bit $\mathsf{seen}[j] = 0$ for each $j \in [n]$, an empty list, and the parser state. After the initial #, when $a_j$ arrives, append $j$ to the list if $\mathsf{seen}[j] = 0$, and set $\mathsf{seen}[j] := 1$. When the first selector $c_q$ arrives, scan the list, report "yes" exactly when none of its indices belongs to $S_q$, and enter a terminal state. All earlier reports are "no". An initial symbol other than #, a second #, or any symbol after a selector causes permanent rejection, at constant cost per later symbol. The scan examines at most as many indices as there are preceding data symbols, so every prefix of length $L$ costs $O(L)$ in total, also for malformed texts. The expression has length $m \le M = O(mn)$. For $m = n$, we have $\log M = \Theta(\log n)$, the alphabet has $2n+1$ symbols, and the $O(n^2)$-bit incidence table and the $O(n)$ auxiliary words use polynomial space.

**Theorem A.6** (Streaming membership: amortized versus worst case)**.** *Let $\mathcal{R}_n := \{R_{\vec{S}} : S_1, \dots, S_n \subseteq [n]\}$, and let $M_n := \max_{R \in \mathcal{R}_n} |R| = \Theta(n^2)$. Streaming membership for this family has a deterministic algorithm with $O(1)$ amortized time per symbol on every input stream.*

*Consider a zero-error cell-probe algorithm for streaming membership for all of $\mathcal{R}_n$, with word size $w = \Theta(\log M_n)$ and arbitrary preprocessing of the given expression, which stores all information retained between symbols in its memory. Let $t_a$ be its worst-case cost on a data symbol, over all expressions, texts and random tapes, and let $t_c$ be an upper bound on its expected cost on a selector, for every expression and every text that ends with this selector. Then*

$$t_c = \Omega\left(\left(\frac{\log M_n}{\log(t_a w)}\right)^2\right).$$

*In particular, the worst-case time on some symbol, for some expression in the family and some text, is $\Omega\big((\log M_n / \log\log M_n)^2\big)$.*

*Proof.* The algorithm above gives the upper bound. For the lower bound, given an instance $S_1, \dots, S_n$ of insert-only multiphase Disjointness, preprocess $R_{\vec{S}}$ and process the symbol # during Phase I. Inserting $j$ into $X$ feeds the symbol $a_j$, and the query $q$ feeds the selector $c_q$; by (3), the report on $c_q$ answers the query. Keeping the writes made while processing $c_q$ in temporary workspace, as in Lemma A.1, we obtain a zero-error data structure with worst-case update time $t_a$ and expected query time $t_c$. Since $\log M_n = \Theta(\log n)$, Theorem 3.1 gives the bound on $t_c$. For the last statement, let $t$ be the worst-case time on all symbols, so that $t_a, t_c \le t$. If $t \ge \log^2 M_n$ there is nothing to prove; otherwise $\log(t_a w) = O(\log\log M_n)$, so $t \ge t_c = \Omega\big((\log M_n / \log\log M_n)^2\big)$. □

**Binary alphabet.** Encode the symbols by distinct codewords of length $b = \Theta(\log n)$. The maximum expression length becomes $M_n^{\mathrm{bin}} = O(n^2 \log n)$, with $\log M_n^{\mathrm{bin}} = \Theta(\log n)$; take $w = \Theta(\log M_n^{\mathrm{bin}})$. Decoding preserves constant amortized time per bit: report "no" within incomplete codewords and reject invalid codewords permanently. A matcher with worst-case cost $t$ per bit yields source update and query costs at most $bt$. If $t > \log M_n^{\mathrm{bin}}$, the bound below is immediate. Otherwise $\log(btw) = O(\log\log M_n^{\mathrm{bin}})$, so Theorem 3.1 gives $bt = \Omega\big((\log M_n^{\mathrm{bin}} / \log\log M_n^{\mathrm{bin}})^2\big)$. Since $b = \Theta(\log M_n^{\mathrm{bin}})$,

$$t = \Omega\left(\frac{\log M_n^{\mathrm{bin}}}{(\log\log M_n^{\mathrm{bin}})^2}\right).$$

**Related separations.** Union–find admits amortized $\Theta(\alpha(n))$ time [Tar75, FS89] but requires $\Theta(\log n / \log\log n)$ worst-case time [Blu85, FS89]. Cohen, Duffield, Kaplan, Lund and Thorup [CDK+11] obtain $O(\log\log k)$ amortized time and an $\Omega(\log k / \log\log k)$ worst-case cell-probe bound for variance-optimal reservoir sampling in their finite-precision model. Streaming $k$-mismatch has conditional bounds from combinatorial Boolean matrix multiplication [GKKP20, GU18].