Electronic Colloquium on Computational Complexity, Report No. 171 (2026)

Quantum Query Advantage Requires Space

Minbo Gao*[1,2], Zhengfeng Ji†[3], and Ziyi Xie‡[3]

[1] *Institute of Software, Chinese Academy of Sciences.*
[2] *University of Chinese Academy of Sciences.*
[3] *Tsinghua University.*

Abstract

Hao, Huang, and Liu [HHL26] recently showed that optimal quantum query complexity may require large workspace even for short-output problems, and asked whether a quantum query advantage over classical computation can itself require space. We resolve this question by exhibiting an explicit total Boolean function with quantum query complexity $Q = \widetilde{\Theta}(M)$, and randomized query complexity $R = \Theta(M^{21/20})$, whereas, for every fixed $0 < \eta < 1/100$ and $S \leq O(M^{1/100-\eta})$, its S-space quantum query complexity satisfies $Q_S = \omega(M^{21/20})$. Consequently,

$$Q < R < Q_S,$$

so the unrestricted quantum query advantage disappears under sufficiently small workspace.

The separation is obtained through a one-bit filtered-parity construction and a space-sensitive quantum lower bound based on compressed-oracle capacity and a new parity-to-capacity inequality, which may be of independent interest.

*gaomb@ios.ac.cn or gmb17@tsinghua.org.cn.
†jizhengfeng@tsinghua.edu.cn.
‡xie-zy21@tsinghua.edu.cn.

ISSN 1433-8092

Contents

1 Introduction

Time–space tradeoffs are a classical topic in complexity theory, which asks how restrictions on workspace affect the time required for computation. A substantial body of work establishes quantitative time–space tradeoffs for fundamental (classical) problems, including long-output problems like sorting [BFK+81, BC82], finding unique elements [Bea91], collision finding [Din20]; and one-bit output problems, like branching-programs [BJS01], randomized decision-problem [BSSV03], and element distinctness [BFMADH+87, Yao94].

The same issue arises in quantum query complexity. The standard query model measures the number of oracle queries while imposing no a priori bound on the algorithm's workspace. Thus, an algorithm that is optimal in query complexity may require substantially more space than is achievable in a space-bounded model. This has motivated the study of quantum time–space tradeoffs for a variety of witness-producing problems, including sorting [Kla03, KŠdW07], function inversion [CGLQ20], finding multiple collision pairs [HM23], and matrix problems [KŠdW07, BKW26]. In the meanwhile, time–space tradeoffs for decision problems have been less explored. In a restricted model, where only quantum branching-programs are allowed, [BS24] shows the one-bit quantum time–space tradeoff, but things are tricky in the general query model.

Collision and element distinctness might be a good candidate for one-bit quantum time–space, or query–space tradeoff. For the collision problem, the Brassard–Høyer–Tapp algorithm achieves $O(N^{1/3})$ quantum queries while storing a table of size $O(N^{1/3})$ [BHT97], whereas a Grover-search-based algorithm can be implemented with logarithmic workspace using $O(\sqrt{N})$ queries [Gro96, Aar21]. For element distinctness, the optimal $O(N^{2/3})$-query quantum walk algorithm also uses polynomial workspace [Amb07], matching the unrestricted quantum query lower bound [AS04]. Despite the tight characterization of their unrestricted quantum query complexities, the optimal tradeoffs for the two remain open [Aar21]. Besides the tradeoff itself, a stronger question is whether the tradeoff can eliminate the quantum query advantage in bounded space.

An immediate barrier to this question is that known quantum time–space lower-bound methods only work for multi-output problem structures, in which the output itself provides a progress measure: after partitioning the computation into segments, one can bound how many output items a small-space segment can produce. Naturally, this argument does not directly extend to Boolean-output problems. We need a different progress measure.

Recent work of Hao, Huang, and Liu makes substantial progress in this direction [HHL26]. They show that even for problems with short outputs, achieving the optimal quantum query complexity can require large workspace, thereby separating space-bounded from unrestricted quantum query complexity in quantum random oracle model. What remains open are whether the tradeoff could be proven for a one-bit output problem, and furthermore, the quantum advantage disappears under an appropriate space bound. We furthermore hope to build this relationship in query complexity instead of oracle models, that is, consider the worst case instead of the average.

This motivates the central question of this work:

Can the size of workspace determine whether a quantum query advantage exists?

Formally, let $Q(f)$ denote the bounded-error quantum query complexity of a total Boolean function f with unrestricted workspace, $R(f)$ its bounded-error randomized query complexity, and $Q_S(f)$ its bounded-error quantum query complexity with workspace S. We still wonder whether $Q(f) < R(f) < Q_S(f)$, which shows that unrestricted quantum algorithms outperform classical ones, while space-bounded quantum algorithms do not.

We answer this question affirmatively. Our starting point is a filtered-parity problem, which has one-bit output, and predicting the output bit forces a quantum algorithm to record information with many queries. We formalize this through a new parity-to-capacity inequality and combine it with the compressed-oracle capacity bound of Hao, Huang, and Liu [HHL26] to obtain a strong query–space tradeoff despite the one-bit output. Recall that Grover's algorithm gives a quadratic speedup for OR. The mechanism above could be turned into a worst-case total Boolean function f satisfying $Q(f) < R(f) < Q_S(f)$, mainly by appending an independent OR block.

1.1 Main Results

We first introduce a family of one-bit oracle problems, called ℓ-Sum-Filtered Parity. The problem is defined over two independent random oracles: one selects a structured set of ℓ-tuples, and the other assigns a Boolean value to each tuple. For an integer k, write $[k] = \{0, 1, \ldots, k-1\}$.

Definition 1.1 (ℓ-Sum-Filtered Parity)**.** *Fix a constant integer $\ell \geq 4$. Let*

$$N = 2^n, \qquad M = \lfloor N^{\frac{1}{\ell-1}} \rfloor,$$

and let

$$H\colon [M] \to [N], \qquad G\colon [M]^\ell \to \{0,1\}$$

be independent uniformly random functions. Define

$$Y_H := \left\{ (x_1, \ldots, x_\ell) \in [M]^\ell : x_1 < \cdots < x_\ell, \quad \sum_{i=1}^{\ell} H(x_i) = 0 \pmod N \right\}.$$

Given oracle access to (H, G), the task is to output $F(H,G) := \bigoplus_{x \in Y_H} G(x)$, where the parity of the empty set is zero.

Our first result gives a general query–space tradeoff for this problem.

Theorem 1.1 (General query–space tradeoff)**.** *Fix a constant integer $\ell \geq 4$. The ℓ-Sum-Filtered Parity problem has the following properties.*

1. *There is a deterministic classical algorithm that succeeds with probability $1 - o(1)$, uses $O(M)$ oracle queries, and uses $O(M \log N)$ bits of space.*

2. *For every constant $0 < \varepsilon \leq 1/2$, every quantum algorithm that succeeds with probability at least $1/2 + \varepsilon$, uses S qubits of space, and makes T oracle queries satisfies*

$$S^{\ell+1} T = \Omega\left(N^{\frac{1}{2(\ell-1)} + \frac{1}{\ell+1}} \right),$$

whenever $S = \Omega(\log N)$ and $S = O(N^{\frac{1}{(\ell+1)(2\ell+1)}})$.

The choice $M = \Theta(N^{\frac{1}{\ell-1}})$ balances the number of H-queries and selected G-entries. Consequently, within the stated space range, if $S = o\left(N^{\frac{\ell-3}{2(\ell-1)(\ell+1)^2}} \right)$, then every such quantum algorithm makes $\omega(N^{\frac{1}{\ell-1}})$ queries, whereas the classical algorithm makes only $O(N^{\frac{1}{\ell-1}})$ queries.

We next turn this separation into one for an explicit total Boolean function.

Definition 1.2 (ℓ-Sum-OR Parity)**.** *Let $N = 2^n$ and $M = \lfloor N^{1/(\ell-1)} \rfloor$. Let $H \in \{0,1\}^{[M]\times[n]}$, $G \in \{0,1\}^{[M]^\ell}$, and $z \in \{0,1\}^L$ be arbitrary input blocks, where $L = \lceil M^{1+(\ell-3)/(4(\ell+1))} \rceil$. For $x \in [M]$, let $H(x) \in [N]$ be the integer encoded by $(H_{x,1}, \dots, H_{x,n})$, and similarly for $G(x)$. Let*

$$Y_H = \left\{ (x_1, \dots, x_\ell) \in [M]^\ell : x_1 < \cdots < x_\ell, \quad \sum_{i=1}^{\ell} H(x_i) = 0 \pmod N \right\}.$$

Given query access to these Boolean blocks, the required output is determined as follows. If

$$|Y_H| \le \left\lceil 2\binom{M}{\ell}/N \right\rceil,$$

then

$$\mathsf{SumOR}_{\ell,n}(H, G, z) = \left(\bigoplus_{\mathbf{x}\in Y_H} G(\mathbf{x}) \right) \oplus \mathrm{OR}(z).$$

Otherwise,

$$\mathsf{SumOR}_{\ell,n}(H, G, z) = \mathrm{OR}(z).$$

Here we set the parity of the empty set to be zero.

Our main result gives the desired separation between unrestricted quantum, randomized classical, and space-bounded quantum query complexity.

Theorem 1.2 (Total quantum speedup and small-space reversal)**.** *Fix a constant integer $\ell \ge 4$. For all sufficiently large n,*

$$D(\mathsf{SumOR}_{\ell,n}) = R(\mathsf{SumOR}_{\ell,n}) = \Theta\left(M^{1+\frac{\ell-3}{4(\ell+1)}} \right), \qquad Q(\mathsf{SumOR}_{\ell,n}) = \widetilde{\Theta}(M).$$

Moreover, every S-qubit quantum algorithm with constant advantage and T queries satisfies

$$S^{\ell+1} T = \widetilde{\Omega}\left(M^{3/2-2/(\ell+1)} \right)$$

for $\Omega(\log M) \le S \le O\left(M^{\frac{\ell-1}{(\ell+1)(2\ell+1)}} \right)$. Consequently, for every fixed $0 < \eta < \frac{\ell-3}{4(\ell+1)^2}$, if $S \le O(M^{\frac{\ell-3}{4(\ell+1)^2}-\eta})$, then

$$Q(\mathsf{SumOR}_{\ell,n}) < R(\mathsf{SumOR}_{\ell,n}) < Q_S(\mathsf{SumOR}_{\ell,n}).$$

In particular, setting $\ell = 4$ gives an explicit family of total Boolean functions f satisfying

$$Q(f) = \widetilde{\Theta}(M), \qquad R(f) = D(f) = \Theta(M^{21/20}), \qquad S^5 Q_S(f) = \widetilde{\Omega}(M^{11/10}).$$

Therefore, for every fixed $0 < \eta < 1/100$ and $S \le O(M^{1/100-\eta})$, we have $Q(f) < R(f) < Q_S(f)$.

1.2 Technique Overview

The classical upper bound is straightforward. The algorithm queries and stores all values of H, enumerates the filtered set Y_H, and queries G only on tuples in Y_H. With high probability, $|Y_H| = \Theta(M)$, which gives the claimed query and space complexities.

For the quantum lower bound, we use the compressed-oracle framework of Zhandry [Zha19] together with the tuple-capacity measure introduced by Hao–Huang–Liu [HHL26]. Informally, the capacity $K(A)$ measures how much of the oracle G has been recorded on tuples satisfying the sum constraint defined by H. Hao–Huang–Liu showed that an S-space, T-query quantum algorithm can have capacity at most

$$K(A) = O\left(S^{2\ell+2}T^2N^{-2/(\ell+1)}\right).$$

The main new step in our proof is to lower-bound the same quantity from the one-bit output. Conditioned on H, the target is the parity of the G-bits indexed by Y_H. We prove a parity-to-capacity inequality showing that any algorithm computing this parity with constant advantage must satisfy $K(A) = \Omega(M)$. Combining the two bounds and using $M = \Theta(N^{\frac{1}{\ell-1}})$ gives

$$S^{\ell+1}T = \Omega\left(N^{\frac{1}{2(\ell-1)}+\frac{1}{\ell+1}}\right).$$

In order to apply this to the total Boolean function, we first notice that the capacity lowerbound holds for every H, hence cutting off the tail of the distribution of $|Y_H|$ could trivially preserve the lowerbound in the worst-case senario while leave the upperbound untouched. Then, we add an independent OR block to calibrate the randomized query complexity between the unrestricted and bounded-space quantum scales. We call the resulted boolean function as $\mathsf{SumOR}_{\ell,n}$, which satisfies $Q < R < Q_S$. We will also discuss about why we chose $M = N^{1/(\ell-1)}$ and the length of the OR block.

Remark 1.1. *When considering the tradeoff of the total boolean function* $\mathsf{SumOR}_{\ell,n}$*, the queries are bit-by-bit. However, ℓ-Sum-Filtered Parity is a one-bit oracle problem, and the queries, by definition, are tuple-by-tuple. But they translate to each other using standard encoding with logarithmic loss, hence not affecting the tradeoff fundamentally.*

2 Problem Setup

2.1 Size of the Filtered Subset

We first analyze the size of the filtered set Y_H.

Lemma 2.1. *Fix a constant integer $\ell \geq 4$. Then*

$$\Pr_H\left[\frac{1}{2N}\binom{M}{\ell} \leq |Y_H| \leq M\right] \geq 1 - O(M^{-1}).$$

Proof. Express the size of the set Y_H, which is defined in Definition 1.1, as a sum of indicator functions.

$$|Y_H| = \sum_{A\in\binom{[M]}{\ell}} \mathbb{1}\left[\sum_{x\in A} H(x) = 0 \pmod N\right].$$

For each $A \in \binom{[M]}{\ell}$, condition on $\ell - 1$ of the ℓ values $\{H(x) : x \in A\}$. Exactly one value of the remaining uniform variable makes the sum zero modulo N. Therefore,

$$\mathbb{E}_H\mathbb{1}\left[\sum_{x\in A} H(x) = 0 \pmod N\right] = \Pr_H\left[\sum_{x\in A} H(x) = 0 \pmod N\right] = \frac{1}{N}.$$

Thus the expected size is $\mathbb{E}_H|Y_H| = \binom{M}{\ell}/N$.

For distinct $A, B \in \binom{[M]}{\ell}$, choose $a \in A \setminus B$ and $b \in B \setminus A$. Condition on every $H(x)$ except $H(a)$ and $H(b)$. The zero-sum equations for A and B then determine $H(a)$ and $H(b)$, respectively. Hence

$$\Pr_H\left[\sum_{x\in A} H(x) = 0 \pmod N \text{ and } \sum_{x\in B} H(x) = 0 \pmod N\right] = \frac{1}{N^2}.$$

Thus, the covariance terms for distinct A and B vanish. It follows that

$$\begin{aligned}\mathrm{Var}_H(|Y_H|) &= \sum_{A\in\binom{[M]}{\ell}} \mathrm{Var}_H\left(\mathbb{1}\left[\sum_{x\in A} H(x) = 0 \pmod N\right]\right)\\ &= \binom{M}{\ell}\frac{1}{N}\left(1-\frac{1}{N}\right) \le \binom{M}{\ell}\frac{1}{N}.\end{aligned}$$

Define $\mu := \mathbb{E}_H|Y_H| = \binom{M}{\ell}\frac{1}{N}$. The identity $M = \lfloor N^{\frac{1}{\ell-1}}\rfloor$ implies $M^{\ell-1} \le N < (M+1)^{\ell-1}$. Thus, $N = \Theta(M^{\ell-1})$. Since ℓ is fixed, $\binom{M}{\ell} = \Theta(M^\ell)$, so $\mu = \Theta(M)$. Moreover, $N \ge M^{\ell-1}$ and $\binom{M}{\ell} \le M^\ell/\ell!$, which give $\mu \le M/\ell!$. Thus, Chebyshev's inequality and the variance bound above give

$$\begin{aligned}\Pr_H[|Y_H| < \mu/2] &\le \frac{4\,\mathrm{Var}_H(|Y_H|)}{\mu^2}\\ &\le \frac{4}{\mu}\\ &= O(M^{-1}).\end{aligned}$$

For the upper tail, the bound on μ gives $M - \mu \ge (1 - 1/\ell!)M$. Applying Chebyshev's inequality again yields

$$\begin{aligned}\Pr_H[|Y_H| > M] &\le \frac{\mathrm{Var}_H(|Y_H|)}{(M-\mu)^2}\\ &\le (1-1/\ell!)^{-2}\frac{\mu}{M^2}\\ &= O(M^{-1}).\end{aligned}$$

A union bound now proves the claim.

□

2.2 Compressed Oracle and the Computational Model

In this section, we explain how queries in the standard quantum random-oracle model (QROM) can be "recorded" using Zhandry's compressed-oracle technique [Zha19]. We focus on the representation needed for ℓ-Sum-Filtered Parity.

We use the following four registers.

- Q contains the query index and answer/target registers.
- W contains all working registers except for Q.

Together, W and Q are working registers used by the algorithm. They are initialized as $|0\rangle$ before the computation, and the joint register WQ contains at most S qubits. The two remaining registers, while not considered as working registers, play an important role in compressed-oracle analysis.

- H is the database register that stores the random oracle H in the compressed oracle view.
- G is the database register that stores the random oracle G in the compressed oracle view.

The compressed standard oracle can be viewed as a type of lazy sampling technique. Instead of designating a particular H or G at the very beginning, the compressed oracle maintains databases on H, G in superposition. Oracle queries are hence interactions between working registers and database registers. Informally speaking, H, G would be initialized as uniform superposition of the states defined as follows,

$$|H\rangle_{\mathrm{H}} := \bigotimes_{j \in [M]} |H(j)\rangle_{\mathrm{H}_j}, \quad |G\rangle_{\mathrm{G}} := \bigotimes_{\mathbf{j} \in [M]^{\ell}} |G(\mathbf{j})\rangle_{\mathrm{G}_{\mathbf{j}}}.$$

Standard oracle. Registers H and G are initialized to $\sum_H |H\rangle_{\mathrm{H}}$ and $\sum_G |G\rangle_{\mathrm{G}}$, respectively, ignoring the normalizing factors. An algorithm accesses the oracle H coherently by applying the unitary O_H^{std}, which acts nontrivially only on Q and H. Specifically,

$$O_H^{\mathrm{std}} |x, z\rangle_{\mathrm{Q}} |H\rangle_{\mathrm{H}} := |x, z \oplus H(x)\rangle_{\mathrm{Q}} |H\rangle_{\mathrm{H}}.$$

The same discussion applies to G, except that G's domain and codomain have different size. Note that an H-query requires $O(\log M + \log N) = O(\log N)$ qubits on Q, while a G-query requires $O(\ell \log M + 1) = O(\log N)$ qubits because ℓ is fixed. Hence they both can be included in an $O(\log N)$ query workspace. We therefore slightly abuse notation and use Q for the query register in both cases. Then

$$O_G^{\mathrm{std}} |\mathbf{x}, b\rangle_{\mathrm{Q}} |G\rangle_{\mathrm{G}} := |\mathbf{x}, b \oplus G(\mathbf{x})\rangle_{\mathrm{Q}} |G\rangle_{\mathrm{G}}.$$

It is the most standard form of oracle queries, and Zhandry [Zha19] showed that it can also be viewed as the following.

Compressed oracle. Write $|\widehat{0}\rangle := (|0\rangle + |1\rangle)/\sqrt{2}$ and $|\widehat{1}\rangle := (|0\rangle - |1\rangle)/\sqrt{2}$ for the Fourier-basis states of the answer qubit. In this basis, an H-query acts non-trivially only on the register Q and H as

$$O_H |x, \widehat{z}\rangle_{\mathrm{Q}} |H\rangle_{\mathrm{H}} = (-1)^{z \cdot H(x)} |x, \widehat{z}\rangle_{\mathrm{Q}} |H\rangle_{\mathrm{H}}.$$

Unlike the standard XOR queries, the Fourier-basis query changes a phase, and hence also "acts on" the database registers. Moreover, if we encode the database registers H and G in the basis $|\widehat{0}\rangle, |\widehat{1}\rangle$, they encode information about the query more clearly.

Let $\mathbf{0} = 00 \cdots 0$ be the all-zero boolean string. Initially, ignoring the normalizing factor, the H register is $\sum_H |H\rangle_{\mathrm{H}} = |\widehat{\mathbf{0}}\rangle_{\mathrm{H}}$. In this representation, an H-phase-query acts as follows:

$$\begin{aligned} &O_H |x, z\rangle_{\mathrm{Q}} \otimes |\widehat{s}\rangle_{\mathrm{H}_x} |\widehat{S}\rangle_{\mathrm{H}_{-x}} \\ &= |x, z\rangle_{\mathrm{Q}} \otimes |\widehat{s \oplus z}\rangle_{\mathrm{H}_x} |\widehat{S}\rangle_{\mathrm{H}_{-x}}. \end{aligned}$$

Here H_x is the database cell register at position $x \in [M]$, while H_{-x} contains the rest.

A similar argument applies to G.

$$\begin{aligned} &O_G |\mathbf{x}, b\rangle_{\mathrm{Q}} \otimes |\widehat{p}\rangle_{\mathrm{G}_{\mathbf{x}}} |\widehat{P}\rangle_{\mathrm{G}_{-\mathbf{x}}} \\ &= |\mathbf{x}, b\rangle_{\mathrm{Q}} \otimes |\widehat{p \oplus b}\rangle_{\mathrm{G}_{\mathbf{x}}} |\widehat{P}\rangle_{\mathrm{G}_{-\mathbf{x}}}. \end{aligned} \tag{1}$$

Theorem 2.2 ([Zha19] and [HHL26, Lemma 4.9 in the full version])**.** *Let $\mathcal{A}$ be an (unbounded) quantum algorithm making oracle queries. The output of $\mathcal{A}$ given access to the compressed oracles O_H, O_G is identical to the output of $\mathcal{A}$ given access to a standard oracle $O_H^{\mathrm{std}}, O_G^{\mathrm{std}}$.*

3 Classical Upper Bound

In this section, we give a classical upper bound for ℓ-Sum-Filtered Parity using the following algorithm.

(1) Query and store $H(x)$ for every $x \in [M]$.

(2) Enumerate the ℓ-tuples in Y_H, query the corresponding G-bits, and compute their parity, stopping with output 0 if more than M such tuples are found.

This requires $O(M \log N)$ space and gives a worst-case $O(M)$ query bound. By Lemma 2.1, the cutoff changes the output only with probability $O(M^{-1})$.

Algorithm 1 Classical algorithm for ℓ-Sum-Filtered Parity

Query and store $H(x)$ for every $x \in [M]$.
answer $\leftarrow 0$ and *count* $\leftarrow 0$.
for all $x_1 < \cdots < x_\ell$ in $[M]$ **do**
    if $\sum_{i=1}^{\ell} H(x_i) = 0 \pmod{N}$ **then**
        if *count* $= M$ **then**
            return 0
        end if
        Query $G(x_1, \ldots, x_\ell)$ and XOR it into *answer*.
        count $\leftarrow$ *count* $+ 1$.
    end if
end for
return *answer*

Proposition 3.1. *Fix a constant integer $\ell \geq 4$. There is a deterministic classical algorithm that outputs $F(H, G)$ with probability $1 - O(M^{-1})$, using M queries to H and at most M queries to G with $O(M \log N)$-bit space.*

Proof. If $|Y_H| \leq M$, Algorithm 1 queries every G-bit appearing in $F(H, G)$ exactly once and returns their parity. Thus, it can fail only when $|Y_H| > M$, which has probability $O(M^{-1})$ by Lemma 2.1. The algorithm makes M queries to H and at most M queries to G. The stored H-table uses $M \log N$ bits; the loop indices, counter, query registers, and output bit use $O(\log N)$ additional space. The algorithm therefore has space complexity $O(M \log N)$. □

4 Parity Forces Recording Capacity

Fix a quantum algorithm $\mathcal{A}$. We show that any advantage in predicting the parity on H-selected domain, i.e., solving ℓ-Sum-Filtered Parity, forces a large expected recording weight on the database register G for random oracle G.

4.1 Introducing the capacity.

We first specify what "large expected recording weight" means. Let $\mathcal{A}$ be the algorithm that aims to solve ℓ-Sum-Filtered Parity. For fixed H, we can omit H as it stores unentangled information about H. During the computation, $\mathcal{A}$ queries G by applying O_G, and the inter-query operations may depend on H. Let $q \leq T$ be the number of G-queries, and let $\Lambda_0^{(H)}, \ldots, \Lambda_q^{(H)}$ be the corresponding inter-query quantum channels on WQ. Each $\Lambda_i^{(H)}$ is implicitly extended by the identity on G. Finally, $\mathcal{A}$ performs a two-outcome POVM on the working registers WQ. Thus, immediately before the final measurement, the joint state is

$$\begin{aligned} \rho_{H,0} &:= \Lambda_0^{(H)}\left(|0\rangle\langle 0|_{\mathrm{WQ}} \otimes |\widehat{0}\rangle\langle\widehat{0}|_{\mathrm{G}}\right), \\ \rho_{H,i+1} &:= \Lambda_{i+1}^{(H)}\left(O_G \rho_{H,i} O_G^{\dagger}\right), \qquad 0 \le i < q, \\ \rho_H &:= \rho_{H,q}. \end{aligned} \tag{2}$$

Here and below, identities on omitted registers are implicit: $\Lambda_i^{(H)}$ acts on WQ, whereas O_G acts on QG.

For $Y \subseteq [M]^{\ell}$, define the parity-flip operator

$$X_Y := \prod_{\mathbf{x}\in Y} X_{\mathrm{G}_{\mathbf{x}}}, \tag{3}$$

and the recording-weight observable

$$W_Y := \sum_{\mathbf{x}\in Y} |\hat{1}\rangle\langle\hat{1}|_{\mathrm{G}_{\mathbf{x}}}. \tag{4}$$

We next define the recording capacity that connects the algorithm's success bias to the HHL capacity bound.

Definition 4.1 (Selected recording capacity)**.** *Focus on algorithm $\mathcal{A}$ with fixed choice of $H\colon [M] \to [N]$ first. Define*

$$K_H(\mathcal{A}) := \mathrm{Tr}\big[\rho_H\big(I_{\mathrm{WQ}} \otimes W_{Y_H}\big)\big],$$

and define the selected recording capacity of $\mathcal{A}$ as $K(\mathcal{A}) := \mathbb{E}_H[K_H(\mathcal{A})]$, where the expectation is over a uniformly random H.

4.2 The success probability as observable value

Lemma 4.1. *Represent the final binary POVM of $\mathcal{A}$ by a Hermitian operator B on* WQ *with $\|B\| \le 1$. Define the conditional success bias as $\beta_H := 2\Pr_{G,\mathcal{A}}[\mathcal{A}^{H,G} = F(H,G)] - 1$. Then*

$$\beta_H = \mathrm{Tr}\big[\rho_H\big(B \otimes X_{Y_H}\big)\big].$$

Proof. For each G, let $\rho_{H,G}$ be the premeasurement density operator on WQ when the Boolean oracle is G. Before Fourier transforming the truth-table register G, let σ_H be the joint density operator. Its diagonal blocks satisfy

$$\langle G|_{\mathrm{G}}\sigma_H|G\rangle_{\mathrm{G}} = 2^{-M^{\ell}}\rho_{H,G}.$$

For fixed G, the definition of B gives

$$2\Pr_{\mathcal{A}}[\mathcal{A}^{H,G} = F(H,G)] - 1 = (-1)^{F(H,G)}\,\mathrm{Tr}(B\rho_{H,G}).$$

Because G is uniform, averaging this identity over G gives

$$\begin{aligned} \beta_H &= 2^{-M^{\ell}} \sum_{G\colon [M]^{\ell}\to\{0,1\}} \left(2\Pr_{\mathcal{A}}[\mathcal{A}^{H,G} = F(H,G)] - 1\right) \\ &= 2^{-M^{\ell}} \sum_{G\colon [M]^{\ell}\to\{0,1\}} (-1)^{F(H,G)}\,\mathrm{Tr}(B\rho_{H,G}). \end{aligned}$$

For each $\mathbf{x}$, let $Z_{\mathrm{G}_\mathbf{x}}$ act on the truth-table qubit indexed by $\mathbf{x}$. Since $F(H,G) = \bigoplus_{\mathbf{x}\in Y_H} G(\mathbf{x})$,

$$\left(\prod_{\mathbf{x}\in Y_H} Z_{\mathrm{G}_\mathbf{x}}\right)|G\rangle_{\mathrm{G}} = (-1)^{F(H,G)}|G\rangle_{\mathrm{G}}.$$

Substituting the diagonal-block identity above and using the orthogonality of the states $|G\rangle_{\mathrm{G}}$ now yields

$$\begin{aligned}&\mathrm{Tr}\left[\sigma_H\left(B\otimes\prod_{\mathbf{x}\in Y_H} Z_{\mathrm{G}_\mathbf{x}}\right)\right]\\&= 2^{-M^\ell}\sum_{G\colon [M]^\ell\to\{0,1\}}(-1)^{F(H,G)}\,\mathrm{Tr}(B\rho_{H,G})\\&= \beta_H.\end{aligned}$$

Finally, the Hadamard transform changes σ_H into ρ_H and maps each $Z_{\mathrm{G}_\mathbf{x}}$ to $X_{\mathrm{G}_\mathbf{x}}$. Hence, by Equation (3),

$$\beta_H = \mathrm{Tr}\left[\rho_H\left(B\otimes\prod_{\mathbf{x}\in Y_H} X_{\mathrm{G}_\mathbf{x}}\right)\right] = \mathrm{Tr}[\rho_H(B\otimes X_{Y_H})]\,.$$

□

4.3 An exact parity-to-capacity inequality

Lemma 4.2. *For every fixed H, $K_H(\mathcal{A}) \geq |Y_H|\beta_H^2/4$.*

Proof. Since we are working with fixed H, write $\rho := \rho_H$, $Y := Y_H$, $K := K_H(\mathcal{A})$, and $\beta := \beta_H$. If $Y = \varnothing$, then $K = 0$ and the claim is immediate. Assume that $Y \neq \varnothing$, and define the Hermitian operators

$$O_1 := I_{\mathrm{WQ}}\otimes\left(I_{\mathrm{G}} - \frac{2}{|Y|}W_Y\right), \qquad O_2 := B\otimes X_Y.$$

The spectrum of W_Y is contained in $\{0,\dots,|Y|\}$. Together with $\|B\|\leq 1$ and $X_Y^2 = I_{\mathrm{G}}$, this gives

$$O_1^2 \preceq I, \qquad O_2^2 = B^2\otimes I_{\mathrm{G}} \preceq I.$$

The operator X_Y flips each recording qubit indexed by Y, so $X_Y W_Y X_Y = |Y|I_{\mathrm{G}} - W_Y$. Consequently,

$$\begin{aligned}&X_Y\left(I_{\mathrm{G}} - \frac{2}{|Y|}W_Y\right)X_Y\\&= I_{\mathrm{G}} - \frac{2}{|Y|}\left(|Y|I_{\mathrm{G}} - W_Y\right)\\&= -\left(I_{\mathrm{G}} - \frac{2}{|Y|}W_Y\right).\end{aligned}$$

This identity gives $O_1O_2 + O_2O_1 = 0$.

For $a,b\in\mathbb{R}$, set $O_{a,b} := aO_1 + bO_2$. The preceding operator inequalities and the anticommutation relation imply

$$\begin{aligned}O_{a,b}^2 &= a^2O_1^2 + b^2O_2^2 + ab(O_1O_2 + O_2O_1)\\&\preceq (a^2+b^2)I.\end{aligned}$$

By the definitions of K and β,

$$\mathrm{Tr}(\rho O_1) = 1 - \frac{2K}{|Y|}, \qquad \mathrm{Tr}(\rho O_2) = \beta.$$

Since $O_{a,b}$ is Hermitian, Cauchy–Schwarz gives

$$\begin{aligned}\left|a\left(1 - \frac{2K}{|Y|}\right) + b\beta\right|^2 &= |\mathrm{Tr}(\rho O_{a,b})|^2 \\ &\le \mathrm{Tr}(\rho O_{a,b}^2) \\ &\le a^2 + b^2.\end{aligned}$$

By Cauchy–Schwarz again,

$$\max_{a^2+b^2=1} \left|a\left(1 - \frac{2K}{|Y|}\right) + b\beta\right|^2 = \left(1 - \frac{2K}{|Y|}\right)^2 + \beta^2.$$

Combining this identity with the preceding bound yields $(1 - 2K/|Y|)^2 + \beta^2 \le 1$. In particular, $K \ge |Y|(1 - \sqrt{1-\beta^2})/2$. Finally,

$$1 - \sqrt{1-\beta^2} = \frac{\beta^2}{1 + \sqrt{1-\beta^2}} \ge \frac{\beta^2}{2},$$

which proves the desired inequality. □

4.4 Averaging over H

Using Definition 4.1, we now combine the fixed-H inequality with the concentration of Y_H to lower-bound $K(\mathcal{A})$.

Theorem 4.3. *Fix $0 < \varepsilon \le 1/2$, and let $\mathcal{A}$ be a quantum algorithm for ℓ-*Sum-Filtered Parity*. Let $K(\mathcal{A})$ be its selected recording capacity as defined in Definition 4.1. If $\mathcal{A}$ outputs $F(H,G)$ with probability at least $1/2 + \varepsilon$, then, for all sufficiently large N,*

$$K(\mathcal{A}) \ge \binom{M}{\ell}\frac{\varepsilon^2}{8N} = \Omega(\varepsilon^2 M).$$

Proof. For each H, let β_H be the conditional success bias from Lemma 4.1. Set

$$\mu := \mathbb{E}_H[|Y_H|] = \binom{M}{\ell}\frac{1}{N}, \qquad \mathcal{E} := \{H : |Y_H| \ge \mu/2\}.$$

By Lemma 2.1, $\Pr_H[\neg\mathcal{E}] = O(M^{-1})$. Since $\varepsilon > 0$ is fixed, this probability is at most ε for all sufficiently large N.

The success assumption gives

$$\mathbb{E}_H[\beta_H] = 2\Pr_{H,G,\mathcal{A}}\big[\mathcal{A}^{H,G} = F(H,G)\big] - 1 \ge 2\varepsilon.$$

Since $\beta_H \le 1$ for every H,

$$\begin{aligned}\mathbb{E}_H[\mathbb{1}_{\mathcal{E}}\beta_H] &= \mathbb{E}_H[\beta_H] - \mathbb{E}_H[\mathbb{1}_{\neg\mathcal{E}}\beta_H] \\ &\ge 2\varepsilon - \Pr_H[\neg\mathcal{E}] \\ &\ge \varepsilon.\end{aligned}$$

Cauchy–Schwarz therefore gives

$$\begin{aligned}\varepsilon^2 &\le (\mathbb{E}_H[\mathbb{1}_{\mathcal{E}}\beta_H])^2 \\ &\le \Pr_H[\mathcal{E}]\,\mathbb{E}_H[\mathbb{1}_{\mathcal{E}}\beta_H^2] \\ &\le \mathbb{E}_H[\mathbb{1}_{\mathcal{E}}\beta_H^2].\end{aligned}$$

Lemma 4.2 gives $K_H \ge |Y_H|\beta_H^2/4$. Using Definition 4.1 and restricting the expectation to $\mathcal{E}$, we obtain

$$\begin{aligned}K(\mathcal{A}) &= \mathbb{E}_H[K_H] \\ &\ge \frac{1}{4}\mathbb{E}_H[|Y_H|\beta_H^2] \\ &\ge \frac{\mu}{8}\mathbb{E}_H[\mathbb{1}_{\mathcal{E}}\beta_H^2] \\ &\ge \frac{\mu\varepsilon^2}{8}.\end{aligned}$$

Substituting $\mu = \binom{M}{\ell}/N$ proves the explicit bound. Since $M = \lfloor N^{\frac{1}{\ell-1}} \rfloor$, we also have $\mu = \Theta(M)$, which proves the asymptotic bound. □

5 Capacity Upper Bound from HHL

We show that a bounded-space, bounded-query algorithm cannot produce large capacity $K(\mathcal{A})$.

Theorem 5.1 (Selected-capacity upper bound). *Fix a constant integer $\ell \ge 4$, let $\mathcal{A}$ be a uniform quantum algorithm using at most S qubits and T oracle queries. If $S = \Omega(\log N)$ and $S = O\big(N^{1/((\ell+1)(2\ell+1))}\big)$, then*

$$K(\mathcal{A}) = O\Big(S^{2\ell+2}T^2N^{-2/(\ell+1)}\Big).$$

In particular, for $\ell = 4$, $K(\mathcal{A}) = O\big(S^{10}T^2N^{-2/5}\big)$ whenever $S = \Omega(\log N)$ and $S = O(N^{1/45})$.

Proof. We use the following one-bit specialization of HHL's capacity bound. For each H, define the unrestricted ordered zero-sum set

$$\overline{Y}_H := \left\{\mathbf{x} = (x_1,\ldots,x_\ell) \in [M]^\ell : \sum_{i=1}^{\ell} H(x_i) = 0 \pmod N\right\}.$$

This is the set used by the HHL capacity for the constant target sequence $\{y_H\}$ given by $y_H = 0$ for every H. Unlike Y_H, the set $\overline{Y}_H$ allows repeated coordinates and all orderings. The strict-ordering condition in Definition 1.1 gives $Y_H \subseteq \overline{Y}_H$. Let $\overline{K}$ be the recorded capacity obtained from K in Definition 4.1 by replacing Y_H with $\overline{Y}_H$.

Theorem 5.2 (One-bit specialization of [HHL26, Theorem 9.3 in the full version]). *Fix a constant integer $\ell \ge 4$, and let H and G be the two random oracles in ℓ-Sum-Filtered Parity. Let $\mathcal{A}$ be a uniform quantum algorithm using S qubits, at most T queries to H, and at most T queries to G. Then, we have*[1]

$$\overline{K}(\mathcal{A}) = O\Big(S^{2\ell+2}T^2N^{-2/(\ell+1)}\Big)$$

for $S = \Omega(\log N)$ and $S = O(N^{1/((\ell+1)(2\ell+1))})$.

[1] In [HHL26]'s original notation, $N_0 = 2$ and hence $n_0 = \log N_0 = 1$, so their bound reduces to the special form.

For every H,

$$W_{\overline{Y}_H} - W_{Y_H} = \sum_{\mathbf{x}\in\overline{Y}_H\setminus Y_H} |1\rangle\langle 1|_{\mathrm{G}_{\mathbf{x}}} \succeq 0.$$

Taking the trace of this operator against ρ_H and then averaging over H gives $K(\mathcal{A}) \le \overline{K}(\mathcal{A})$.

Thus, Theorem 5.2 gives

$$K(\mathcal{A}) = O\Big(S^{2\ell+2}T^2N^{-2/(\ell+1)}\Big).$$

Substituting $\ell = 4$ gives $2\ell+2 = 10$, $2/(\ell+1) = 2/5$, and $(\ell+1)(2\ell+1) = 45$, which proves the specialization. □

6 Proof of the Separation

Proof of Theorem 1.1. For part (i), check Proposition 3.1.

For part (ii), fix $0 < \varepsilon \le 1/2$ and a quantum algorithm $\mathcal{A}$ as in the theorem. By Lemma 2.1 and Theorem 4.3,

$$K(\mathcal{A}) \ge \binom{M}{\ell}\frac{\varepsilon^2}{8N} = \Omega\Big(N^{\frac{1}{\ell-1}}\Big).$$

By Theorem 5.1,

$$K(\mathcal{A}) = O\Big(S^{2\ell+2}T^2N^{-2/(\ell+1)}\Big),$$

when $S = \Omega(\log N)$ and $S = O(N^{1/((\ell+1)(2\ell+1))})$. Comparing these two bounds yields

$$S^{2\ell+2}T^2N^{-2/(\ell+1)} = \Omega\Big(N^{\frac{1}{\ell-1}}\Big).$$

Since S and T are non-negative:

$$S^{\ell+1}T = \Omega\Big(N^{\frac{1}{2(\ell-1)}+\frac{1}{\ell+1}}\Big).$$ □

7 Space-sensitive Quantum Query Advantage for a Total Boolean Function

7.1 Construction and Main Result

The construction in ℓ-Sum-OR Parity differs from *ℓ-Sum-Filtered Parity* defined in Definition 1.1 in three aspects.

1. We turn the original problem concerning random functions into a worst-case statement about a total function. Hence the three input "oracles" are now arbitrary truth tables instead of random functions. We remark that here the truth table for H should be bit encoded, so querying one whole value $H(x) \in [N]$ costs $\log N$ bit queries.

2. The function discards the filtered-parity term when the filtered set, Y_H, is too large (specifically, when $|Y_H| > \left\lceil 2\binom{M}{\ell}/N \right\rceil$).

3. A disjoint OR block is XORed with the parity block.

Theorem 7.1 (Total quantum speedup and small-space reversal)**.** *For every sufficiently large integer n, the following bounds hold for ℓ-*Sum-OR Parity Problem*:*

$$D = R = \widetilde{\Theta}\left(M^{1+(\ell-3)/(4(\ell+1))}\right),$$
$$Q = \widetilde{\Theta}(M).$$

An deterministic classical algorithm uses $O(M \log M)$ bits of space. Fix a constant $0 < \varepsilon \leq 1/2$. Let a uniform quantum algorithm, that uses S qubits and T total queries, succeeds with probability at least $1/2 + \varepsilon$ in the worst case. Then $T = \omega\left(M^{1+(\ell-3)/(4(\ell+1))}\right)$ when $\Omega(\log M) \leq S \leq M^{(\ell-3)/(4(\ell+1)^2)-\eta}$, where η is a fixed positive constant satisfying $0 < \eta < \frac{\ell-3}{4(\ell+1)^2}$. In fact, the following tradeoff holds:

$$S^{\ell+1}T = \widetilde{\Omega}\left(\varepsilon M^{3/2-2/(\ell+1)}\right).$$

when $S = \Omega(\log M)$ and $S = O\left(M^{(\ell-1)/((\ell+1)(2\ell+1))}\right)$.

Substituting $\ell = 4$ immediately gives the following corollary.

Corollary 7.2. *For $\ell = 4$, let $M = \lfloor N^{1/3} \rfloor$. Set $L = \lceil M^{21/20} \rceil$. For algorithms with constant advantage, if they are space-unrestricted, then*

$$D = R = \Theta(M^{21/20}) = \Theta(N^{7/20}),$$
$$Q = \widetilde{\Theta}(M) = \widetilde{\Theta}(N^{1/3}).$$

Hence quantum algorithms are strictly faster than classical ones. Consider algorithms using space S, and query bound T, the following tradeoff holds:

$$S^5 T = \widetilde{\Omega}(M^{11/10}) = \widetilde{\Omega}(N^{11/30})$$

when $\Omega(\log M) \leq S \leq O(M^{1/15})$. Consequently, $T = \omega(M^{21/20}) = \omega(N^{7/20})$ when $\Omega(\log M) \leq S \leq M^{1/100-\eta}$ and η is a constant satisfying $0 < \eta < 1/100$. In this space range, the quantum query exceeds the classical one.

The rest of this section proves Theorem 7.1 and Corollary 7.2: Proposition 7.7 gives the unrestricted query complexities, and Proposition 7.9 gives the bounded-space tradeoff and reversal.

7.2 Standard Query Facts

We will use the following standard facts about OR and parity.

Lemma 7.3. *For OR on L bits, $R(\mathrm{OR}_L) = \Omega(L)$ and $Q(\mathrm{OR}_L) = O(\sqrt{L})$.*

Lemma 7.4. *For parity on k bits, $Q(\mathrm{PARITY}_k) = \Omega(k)$.*

We also need to encode the random oracles as Boolean tables, which is specifically listed in the following lemma.

Lemma 7.5. *Let $F : [K] \to [R]$ be an oracle. Assume that K and R are powers of two, and set $r = \log R$. Fix an r-bit encoding of the range of F. The standard oracle query maps*

$$|x, y\rangle \mapsto |x, y \oplus F(x)\rangle.$$

The corresponding bit-query maps

$$|x, j, b\rangle \mapsto |x, j, b \oplus F_j(x)\rangle,$$

where $j \in [r]$ and $F_j(x)$ is the jth bit of the encoding of $F(x)$. Then the following simulations hold.

1. *If a quantum algorithm $\mathcal{A}$ uses S qubits and T bit queries, then a standard-oracle algorithm $\mathcal{B}$ has the same output distribution, uses $S + O(r)$ qubits, and $2T$ standard queries.*

2. *If a quantum algorithm $\mathcal{A}$ uses S qubits and T standard queries to F, then a bit-query algorithm $\mathcal{B}$ has the same output distribution, uses $S + O(\log r)$ qubits, and rT bit queries.*

Proof. Bit queries from standard queries. The simulator $\mathcal{B}$ runs the circuit of $\mathcal{A}$ and simulates each bit query. To simulate a query to $F_j(x)$, it uses an r-qubit scratch register initialized to zero. It first queries the standard oracle to write $F(x)$ into the scratch register. It CNOTs the jth scratch bit into the answer qubit of the bit query. It then queries the standard oracle again to restore the scratch register to zero. On each computational-basis query state, this is exactly

$$|x, j, b\rangle \mapsto |x, j, b \oplus F_j(x)\rangle.$$

Thus every bit query costs at most two standard queries.

Standard queries from bit queries. The simulator $\mathcal{B}$ again runs the circuit of $\mathcal{A}$. To simulate one standard query, it applies the bit-query oracle once for each $j \in [r]$, their product is exactly

$$|x, y\rangle \mapsto |x, y \oplus F(x)\rangle.$$

Thus every standard query costs at most r bit queries. $\square$

We've introduced Lemma 2.1 to analyze the size of the filtered set Y_H. However, the following corollary would still simplify the analysis of the Boolean version of the problem.

Corollary 7.6. *Set $\mu := \binom{M}{\ell}/N$. For a uniform function $H : [M] \to [N]$,*

$$\Pr_H\left[\frac{\mu}{2} \leq |Y_H| \leq 2\mu\right] \geq 1 - \frac{5}{\mu}.$$

Proof. The moment calculation in the proof of Lemma 2.1 applies here. It gives $\mathbb{E}_H[|Y_H|] = \mu$ and $\mathrm{Var}_H(|Y_H|) \leq \mu$. Chebyshev's inequality and a union bound proves the claim. $\square$

7.3 Unrestricted Query Complexities

We next determine the three unrestricted query complexities.

Proposition 7.7. *The unrestricted query complexities satisfy*

$$D = R = \Theta\left(M^{1+(\ell-3)/(4(\ell+1))}\right),$$
$$Q = \widetilde{\Theta}(M).$$

Moreover, the upper bound for D is achieved by a deterministic algorithm using $O(M \log M)$ bits.

Proof. The classical part. There is a simple algorithm that shows the upper bound. It first query and store all $M \log N$ bits of H, then enumerate Y_H locally and stop counting after $\lceil 2\mu \rceil + 1$ tuples. If the count exceeds $\lceil 2\mu \rceil$, query every bit of z and return its OR. Otherwise, enumerate Y_H again and query all selected G-bits, XOR their values, and then XOR the result with the OR of z. This algorithm makes at most $M \log N + \lceil 2\mu \rceil + L = \Theta(L)$ queries and uses $O(M \log M)$ bits of space. Recall that $L = \lceil M^{1+(\ell-3)/(4(\ell+1))} \rceil$ shows the desired upper bound. As for the lower bound, fixing every G-bit to zero restricts the function to OR on L bits. Apply Lemma 7.3 to this restriction proves the randomized query complexity $\Omega(L)$.

The quantum part. For the quantum upper bound, the same algorithm described above still works, and we can speed the OR part by Grover search. Hence the quantum query complexity is at most

$$O(M \log N + \lceil 2\mu \rceil + \sqrt{L}) = \widetilde{O}(M).$$

Here we use the fact that $L = o(M^2)$ for every fixed $\ell \geq 4$.

By Corollary 7.6, some function H^* satisfies $\mu/2 \leq |Y_{H^*}| \leq 2\mu$. Hence $Y_{H^*} = \Theta(M)$. Fix this H^* and fix z to the all-zero string. The expected output of the resulting function would be parity on $\Theta(M)$ distinct bits. By Lemma 7.4, it requires $\Omega(M)$ queries. This proves the quantum lower bound. □

7.4 The Bounded-Space Reversal

Lemma 7.8. *Fix $0 < \varepsilon \leq 1/2$. Let $\mathcal{A}$ compute ℓ-Sum-OR Parity with worst-case success is at least $1/2 + \varepsilon$. After fixing z to zero, its selected recording capacity satisfies $K(\mathcal{A}) = \Omega(\varepsilon^2 M)$.*

Proof. Consider an H^* satisfying $\mu/2 \leq |Y_{H^*}| \leq 2\mu$. The threshold rule does not fire, so the target is parity on Y_{H^*}. By Lemma 4.1 and the inequality in Lemma 4.2, $K_{H^*}(\mathcal{A}) \geq \mu\varepsilon^2/2$. Averaging this bound and applying Corollary 7.6 gives

$$K(\mathcal{A}) \geq \left(1 - \frac{5}{\mu}\right) \frac{\mu\varepsilon^2}{2} = \Omega(\varepsilon^2 M).$$ □

Proposition 7.9 (Bounded-space tradeoff and reversal)**.** *Fix a constant $0 < \varepsilon \leq 1/2$. Let a uniform quantum algorithm compute ℓ-Sum-OR Parity with worst-case success at least $1/2 + \varepsilon$. Suppose that it uses S qubits and makes T total queries. If $S = \Omega(\log M)$ and $S = O\left(M^{(\ell-1)/((\ell+1)(2\ell+1))}\right)$, then*

$$S^{\ell+1} T = \widetilde{\Omega}\left(\varepsilon M^{3/2 - 2/(\ell+1)}\right). \tag{5}$$

Proof. Fix z to zero, and omit the queries to the z truth table. Writing the remaining G-queries in the equivalent Hadamard representation does not change the live space or query count. Together with the translation by Lemma 7.5, Theorem 5.1 applies to the resulting algorithm. Combining it with Lemma 7.8 yields, for an absolute constant C,

$$\varepsilon^2 M \leq \widetilde{O}\left((S + C \log N)^{2\ell+2} T^2 N^{-2/(\ell+1)}\right).$$

Using $N = \Theta(M^{\ell-1})$ and taking square roots gives

$$(S + C \log N)^{\ell+1} T = \widetilde{\Omega}\left(\varepsilon M^{3/2 - 2/(\ell+1)}\right).$$

Since $S = \Omega(\log M)$ and $\log N = \Theta(\log M)$, this simplifies to the displayed bound. This proves the tradeoff. □

8 Choosing the Parameters

8.1 Choosing the OR Block

In Section 7 we choose the OR block length L to be $M^{1+(\ell-3)/(4(\ell+1))}$ (without the ceiling). It is the midpoint of the interval that allows both the unrestricted advantage and the small-space reversal. The next proposition shows that in detail.

Proposition 8.1. *Replace the relation in the definition by $N = M^{p+o(1)}$, where $0 < p < \ell$. Consider uniform quantum algorithms with advantage $\varepsilon \in (0, 1/2]$ and space S. Assume that $S = M^{o(1)}$ and $S = \Omega(\log M)$. Let $c_\ell(p)$ denote the exponent of the deterministic classical query upper bound for the filtered-parity block. Let $q_\ell(p)$ denote the exponent yielded by the bounded-space quantum query lower bound. That is,*

$$c_\ell(p) := \max\{1, \ell - p\}, \qquad q_\ell(p) := \frac{\ell}{2} - \frac{p(\ell-1)}{2(\ell+1)}.$$

Let the OR block have length $L = M^{\alpha+o(1)}$. The unrestricted advantage and the small-space reversal both hold when $c_\ell(p) < \alpha < q_\ell(p)$. The width of this interval is uniquely maximized at $p = \ell - 1$. At this value, its midpoint is $\alpha = 1 + (\ell-3)/(4(\ell+1))$.

Proof. We first identify the two exponents in the statement. The classical upper bound could be derived from the algorithm we described before.

$$\begin{aligned} M^{c_\ell(p)+o(1)} &= M + \binom{M}{\ell}/N \\ &= M^{\max\{1,\ell-p\}+o(1)}. \end{aligned}$$

We used the fact that $\binom{M}{\ell}/N = M^{\ell-p+o(1)}$. This explains the definition of $c_\ell(p)$. The exponent $q_\ell(p)$ comes from the capacity lower bound. Specifically, $M^{q_\ell(p)} = N^{1/(\ell+1)}\sqrt{\binom{M}{\ell}/N}$. Substituting $N = M^{p+o(1)}$ gives the following exponent.

$$\begin{aligned} \frac{\ell-p}{2} + \frac{p}{\ell+1} &= \frac{\ell}{2} - \frac{p}{2} + \frac{p}{\ell+1} \\ &= \frac{\ell}{2} - \frac{p(\ell-1)}{2(\ell+1)} \\ &= q_\ell(p). \end{aligned}$$

When $\alpha > c_\ell(p)$, the deterministic algorithm has query cost $M^{\alpha+o(1)}$. The OR restriction gives the matching randomized lower bound. Thus $D = R = M^{\alpha+o(1)}$ in this regime. The unrestricted quantum algorithm gives the upper bound

$$Q \le M^{\max\{\alpha/2, c_\ell(p)\}+o(1)}.$$

Hence $\alpha > c_\ell(p)$ gives the unrestricted quantum advantage. For $S = M^{o(1)}$, the capacity lower-bound argument gives

$$Q_S \ge M^{q_\ell(p)-o(1)}.$$

Thus $\alpha < q_\ell(p)$ gives the small-space reversal. It is therefore enough to choose $c_\ell(p) < \alpha < q_\ell(p)$. The choice $\alpha = 1 + (\ell-3)/(4(\ell+1))$ and $p = \ell - 1$ satisfies these inequalities. We choose them since the choice of p maximizes the width of the interval $\big(c_\ell(p), q_\ell(p)\big)$. Specifically speaking, for $0 < p \le \ell - 1$,

$$q_\ell(p) - c_\ell(p) = -\frac{\ell}{2} + \frac{\ell+3}{2(\ell+1)}p.$$

For $\ell - 1 \le p < \ell$,

$$q_\ell(p) - c_\ell(p) = \frac{\ell}{2} - 1 - \frac{\ell-1}{2(\ell+1)}p.$$

The first branch is strictly increasing in p. The second branch is strictly decreasing in p. The unique maximum is therefore attained at $p = \ell - 1$. At $p = \ell - 1$, we have $c_\ell(\ell-1) = 1$ and $q_\ell(\ell-1) = 3/2 - 2/(\ell+1)$. Hence we need to choose α such that $1 < \alpha < 3/2 - 2/(\ell+1)$. So we just choose the midpoint of it. $\square$

8.2 Choosing the Filter Scale

The construction sets $M = N^{1/(\ell-1)}$ for another reason, that is, make the selected set Y_H has the same order as the list of H-values. To see this, ignore floors and write $N = M^{p+o(1)}$; then $\mu := \mathbb{E}_H[|Y_H|] = \binom{M}{\ell}/N = M^{\ell-p+o(1)}$. The classical deterministic algorithm that stores H and reads the selected G-bits has parity-block query scale $M + \binom{M}{\ell}/N$, which is balanced when $\ell - p = 1$. Equivalently, $M = N^{1/(\ell-1)}$.

Acknowledgements and AI-disclosure

The authors used Large Language Models as AI-assisted research and writing tools throughout the preparation of this manuscript. The tool was used to help brainstorm ideas and explore proof strategies. Portions of the manuscript text were redrafted or modified with AI assistance across all sections. All final mathematical claims, algorithms, proofs, citations, and wording were reviewed, edited, and validated by the authors. The authors assume responsibility for all content of the paper.

References

[Aar21] Scott Aaronson. Open problems related to quantum query complexity. *ACM Transactions on Quantum Computing*, 2(4):14:1–14:9, December 2021. doi:10.1145/3488559. 3

[Amb07] Andris Ambainis. Quantum Walk Algorithm for Element Distinctness. *SIAM Journal on Computing*, 37(1):210–239, January 2007. doi:10.1137/S0097539705447311. 3

[AS04] Scott Aaronson and Yaoyun Shi. Quantum lower bounds for the collision and the element distinctness problems. *Journal of the ACM*, 51(4):595–605, July 2004. doi:10.1145/1008731.1008735. 3

[BC82] A. Borodin and S. Cook. A Time-Space Tradeoff for Sorting on a General Sequential Model of Computation. *SIAM Journal on Computing*, 11(2):287–297, May 1982. doi:10.1137/0211022. 3

[Bea91] Paul Beame. A general Sequential Time-Space Tradeoff for Finding Unique Elements. *SIAM Journal on Computing*, 20(2):270–277, April 1991. doi:10.1137/0220017. 3

[BFK+81] Allan Borodin, Michael J. Fischer, David G. Kirkpatrick, Nancy A. Lynch, and Martin Tompa. A time-space tradeoff for sorting on non-oblivious machines. *Journal of Computer and System Sciences*, 22(3):351–364, June 1981. doi:10.1016/0022-0000(81)90037-4. 3

[BFMADH+87] A. Borodin, F. Fich, F. Meyer Auf Der Heide, E. Upfal, and A. Wigderson. A Time-Space Tradeoff for Element Distinctness. *SIAM Journal on Computing*, 16(1):97–99, February 1987. doi:10.1137/0216007. 3

[BHT97] Gilles Brassard, Peter Høyer, and Alain Tapp. Quantum cryptanalysis of hash and claw-free functions. *ACM SIGACT News*, 28(2):14–19, June 1997. doi:10.1145/261342.261346. 3

[BJS01] Paul Beame, T. S. Jayram, and Michael Saks. Time-space tradeoffs for branching programs. *Journal of Computer and System Sciences*, 63(4):542–572, December 2001. Special issue of selected papers from FOCS 1998. doi:10.1006/jcss.2001.1778. 3

[BKW26] Paul Beame, Niels Kornerup, and Michael Whitmeyer. Quantum Time-Space Tradeoffs for Matrix Problems. *SIAM Journal on Computing*, 55(3):469–519, June 2026. doi:10.1137/24M1710164. 3

[BS24] Debajyoti Bera and Tharrmashastha SAPV. Quantum Query-Space Lower Bounds Using Branching Programs. *CoRR*, abs/2407.06872, 2024. arXiv:2407.06872, doi:10.48550/arXiv.2407.06872. 3

[BSSV03] Paul Beame, Michael Saks, Xiaodong Sun, and Erik Vee. Time-space trade-off lower bounds for randomized computation of decision problems. *Journal of the ACM*, 50(2):154–195, March 2003. doi:10.1145/636865.636867. 3

[CGLQ20] Kai-Min Chung, Siyao Guo, Qipeng Liu, and Luowen Qian. Tight Quantum Time-Space Tradeoffs for Function Inversion. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 673–684, Durham, NC, USA, November 2020. IEEE. doi:10.1109/FOCS46700.2020.00068. 3

[Din20] Itai Dinur. Tight Time-Space Lower Bounds for Finding Multiple Collision Pairs and Their Applications. In *Advances in Cryptology – EUROCRYPT 2020*, volume 12105 of *Lecture Notes in Computer Science*, pages 405–434. Springer, 2020. doi:10.1007/978-3-030-45721-1_15. 3

[Gro96] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing - STOC '96*, pages 212–219, Philadelphia, Pennsylvania, United States, 1996. ACM Press. doi:10.1145/237814.237866. 3

[HHL26] Zihan Hao, Zikuan Huang, and Qipeng Liu. On the Need for (Quantum) Memory with Short Outputs. In *Proceedings of the 58th Annual ACM Symposium on Theory of Computing*, pages 1901–1912, Salt Lake City UT USA, June 2026. ACM. Full version: ECCC TR26-031. arXiv:2602.23763, doi:10.1145/3798129.3800896. 1, 3, 4, 5, 8, 13

[HM23] Yassine Hamoudi and Frédéric Magniez. Quantum Time–Space Tradeoff for Finding Multiple Collision Pairs. *ACM Transactions on Computation Theory*, 15(1-2):1–22, June 2023. doi:10.1145/3589986. 3

[Kla03] Hartmut Klauck. Quantum time-space tradeoffs for sorting. In *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing*, pages 69–76, San Diego CA USA, June 2003. ACM. doi:10.1145/780542.780553. 3

[KŠdW07] Hartmut Klauck, Robert Špalek, and Ronald de Wolf. Quantum and Classical Strong Direct Product Theorems and Optimal Time-Space Tradeoffs. *SIAM Journal on Computing*, 36(5):1472–1493, January 2007. doi:10.1137/05063235X. 3

[Yao94] Andrew Chi-Chih Yao. Near-Optimal Time-Space Tradeoff for Element Distinctness. *SIAM Journal on Computing*, 23(5):966–975, October 1994. doi:10.1137/S0097539788148959. 3

[Zha19] Mark Zhandry. How to Record Quantum Queries, and Applications to Quantum Indifferentiability. In *Advances in Cryptology – CRYPTO 2019*, volume 11693 of *Lecture Notes in Computer Science*, pages 239–268. Springer, 2019. doi:10.1007/978-3-030-26951-7_9. 5, 7, 8

ECCC ISSN 1433-8092
https://eccc.weizmann.ac.il