







Quantum Search with In-Place Queries

Blake Holman   

Sandia National Laboratories, Albuquerque, NM, USA
Purdue University, West Lafayette, IN, USA

Ronak Ramachandran   

The University of Texas at Austin, TX, USA

Justin Yirka   

Sandia National Laboratories, Albuquerque, NM, USA
The University of Texas at Austin, TX, USA

Abstract

Quantum query complexity is typically characterized in terms of XOR queries $|x, y\rangle \mapsto |x, y \oplus f(x)\rangle$ or phase queries, which ensure that even queries to non-invertible functions are unitary. When querying a permutation, another natural model is unitary: in-place queries $|x\rangle \mapsto |f(x)\rangle$.

Some problems are known to require exponentially fewer in-place queries than XOR queries, but no separation has been shown in the opposite direction. A candidate for such a separation was the problem of inverting a permutation over N elements. This task, equivalent to unstructured search in the context of permutations, is solvable with $O(\sqrt{N})$ XOR queries but was conjectured to require $\Omega(N)$ in-place queries.

We refute this conjecture by designing a quantum algorithm for Permutation Inversion using $O(\sqrt{N})$ in-place queries. Our algorithm achieves the same speedup as Grover's algorithm despite the inability to efficiently uncompute queries or perform straightforward oracle-controlled reflections.

Nonetheless, we show that there are indeed problems which require fewer XOR queries than in-place queries. We introduce a subspace-conversion problem called Function Erasure that requires 1 XOR query and $\Theta(\sqrt{N})$ in-place queries. Then, we build on a recent extension of the quantum adversary method to characterize exact conditions for a decision problem to exhibit such a separation, and we propose a candidate problem.

2012 ACM Subject Classification Theory of computation \rightarrow Quantum query complexity

Keywords and phrases Quantum algorithms, query complexity, quantum complexity theory, quantum search, Grover's algorithm, permutation inversion

Digital Object Identifier 10.4230/LIPIcs.TQC.2025.1

Related Version *Full Version:* <https://arxiv.org/abs/2504.03620v1>

Funding Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525. This work is supported by a collaboration between the US DOE and other Agencies. BH and JY acknowledge this work was supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Accelerated Research in Quantum Computing, Fundamental Algorithmic Research for Quantum Utility, with support also acknowledged from Fundamental Algorithm Research for Quantum Computing.

Ronak Ramachandran: Supported by the NSF AI Institute for Foundations of Machine Learning (IFML).

Justin Yirka: This material is based upon work supported by the U.S. Department of Energy, Office of Science, National Quantum Information Science Research Centers, Quantum Systems Accelerator.

Acknowledgements We especially thank John Kallaugh for helpful conversations and mentorship. RR and JY thank Scott Aaronson for stimulating discussions. JY thanks Bill Fefferman for suggesting his conjecture on PERMUTATION INVERSION.



© Blake Holman, Ronak Ramachandran, and Justin Yirka;
licensed under Creative Commons License CC-BY 4.0

20th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2025).

Editor: Bill Fefferman; Article No. 1; pp. 1:1–1:18



Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Quantum algorithms are typically developed and characterized in terms of query complexity. The strongest promises of quantum advantage over classical computation come from unconditional separations proved in terms of black-box queries, including Shor’s period-finding algorithm and Grover’s search algorithm. Understanding the nuances of the query model is therefore essential for advancing quantum algorithm design and sculpting quantum advantages.

Given an arbitrary Boolean function f , the standard query model in quantum computation is defined by XOR oracles S_f , also known as “standard oracles”, which map basis states $|x\rangle|y\rangle$ to $|x\rangle|y \oplus f(x)\rangle$. Other common models, such as phase oracles, are known to be equivalent. The use of XOR oracles goes back to the early days of quantum computation [23, 20, 21, 18, 17] and even reversible computation [14, 15, 36, 16]. XOR oracles embed potentially irreversible functions in a reversible way, ensuring that all queries are unitary. This enables quantum query complexity to encompass arbitrary Boolean functions and offers a standard input-output format for using one algorithm as a sub-routine in another.

Other oracle models for quantum computation have been studied, but most abandon unitarity [37, 26, 41, 30, 27, 39, 32, 33] or provide query access to quantum functions with no analog in classical query complexity, e.g. general unitaries [3, 5].

When querying a permutation, there is another natural oracle model: an *in-place* oracle P_f which maps $|x\rangle$ to $|f(x)\rangle$. These oracles have been called in-place [22, 9], erasing [1, 2], and minimal [28, 8].¹ Just like XOR oracles, in-place oracles can be directly studied and compared in both quantum and classical computation.

In-place oracles were first studied in the quantum setting by Kashefi, Kent, Vedral, and Banaszek [28]. They showed several results comparing XOR oracles and in-place oracles, including a proof that $\Theta(\sqrt{N})$ queries to an XOR oracle are required to simulate an in-place query to the same permutation. Around the same time, Aaronson [1] proved that SET COMPARISON, an approximate version of the COLLISION problem, requires an exponential number of XOR queries but only a constant number of in-place queries.

These oracles relate to multiple topics in quantum algorithms and complexity theory. Aaronson’s lower bound for the collision problem [1] was partially inspired by the desire to separate the in-place and XOR query models. [28] observed that a constant number of in-place queries is sufficient to solve RIGID GRAPH ISOMORPHISM, a necessary subcase for solving general GRAPH ISOMORPHISM. An identical protocol was later generalized to define the concept of QSAMPLING, which is sufficient to solve SZK, by Aharonov and Ta-Shma [4]. These ideas inspired pursuing lower bounds on the INDEX ERASURE problem [40, 7, 31], ruling out potential algorithms for GRAPH ISOMORPHISM using XOR oracles. Fefferman and Kimmel [22] showed an oracle separation of QMA and QCMA relative to randomized in-place oracles. Also, the expressive power of in-place oracles relates to the conjectured existence of one-way permutations [15, p. 926]. Additionally, because in-place oracles are not self-inverse, they offer a setting in which to study computation with inverse-free gate sets [19].

In-place oracles outperform XOR oracles in every established separation between the two query models, but it is conjectured that the oracles are incomparable, each better-suited for certain tasks. Aaronson [2] raised proving such a separation as an open problem. Fefferman

¹ Unfortunately, “permutation oracle” has been used to refer to any oracle which embeds a permutation. Following a suggestion by John Kallaugher, we have found it convenient in conversation to refer to “xoracles” and “smoracles” (for “small oracles”).

and Kimmel [22] conjectured that inverting a permutation over N elements, a task which requires only $O(\sqrt{N})$ queries to an XOR oracle, requires $\Omega(N)$ queries to an in-place oracle. PERMUTATION INVERSION is formally as hard as unstructured search [34], so this conjecture effectively predicts that the speedup of Grover’s algorithm [25] is impossible with an in-place oracle.

Results

We refute the conjecture of [22] by designing a new quantum algorithm that solves PERMUTATION INVERSION with $O(\sqrt{N})$ queries to an in-place oracle, recovering the same speedup as Grover’s search algorithm.

We additionally apply this algorithm to tightly characterize the ability of XOR and in-place oracles to simulate each other. Then, we change focus and make progress towards showing the desired separation. We introduced a subspace-conversion problem that requires 1 XOR query and exponentially-many in-place queries.

Finally, we propose a candidate decision problem that can be solved with $O(\sqrt{N})$ queries to an XOR oracle and that we conjecture requires $\Omega(N)$ queries to an in-place oracle. We then apply recent advances in the quantum adversary bound to define a new class of adversary matrices which must be used if such a decision-problem separation exists.

1.1 Quantum Search

Unstructured search, famously solved by Grover’s algorithm with $O(\sqrt{N})$ queries to an XOR oracle, is one of the most well-studied problems in quantum query complexity. The first non-trivial quantum lower bound was for unstructured search [17]. Later work modifying the query model, for instance by introducing noise or faults into queries, focused on unstructured search [37, 41, 30, 27, 39, 5].

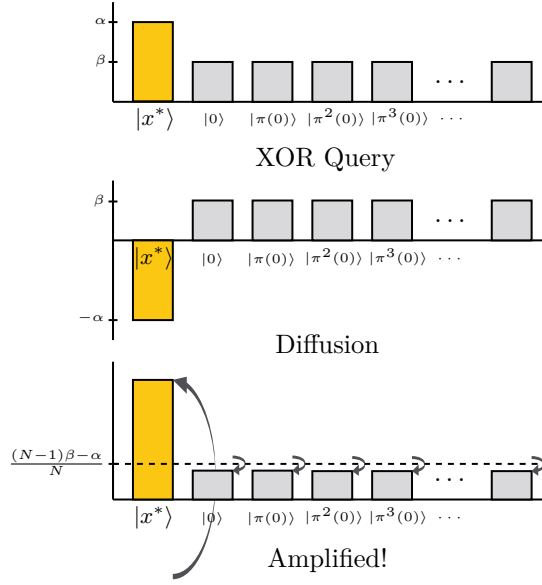
In-place oracles are only defined for bijections (see Section 2). Restricted to permutations, the unstructured search problem is equivalent to PERMUTATION INVERSION [34].²

► **Definition 1.** *Given query access to a permutation π on $[N] = \{0, \dots, N-1\}$, the PERMUTATION INVERSION problem is to output $\pi^{-1}(0)$.*

The choice to invert 0 can of course be replaced with any element. It is also straightforward to define a related decision problem, for example, deciding if $\pi^{-1}(0)$ is odd or even.

Like general unstructured search, PERMUTATION INVERSION has been a frequent target for new lower bound techniques. It can be solved with $O(\sqrt{N})$ queries to an XOR oracle using Grover’s algorithm. Ambainis [6] applied his new quantum adversary method to show that $\Omega(\sqrt{N})$ queries to an XOR oracle are in fact required to solve the problem. Nayak [34] gave an alternative proof by showing the problem is as hard as general unstructured search. Rosmanis [38] also reproduced this tight lower bound using the compressed oracle technique on random permutations. As for in-place oracles, [22] proved that $\Omega(\sqrt{N})$ in-place queries are needed to solve PERMUTATION INVERSION. Belovs and Yolcu [13] later applied their advancements on the quantum adversary method to reprove the same lower bound. We add to this sequence of work, studying PERMUTATION INVERSION in Section 3 to give the following result.

² The reductions between PERMUTATION INVERSION and unstructured search are entirely classical. So the reductions hold using either XOR oracles or in-place oracles, although some quantum garbage registers may differ.



■ **Figure 1** (Color) Illustration of how one iteration of Grover’s search algorithm amplifies $|x^* \rangle := |\pi^{-1}(0)\rangle$. Amplitudes are ordered according to the permutation in order to match Figure 3 later.

► **Theorem 2.** *For a permutation π on $[N]$, PERMUTATION INVERSION can be solved with $O(\sqrt{N})$ in-place queries to π .*

Thus, we refute the conjecture that $\Omega(N)$ in-place queries are required, and we show the $\Omega(\sqrt{N})$ lower bound [22, 12] is tight.

Grover’s Algorithm

Before we sketch our algorithm, we first recall Grover’s algorithm for unstructured search [25] in the context of PERMUTATION INVERSION. Grover’s algorithm repeatedly alternates between using XOR queries to negate the amplitude of $|\pi^{-1}(0)\rangle$ and using the “Grover Diffusion operator” to reflect all amplitudes about the average, steadily amplifying $|\pi^{-1}(0)\rangle$ on every iteration. In other words, the algorithm alternates between the oracle-dependent reflection $I - 2|\pi^{-1}(0)\rangle\langle\pi^{-1}(0)|$ and the diffusion reflection

$$D = I - 2|s\rangle\langle s|, \quad (1)$$

where $|s\rangle$ is the uniform superposition $\frac{1}{\sqrt{N}} \sum |i\rangle$. This is illustrated in Figure 1.

In-place oracles seem at odds with oracle-dependent reflections, since reflections – like XOR queries – are self-inverse, but inverting an in-place query is equivalent to inverting the underlying permutation, which would solve PERMUTATION INVERSION. With this in mind, it would be natural to conjecture, as [22] did, that no Grover-style speedup is possible using in-place oracles.

A New Algorithm

Let $x^* := \pi^{-1}(0)$ be the “marked item” to be found. Our algorithm starts with an equal superposition over $[N]$ along with an ancilla register and a “flag” qubit: $\frac{1}{\sqrt{N}} \sum |i\rangle |0^n\rangle |0\rangle$. The algorithm repeatedly iterates over steps *Mark*, *Shift*, and *Diffuse the Difference*. The intuition behind these steps is as follows.

- *Mark*: For every basis state $i \in [N]$, make a copy of i and query the oracle. Then, conditioned on the output of $\pi(i)$ being 0, flip the flag qubit from $|0\rangle$ to $|1\rangle$.
(The *Mark* step cannot be used to implement Grover's algorithm as usual because the query answer remains in the ancilla register, as garbage, until the next step.)
- *Shift*: In the $|1\rangle$ -flagged branch, all amplitude is concentrated on $|x^*\rangle$, while in the $|0\rangle$ -flagged branch, the amplitude is spread evenly over all basis states except $|x^*\rangle$.
In only the $|0\rangle$ -flagged branch of the superposition, query the oracle to shift the amplitude of each basis state forward according to π (perform a controlled in-place query to π). This shifts amplitude from $|i\rangle$ onto $|\pi(i)\rangle$, and in particular, from $|\pi^{-1}(x^*)\rangle$ onto $|x^*\rangle$.
- *Diffuse the Difference*: The two branches are now such that if they are interfered to produce two branches, one branch which adds amplitudes and another branch which subtracts amplitudes, then the amplitude on $|x^*\rangle$ would be above average in the former branch and below average in the latter branch.
Perform the standard Grover diffusion operator (Equation (1)) controlled on the flag qubit being the $|-\rangle$ state, which reflects the “difference branch” about its average amplitude. This results in the amplitude on $|x^*\rangle$ being similarly amplified in both branches. In fact, we find the branches are inverse-exponentially close to each other, and that after the t -th iteration, the overall state is effectively

$$|\psi_t\rangle = \left(\alpha_t |x^*\rangle + \sum_{i \in [N] \setminus \{x^*\}} \beta_t |i\rangle \right) |0^n\rangle |0\rangle,$$

where α_t increases by approximately $1/\sqrt{N}$ each iteration.

These steps are repeated $O(\sqrt{N})$ times to increase the amplitude on $|x^*\rangle$ until there is a constant probability of measuring it. Each iteration uses a constant number of in-place queries, so the overall query complexity is $O(\sqrt{N})$. For more intuition, see a circuit diagram in Figure 2 and an illustration in Figure 3 similar to Figure 1 above.

In Section 2.1, we give a construction for the controlled in-place query necessary for the *Shift* step of the algorithm. This construction differs significantly from the analogous construction for XOR oracles.

► **Lemma 3.** *There exists a unitary circuit making 1 in-place query to π which for all $x \in [N]$ maps*

$$|a\rangle |x\rangle |y\rangle \mapsto \begin{cases} |a\rangle |x\rangle |y\rangle & \text{when } a = 0 \\ |a\rangle |\pi(x)\rangle |y\rangle & \text{when } a = 1 \end{cases},$$

where y is the image under π of some fixed point, such as $y = \pi(0)$.

Note that although y depends on the oracle π , it is independent of the query x . So while y is garbage, it is effectively negligible. Because it is never entangled with the input register, the garbage can be safely measured and erased. See Section 2.1 for more details.

1.2 Simulating Other Oracles

In Section 4, we tightly characterize the ability of XOR and in-place oracles to simulate each other. We do so by applying our new algorithm to give new upper bounds and by developing a novel lower bound. The contents of Section 4 are deferred to the Full Version. For a permutation π on $[N]$, Grover's algorithm can be used to simulate an XOR query to π^{-1} , an in-place query to π^{-1} , or an in-place query to π using $O(\sqrt{N})$ XOR queries to π , and this complexity is known to be tight [28]. We show how to use our new algorithm

to perform the analogous simulations using $O(\sqrt{N})$ queries to an in-place oracle. The constructions are non-trivial due to the inability of in-place oracles to uncompute garbage. The simulations are approximate with inverse-exponential error due to the error in our algorithm for PERMUTATION INVERSION.

Next, we prove that our simulations are tight by giving matching lower bounds. Inspired by [28], we prove this by arguing that if few in-place queries could simulate an XOR query, then we could violate the lower bound of [22] for performing unstructured search.

► **Theorem 4.** *For a permutation π on $[N]$, $\Omega(\sqrt{N})$ in-place queries to π are necessary to approximately simulate an XOR query to π .*

Given that an XOR query to π can be implemented using 1 XOR query to π , Theorem 4 makes this the first task known to require more in-place queries than XOR queries. We improve on this in the next section.

We can summarize all upper and lower bounds above as follows.

► **Corollary 5** (Summary of relationships). *For a permutation π on $[N]$, $\Theta(\sqrt{N})$ queries to any one of an in-place oracle for π , an in-place oracle for π^{-1} , an XOR oracle for π , or an XOR oracle for π^{-1} are necessary and sufficient to approximately simulate any one of the others.*

1.3 A Subspace-Conversion Separation

In Section 5 we improve the unitary-implementation separation given in the previous section to a subspace-conversion separation. The contents of Section 5 are deferred to the Full Version.

INDEX ERASURE is the task of generating the state $\frac{1}{\sqrt{N}} \sum_{x \in [N]} |f(x)\rangle$ given queries to f . It was introduced by Shi [40] and formalized as a state-generation task by Ambainis, Magnin, Roetteler, and Roland [7]. As noted by [40], solving INDEX ERASURE would imply solutions to SET EQUALITY and GRAPH ISOMORPHISM. Similar work on QSampling [4] suggests many more applications. INDEX ERASURE requires $\Omega(\sqrt{N})$ XOR queries [7, 31] but just 1 in-place query, so the problem seems to capture key differences between the models.

We define the converse problem, FUNCTION ERASURE.

► **Definition 6.** *Given query access to a function f , FUNCTION ERASURE is the subspace-conversion problem of transforming any superposition of the form $\sum \alpha_x |x\rangle |f(x)\rangle$ to $\sum \alpha_x |x\rangle$.*

A state-conversion problem requires implementing an algorithm which, given an oracle to function f , maps an input $|\psi_f\rangle$ to output $|\phi_f\rangle$. A subspace-conversion problem simply generalizes this to multiple input-output pairs for each oracle function f . We discuss the details of unitary-implementation, subspace-conversion, and other types of problems in Section 5.

FUNCTION ERASURE can trivially be solved with 1 XOR query to f . Then by Corollary 5, $O(\sqrt{N})$ in-place queries are sufficient. Finally, we show how FUNCTION ERASURE and one additional in-place query are sufficient to simulate an XOR query. To avoid violating Theorem 4, this implies $\Omega(\sqrt{N})$ queries are necessary.

► **Theorem 7.** *For a permutation π on $[N]$, $\Theta(\sqrt{N})$ in-place queries to π are necessary and sufficient for FUNCTION ERASURE.*

Theorem 7 makes FUNCTION ERASURE the first coherent subspace-conversion problem known to require fewer XOR queries than in-place queries. This improves on the new unitary-implementation separation from the previous section.

1.4 Lower Bounds

The first works to study in-place oracles proved that there are problems which can be solved with asymptotically fewer queries to in-place oracles than to the corresponding XOR oracles [28, 1]. They left open the question of whether a separation could be shown in the opposite direction, making the two oracles formally incomparable, or whether in-place oracles are generically superior to XOR oracles. Our main result (Theorem 2) refutes one conjectured path towards constructing a problem for which XOR oracles are better than in-place oracles. Our study of FUNCTION ERASURE demonstrates the first problem which provably requires fewer queries to an XOR oracle than an in-place oracle, although it is a subspace-conversion problem instead of a decision problem. In Section 6, we consider the possibility of improving this to a decision-problem separation.

Conventional Lower Bound Techniques

Section 6.1 is deferred to the Full Version. There, we discuss how common quantum lower bound techniques, the polynomial method [10] and the unweighted adversary method [6], fail to prove the desired separation. We show that under these techniques, any lower bound on the number of in-place queries implies the same lower bound on the number of XOR queries, making these techniques unable to prove a separation where XOR oracles outperform in-place oracles.

A Candidate Decision Problem

In Section 6.2, we introduce a new problem, EMBEDDED PERMINV, which can be solved with $\Theta(\sqrt{N})$ queries to an XOR oracle and which we conjecture requires $\Omega(N)$ queries to an in-place oracle. As we discuss, the problem is designed to embed an injection from $[N^2]$ to $[N]$ into a bijection on $[N^2]$, which we believe circumvents algorithms using in-place oracles. The idea behind this problem builds on the “Simon’s problem with garbage” proposed by Aaronson [2].

Techniques for a Decision-Problem Separation

Finally, in Section 6.3, we briefly discuss the potential for more sophisticated lower bound methods to prove a decision-problem separation, including for our candidate EMBEDDED PERMINV. A full exposition is given in the Appendix of the full version of this article.

The recent extension of the quantum adversary method by Belovs and Yolcu [13] applies to arbitrary linear transformations, including in-place oracles. The adversary bound is an optimization problem over *adversary matrices* such that the optimal value equals the quantum query complexity for a given problem. Of course, the difficulty with the adversary method is to design a “good” adversary matrix exhibiting a tight bound.

We introduce a special class of feasible solutions which we call *extended adversary matrices*. We show, with some technical caveats, that there exists an XOR query advantage over in-place oracles for a decision problem if and only if it is witnessed by extended adversary matrices. Then, for our candidate problem EMBEDDED PERMINV, we are able to remove these caveats and state that if our conjectured separation is true, then it must be witnessed by extended adversary matrices.

1.5 Open Problems

A list of open problems is given in the full version of this article.

2 Quantum Oracles

As stated previously, the standard query model in quantum computation and classical reversible computation is the XOR oracle. Other common models, such as the phase oracle, are equivalent. For a function f , an XOR oracle S_f maps $|x\rangle|y\rangle \mapsto |x\rangle|y \oplus f(x)\rangle$, where \oplus denotes bitwise XOR with queries encoded in binary.

When querying an invertible function, there is another natural unitary query model.³ An in-place oracle P_π maps $|x\rangle \mapsto |\pi(x)\rangle$.

Here we list several basic identities given by [28].

1. Given query access to both π and π^{-1} , standard and in-place oracles are equivalent. More precisely, P_π can be simulated using 1 query to S_π and 1 query to either of $S_{\pi^{-1}}, P_{\pi^{-1}}$. Similarly, S_π can be simulated using 1 query to P_π and 1 query to either of $S_{\pi^{-1}}, P_{\pi^{-1}}$. So, the interesting case is when we can query π but cannot query its inverse.
2. XOR oracles are self-inverse, $S_\pi = (S_\pi)^\dagger$, but generally $(S_\pi)^\dagger \neq S_{\pi^{-1}}$. In contrast, generally $P_\pi \neq (P_\pi)^\dagger$ but it does hold that $(P_\pi)^\dagger = P_{\pi^{-1}}$.
3. $\Theta(\sqrt{N})$ queries to an XOR oracle S_π can be used to simulate a query to P_π . The upper bound is due to Grover's search algorithm. The lower bound follows by observing that a circuit for P_π querying S_π can be inverted to give a circuit for $P_{\pi^{-1}}$ querying $(S_\pi)^\dagger = S_\pi$, which would solve PERMUTATION INVERSION, which requires $\Omega(\sqrt{N})$ queries to S_π .

The XOR query model was motivated by two needs. First is the need to embed non-invertible functions in a reversible query. Second is that because XOR oracles are self-inverse, they enable uncomputing. An early criticism of reversible computation by Landauer [29] was that in order to maintain reversibility, a computation would need to retain intermediate work until the end, only deferring the cost of information erasure instead of avoiding it. To the contrary, Bennett showed that any circuit can efficiently be made into a reversible one that uncomputes any intermediate work and gives its original output in the form of an XOR query [14, 15]. Given a garbage-producing reversible circuit, first apply the circuit, then copy the desired output into a new register using XOR, and then apply the circuit in reverse, gate-by-gate, to uncompute all intermediate steps, leaving only the input and the copied output. Moreover, such a gate-by-gate reversal works when one algorithm is used as a black-box subroutine for another, since given a black-box following this XOR-model, it is self-inverse. So full algorithms, including subroutines, can indeed be reversed gate-by-gate. Besides these two reasons, XOR oracles simply appeared natural at the time quantum computing was formalized. As far as we are aware, in-place oracles have not been studied in the classical reversible computing literature. There have been just a few references to alternative classical reversible implementations of 1-to-1 functions [36, 16]. So quantum computation, which is based on reversible operations, later inherited the XOR model. At the same time, the ability to uncompute enabled quantum interference [23, 18]. Many early results also only involved binary functions, and other results were motivated more by ensuring quantum computers could implement tasks such as error-reduction and subroutines ($\text{BQP}^{\text{BQP}} = \text{BQP}$ [17]) rather than questioning the query model.

³ We restrict our study to bijections, and without loss of generality to permutations on $[N]$. A similar oracle which queries an injection would still be reversible, but it would be an isometry rather than a unitary. Our algorithm seems to require a bijection since it uses the oracle's previous outputs as its next inputs.

One more important feature of XOR oracles is that for a function f , the complexity of implementing S_f using reversible operations is at most a constant multiplicative factor more than for the general, irreversible circuit implementing f [14]. For in-place oracles, no construction is known for efficiently transforming an irreversible circuit for permutation π into a reversible circuit for P_π . In fact, the widely believed existence of one-way permutations implies that there exist permutations for which this is impossible. This is because given a reversible implementation of P_π , inverting the circuit gate-by-gate gives $(P_\pi)^\dagger = P_{\pi^{-1}}$ with exactly the same circuit size, whereas one-way permutations should have different complexities than their inverses. This may limit the practical instantiation of in-place oracles, although they may lead to useful insights in other ways.

2.1 Controlled In-Place Oracle

The proof of Lemma 3 is deferred to the Full Version.

3 Permutation Inversion

In this section, we prove our main result, that PERMUTATION INVERSION (Definition 1) can be solved with $\Theta(\sqrt{N})$ queries to an in-place oracle.

Proof of Theorem 2. The lower bound was proved by Fefferman and Kimmel [22] and later reproved by [13]. To prove the upper bound, we give an algorithm.

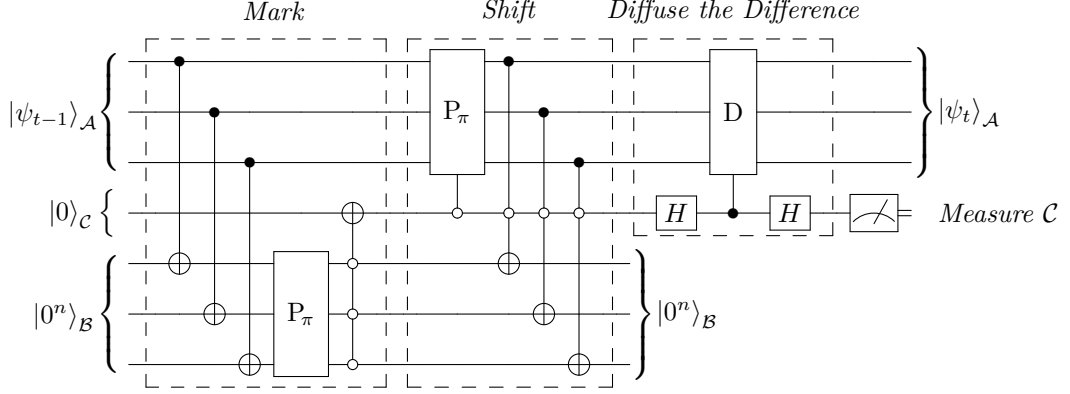
Algorithm. For convenience, we assume $N = 2^n$ and identify the integers $[N]$ by their binary representations in $\{0, 1\}^n$. We denote the target element $\pi^{-1}(0)$ by x^* .

First, query P_π once to check whether $\pi(0)$ is 0, and terminate early with answer 0 if it is. Otherwise, initialize three registers to the state $|\psi_0\rangle := \frac{1}{\sqrt{N}} \sum_{i=1}^N |i\rangle_{\mathcal{A}} |0^n\rangle_{\mathcal{B}} |0\rangle_{\mathcal{C}}$, where \mathcal{A} and \mathcal{B} are each $n = \log N$ qubits and \mathcal{C} is one qubit. Then, repeat the following steps $T = O(\sqrt{N})$ times:

- (1) *Mark*
 XOR register \mathcal{A} into \mathcal{B} , and apply P_π to \mathcal{B} .
 Controlled on \mathcal{B} being $|0^n\rangle$, apply NOT to \mathcal{C} , flagging the branch where \mathcal{A} contains x^* .
- (2) *Shift (and Clean Up)*
 Controlled on \mathcal{C} being $|0\rangle$, apply P_π to \mathcal{A} .
 Controlled on \mathcal{C} being $|0\rangle$, XOR \mathcal{A} into \mathcal{B} , resetting \mathcal{B} to $|0^n\rangle$.
- (3) *Diffuse the Difference*
 Controlled on \mathcal{C} being $|-\rangle$, apply the diffusion operator to \mathcal{A} .
 The diffusion operator $D := 2H^{\otimes n} |0^n\rangle\langle 0^n| H^{\otimes n} - I$ is the same used in Grover's algorithm [25], equivalent to a reflection about the uniform superposition.
- (4) *Optional: Measure*
 Measure \mathcal{C} . If $|1\rangle$ is observed then abort and report failure.

Finally, measure register \mathcal{A} and output the result. See Figure 2 for a circuit diagram of one iteration of the algorithm and Figure 3 for an illustration of the effect.

Below, we will find that each *Measure* step aborts with probability $1/N$. So, these intermediate measurements could be omitted and the qubit reused as it is, and the quantum union bound [24, 35] implies the overall success probability would decrease by at most $\sqrt{T/N} = O(N^{-1/4})$. For now, we include the optional *Measure* step to simplify the analysis.



■ **Figure 2** One iteration of our PERMUTATION INVERSION algorithm. D is the standard diffusion operator. \bullet denotes an operation controlled on $|1\rangle$ and \circ denotes an operation controlled on $|0\rangle$.

Analysis. Now we prove that our algorithm succeeds with high probability.

We use $|\psi_t\rangle$ to denote the state after t iterations. We will show by induction that after each iteration, if the algorithm did not terminate early, then the state is of the form

$$|\psi_t\rangle = \left(\alpha_t |x^*\rangle + \sum_{i \in [N] \setminus \{x^*\}} \beta_t |i\rangle \right)_{\mathcal{A}} \otimes |0^n\rangle_{\mathcal{B}} |0\rangle_{\mathcal{C}} \quad (2)$$

for some real values α_t, β_t . In particular, all $|i\rangle$ for $i \neq x^*$ share the same amplitude. The transformation from $|\psi_{t-1}\rangle$ to $|\psi_t\rangle$ is illustrated in Figure 3.

By construction, the initial state $|\psi_0\rangle$ is the uniform superposition, with $\alpha_0 = \beta_0 = \frac{1}{\sqrt{N}}$.

Next, the t -th iteration begins with the state

$$|\psi_{t-1}\rangle = \left(\alpha_{t-1} |x^*\rangle + \sum_{i \in [N] \setminus \{x^*\}} \beta_{t-1} |i\rangle \right) |0^n\rangle |0\rangle.$$

For ease of notation, we will drop the subscripts so that α, β implicitly refer to $\alpha_{t-1}, \beta_{t-1}$. After the *Mark* step, the state will be

$$|\psi'_{t-1}\rangle = \alpha |x^*\rangle |0^n\rangle |1\rangle + \sum_{i \in [N] \setminus \{x^*\}} \beta |i\rangle |\pi(i)\rangle |0\rangle.$$

After the *Shift* (and *Clean Up*) step, the state will be

$$\begin{aligned} |\psi''_{t-1}\rangle &= \alpha |x^*\rangle |0^n\rangle |1\rangle + \sum_{i \in [N] \setminus \{x^*\}} \beta |\pi(i)\rangle |0^n\rangle |0\rangle \\ &= \alpha |x^*\rangle |0^n\rangle |1\rangle + \sum_{i \in [N] \setminus \{0\}} \beta |i\rangle |0^n\rangle |0\rangle. \end{aligned}$$

As the name suggests, this step shifts amplitudes within the summation off of $|0\rangle$ and onto $|x^*\rangle$.

Next, to prepare for the *Diffuse the Difference* step, we rewrite register \mathcal{C} in the Hadamard basis. The state is equivalent to

$$\begin{aligned} |\psi''_{t-1}\rangle = & \frac{1}{\sqrt{2}} \left[(\beta + \alpha) |x^*\rangle + \sum_{i \in [N] \setminus \{0, x^*\}} \beta |i\rangle \right] |0^n\rangle |+\rangle \\ & + \frac{1}{\sqrt{2}} \left[(\beta - \alpha) |x^*\rangle + \sum_{i \in [N] \setminus \{0, x^*\}} \beta |i\rangle \right] |0^n\rangle |-\rangle. \end{aligned}$$

Next, the *Diffuse the Difference* step applies the diffusion operator D controlled on \mathcal{C} being $|-\rangle$. The diffusion operator can be viewed as reflecting every amplitude about the average amplitude. This results in

$$\begin{aligned} |\psi'''_{t-1}\rangle = & \frac{1}{\sqrt{2}} \left[(\beta + \alpha) |x^*\rangle + \sum_{i \in [N] \setminus \{0, x^*\}} \beta |i\rangle \right] |0^n\rangle |+\rangle \\ & + \frac{1}{\sqrt{2}} \left[\left(\beta + \alpha - \frac{2(\beta + \alpha)}{N} \right) |x^*\rangle + \left(2\beta - \frac{2(\beta + \alpha)}{N} \right) |0\rangle \right. \\ & \quad \left. + \sum_{i \in [N] \setminus \{0, x^*\}} \left(\beta - \frac{2(\beta + \alpha)}{N} \right) |i\rangle \right] |0^n\rangle |-\rangle. \end{aligned}$$

Returning register \mathcal{C} to the standard basis, we see

$$\begin{aligned} |\psi'''_{t-1}\rangle = & \left[\left(\beta + \alpha - \frac{\beta + \alpha}{N} \right) |x^*\rangle + \sum_{i \in [N] \setminus \{x^*\}} \left(\beta - \frac{\beta + \alpha}{N} \right) |i\rangle \right] |0^n\rangle |0\rangle \\ & + \left[\frac{\beta + \alpha}{N} |x^*\rangle - \left(\beta - \frac{\beta + \alpha}{N} \right) |0\rangle + \sum_{i \in [N] \setminus \{0, x^*\}} \frac{\beta + \alpha}{N} |i\rangle \right] |0^n\rangle |1\rangle. \end{aligned}$$

The amplitude on $|x^*\rangle$ is now larger than the original amplitude α .

Finally, for the sake of analysis, we choose to measure \mathcal{C} and abort if $|1\rangle$ is observed. We will handle the failure case later. For now, we postselect on having observed $|0\rangle$. This results in the final (normalized) state

$$\begin{aligned} |\psi_t\rangle = & \sqrt{\frac{N}{N-1}} \left[\left(\beta + \alpha - \frac{\beta + \alpha}{N} \right) |x^*\rangle + \sum_{i \in [N] \setminus \{x^*\}} \left(\beta - \frac{\beta + \alpha}{N} \right) |i\rangle \right] |0^n\rangle |0\rangle \\ = & \left[\sqrt{\frac{N-1}{N}} (\beta + \alpha) |x^*\rangle + \sum_{i \in [N] \setminus \{x^*\}} \left(\sqrt{\frac{N-1}{N}} \beta - \frac{1}{\sqrt{N}\sqrt{N-1}} \alpha \right) |i\rangle \right] |0^n\rangle |0\rangle. \end{aligned}$$

at the end of the t -th iteration. This state has the form we claimed, with

$$\alpha_t = \sqrt{\frac{N-1}{N}} (\beta_{t-1} + \alpha_{t-1}) \quad \text{and} \quad \beta_t = \sqrt{\frac{N-1}{N}} \beta_{t-1} - \frac{1}{\sqrt{N}\sqrt{N-1}} \alpha_{t-1},$$

concluding our induction.

The above recurrence lets us write a closed form for α_t and β_t :

$$\begin{bmatrix} \alpha_t \\ \beta_t \end{bmatrix} = \begin{bmatrix} \sqrt{\frac{N-1}{N}} & \sqrt{\frac{N-1}{N}} \\ \frac{-1}{\sqrt{N}\sqrt{N-1}} & \sqrt{\frac{N-1}{N}} \end{bmatrix} \begin{bmatrix} \alpha_{t-1} \\ \beta_{t-1} \end{bmatrix} = \begin{bmatrix} \sqrt{\frac{N-1}{N}} & \sqrt{\frac{N-1}{N}} \\ \frac{-1}{\sqrt{N}\sqrt{N-1}} & \sqrt{\frac{N-1}{N}} \end{bmatrix}^t \begin{bmatrix} \frac{1}{\sqrt{N}} \\ \frac{1}{\sqrt{N}} \end{bmatrix}.$$

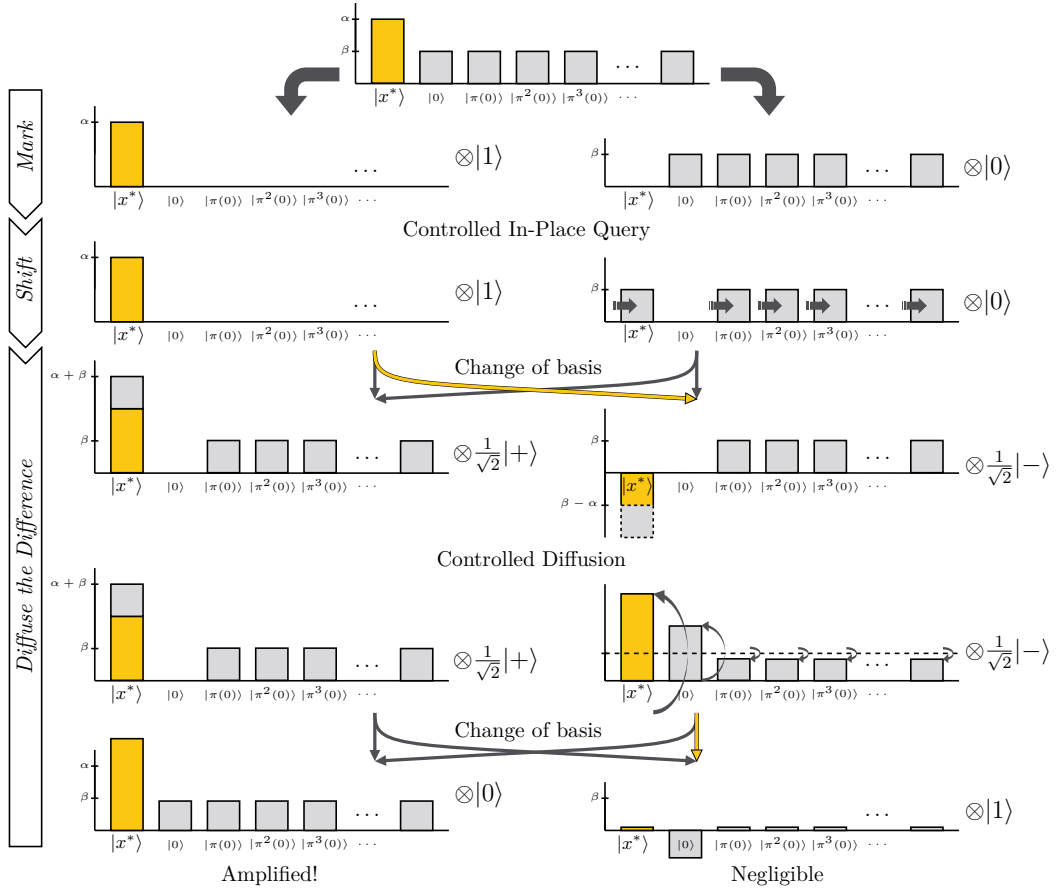


Figure 3 (Color) Illustration of how amplitudes change in each iteration of the algorithm. Register \mathcal{B} is left implicit (note it is unentangled with \mathcal{A} and \mathcal{C} by the end of the *Shift* step). Each iteration begins with the nearly uniform superposition from Equation (2). The *Mark* step queries π and creates a marked branch and an unmarked branch, illustrated in two columns. The *Shift* step makes a query in only the unmarked branch, shifting amplitude onto $|x^*\rangle$. The *Diffuse the Difference* step is controlled on $|-\rangle$, so we first rewrite the basis of \mathcal{C} , rearranging amplitudes accordingly. Black and yellow arrows indicate positive and negative contributions. The diffusion operator reflects all amplitudes about their mean. A final change of basis leaves a state almost entirely entangled with $|0\rangle$ and with increased amplitude on x^* .

For a diagonalizable matrix $M = ADA^{-1}$, we know $M^t = AD^tA^{-1}$, so we can diagonalize the above matrix to find

$$\alpha_t = \frac{1}{\sqrt{N^{t+1}}} \frac{1}{2i} \left[\left(\sqrt{N-1} + i \right)^{t+1} - \left(\sqrt{N-1} - i \right)^{t+1} \right].$$

Rewriting the expression in polar form, this is equivalent to

$$\alpha_t = \frac{1}{\sqrt{N^{t+1}}} \frac{1}{2i} \left[\sqrt{N^{t+1}} e^{i(t+1)\theta} - \sqrt{N^{t+1}} e^{-i(t+1)\theta} \right] \quad \text{for } \theta = \arctan \left(\frac{1}{\sqrt{N-1}} \right).$$

Finally, the identity $\frac{z-\bar{z}}{2i} = \text{Im}(z) = \sin(\phi)$ for $z = e^{i\phi}$ yields

$$\alpha_t = \sin \left[(t+1) \arctan \left(\frac{1}{\sqrt{N-1}} \right) \right].$$

We want to find the value of t that maximizes α_t . Setting

$$t^* = \frac{\frac{\pi}{2}}{\arctan \left(\frac{1}{\sqrt{N-1}} \right)} - 1$$

achieves $\alpha_{t^*} = 1$. The series expansion of this formula shows t^* is asymptotically $\frac{\pi}{2}\sqrt{N} + O(1)$, as desired. However, t must be an integer, so we set the number of iterations to $T = \lfloor t^* \rfloor$. Observe that $\sin(x)$ increases as x approaches $\frac{\pi}{2}$, so it is sufficient to lower bound $\alpha_{t^*-1} \leq \alpha_T$. Substituting and then simplifying, we find

$$\alpha_{t^*-1} = \sin \left[\frac{\pi}{2} - \arctan \left(\frac{1}{\sqrt{N-1}} \right) \right] = \cos \left[\arctan \left(\frac{1}{\sqrt{N-1}} \right) \right] = \sqrt{1 - \frac{1}{N}}.$$

So, given the algorithm never terminates early, it outputs $|x^*\rangle$ with probability at least $|\alpha_T|^2 \geq 1 - 1/N$.

Finally, we handle the possibility of the algorithm terminating early. In each iteration, given $|\psi''_{t-1}\rangle$, the probability of measuring $|1\rangle$ is $(\alpha^2 + (N-1)\beta^2)/N = 1/N$. Therefore, in $T = O(\sqrt{N})$ iterations, the probability of aborting is at most a negligible $T/N = O(1/\sqrt{N})$.

Overall, we have that our algorithm aborts with probability at most $O(1/\sqrt{N})$, while if it does not abort, then it fails to find $|x^*\rangle$ with probability at most $O(1/N)$. We conclude that with $T = \frac{\pi}{2}\sqrt{N} + O(1)$ queries to P_π , we can solve PERMUTATION INVERSION with probability $1 - O(1/\sqrt{N})$. ◀

4 Simulating Other Oracles

This section is omitted due to space constraints and appears in the Full Version.

5 A Subspace-Conversion Separation

This section is omitted due to space constraints and appears in the Full Version.

6 Lower Bounds

In this section, we consider avenues for improving our separations with XOR oracles outperforming in-place oracles to demonstrate a decision-problem separation.

In Section 6.1, we explain the limitations of conventional lower bound techniques for showing that fewer XOR queries are required for a task than in-place queries. In Section 6.2, we introduce a candidate decision problem which we conjecture exhibits such a separation. Then in Section 6.3, we explore recently developed tools for proving lower bounds for arbitrary oracles, including in-place oracles. We develop exact conditions for a decision problem to exhibit a separation. Further details are given in the Appendix of the Full Version.

6.1 Conventional Lower Bound Techniques

In pursuit of proving a decision-problem separation with XOR oracles outperforming in-place oracles, we begin by considering standard tools. Two techniques have dominated quantum query complexity: the polynomial method and the (basic) adversary method. See the thesis of Belovs [11, Chap. 3] for an excellent survey of these tools. Unfortunately, we find that these two methods are insufficient for proving the desired separation.

The remainder of this section is omitted and appears in the full version of this article.

6.2 A Candidate Decision Problem

In this section, we introduce a decision problem called EMBEDDED PERMINV which can be solved with $\Theta(\sqrt{N})$ XOR queries and which we conjecture requires $\Omega(N)$ in-place queries.

Earlier, we showed that in-place query algorithms can achieve the same query complexity as XOR oracles for PERMUTATION INVERSION. As noted in Footnote 3, our algorithm appears to crucially rely on the fact that it is inverting a permutation rather than an injection. The algorithm uses the image of the permutation from one iteration as the input in the next. Now that our goal is to find a problem for which in-place queries are less useful than XOR queries, we leverage this limitation. (Below, S_i is the symmetric group of degree i .)

► **Definition 8** (Promised Permutation Inversion). *Given query access to a permutation f on $[N^2] = \{0, \dots, N^2 - 1\}$, the decision problem EMBEDDED PERMINV : $S_{N^2} \rightarrow \{0, 1\}$ is defined by*

$$\text{EMBEDDED PERMINV}(f) = \begin{cases} 1 & \text{if } f^{-1}(0) \leq N, \text{ and} \\ 0 & \text{otherwise.} \end{cases}$$

In effect, this problem embeds an injection from $[N] \rightarrow [N^2]$ into a bijection on $[N^2]$, with the promise that an algorithm only needs to search over $[N]$. This problem is inspired by a candidate proposed by Aaronson [2] which was a version of Simon's problem with garbage appended to each query. When querying an XOR oracle, it is easy to copy the desired part of any answer and then uncompute with an additional query, allowing the garbage to be ignored. In contrast, it is unclear how to uncompute or erase the garbage with an in-place oracle, which would prevent interference. Here, instead of Simon's problem we focus on PERMUTATION INVERSION, and we formalize the idea of appending garbage as embedding an injection into a bijection.

► **Lemma 9.** EMBEDDED PERMINV can be decided with at most $\Theta(\sqrt{N})$ XOR queries.

The proof of Lemma 9 is deferred to the Full Version.

It is unclear how to solve EMBEDDED PERMINV as efficiently as the above algorithm when using in-place queries. Simply querying $\sum |x\rangle \mapsto \sum |f(x)\rangle$ would be useless. One can instead consider algorithms that involve mapping $|x\rangle |0\rangle \mapsto |x\rangle |f(x)\rangle$. Any such query $x \in [N]$ will lead to an unknown element $f(x) \in [N^2]$. Since $f(x)$ may not be in $[N]$, this (a priori) unknown element seems useless for finding the pre-image $f^{-1}(0) \in [N]$. Moreover,

in-place oracles are not self-inverse and do not readily allow uncomputing queries. So the image is both useless to keep around and not readily uncomputable using in-place queries. We conjecture this task as a candidate for which XOR oracles outperform in-place oracles.

► **Conjecture 10.** *EMBEDDED PERMINV requires at least $\Omega(N)$ queries to an in-place oracle.*

Note that even a classical algorithm can solve the problem with N queries by simply querying every element of $[N]$. Also note that while an exponential query separation is possible for Simon’s with garbage, the largest separation possible with EMBEDDED PERMINV is polynomial. We hope that the structure of the problem makes a separation more tractable.

6.3 Sketch of Techniques for a Decision Problem Separation

Here, we briefly explore applying a recent version of the quantum adversary bound to prove the desired decision-problem separation. A full exposition is given in the Appendix of the Full Version.

Quantum query complexity can be characterized by the adversary method. This method has been used to develop several different adversary bounds or adversary theorems in different contexts. For example, prior work derived adversary bounds in the XOR oracle model. In general, an adversary bound for a decision problem $\phi : D \rightarrow \{0, 1\}$ is an optimization problem such that the optimum is a lower bound on the query complexity. Belovs and Yolcu [13] recently developed a new version of the adversary bound that applies to arbitrary linear transformations. In fact, [13, Section 10] specifically observed this includes in-place oracles in addition to XOR oracles. Moreover, the bound of [13] is tight, meaning the optimum value of the optimization problem corresponds to the optimum query complexity and vice versa.

One caveat is that the lower bound of [13] is for *Las Vegas* query complexity, a quantum analog of the expected number of queries needed for a zero-error algorithm, in contrast to the usual notion of bounded-error complexity. So, our results in this section are primarily focused on Las Vegas complexity. But, for the special case of EMBEDDED PERMINV, we are able to extend the analysis to bounded-error complexity.

The optimization problem in the adversary bound developed by [13] is specifically an optimization over *adversary matrices* Γ . The optimal choice of adversary matrix then corresponds to the optimal query algorithm. In other versions of the adversary method, adversary matrices have been restricted to nonnegative values (the positive weight method) or to general real numbers (the negative weights method). For a decision problem $\phi : D \rightarrow \{0, 1\}$, previous methods have nearly always restricted Γ such that an entry $\Gamma[f, g]$ indexed by problem instances f and g satisfies that if $\phi(f) = \phi(g)$, then $\Gamma[f, g] = 0$. But, one feature of this new version of the adversary method is that it removes that restriction: we are free to assign nonzero values to *all* entries of Γ .

We call these matrices, with nonzero entries corresponding to problem instances with the same answer, *extended* adversary matrices. We show that, just as negative-weight adversary matrices are necessary to prove tight lower bounds for certain problems, these “extended” adversary matrices are necessary to prove the desired decision-problem separation with XOR oracles outperforming in-place oracles. In other words, if we use only tools from the negative-weight adversary bound to construct adversary matrices Γ , then we cannot prove our desired query separation.

► **Theorem (Informal statement).** *For a decision problem $\phi : D \rightarrow \{0, 1\}$, the Las Vegas query complexity using XOR oracles is asymptotically less than the Las Vegas query complexity using in-place oracles if and only if optimizing over extended adversary matrices witnesses it.*

Again, the above statement is in terms of Las Vegas complexity instead of the more typical bounded-error complexity. But, for our candidate problem EMBEDDED PERMINV introduced in the previous section, we are able to extend the statement to bounded-error complexity

► **Theorem** (Informal statement). *For the decision problem EMBEDDED PERMINV, the bounded-error query complexity using XOR oracles is asymptotically less than the bounded-error query complexity using in-place oracles if and only if optimizing over extended adversary matrices witnesses it.*

See the Appendix of the Full Version for details. In sum, we considerably narrow down what techniques could possibly prove an $\Omega(N)$ lower bound on EMBEDDED PERMINV. Although we rule out the polynomial and unweighted adversary methods, the new adversary method of Belovs and Yolcu [13] is tight, so that if such a lower bound is possible, then it is witnessed by adversary matrices. By the above theorem, we see that any lower bound stronger than $\Omega(\sqrt{N})$ must use this new class of *extended* adversary matrices.

References

- 1 Scott Aaronson. Quantum lower bound for the collision problem. In *Proceedings of the Thirty-Fourth Annual ACM Symposium on Theory of Computing*, pages 635–642. ACM, 2002. doi:10.1145/509907.509999.
- 2 Scott Aaronson. Open problems related to quantum query complexity. *ACM Transactions on Quantum Computing*, 2(4), 2021. doi:10.1145/3488559.
- 3 Scott Aaronson and Greg Kuperberg. Quantum versus classical proofs and advice. *Theory of Computing*, 3(7):129–157, 2007. doi:10.4086/toc.2007.v003a007.
- 4 Dorit Aharonov and Amnon Ta-Shma. Adiabatic quantum state generation and statistical zero knowledge. In *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing*, pages 20–29. ACM, 2003. doi:10.1145/780542.780546.
- 5 Gorjan Alagic, Chen Bai, Alexander Poremba, and Kaiyan Shi. On the two-sided permutation inversion problem. *IACR Communications in Cryptology*, 1(1), 2024. doi:10.62056/a0qj89n4e.
- 6 Andris Ambainis. Quantum lower bounds by quantum arguments. *Journal of Computer and System Sciences*, 4(64):750–767, 2002. doi:10.1006/jcss.2002.1826.
- 7 Andris Ambainis, Loïck Magnin, Martin Roetteler, and Jérémie Roland. Symmetry-assisted adversaries for quantum state generation. In *2011 IEEE 26th Annual Conference on Computational Complexity*, pages 167–177, 2011. doi:10.1109/CCC.2011.24.
- 8 Alp Atici. Comparative computational strength of quantum oracles, 2004. arXiv:quant-ph/0312107v3.
- 9 Roozbeh Bassirian, Bill Fefferman, and Kunal Marwaha. On the power of nonstandard quantum oracles. In Omar Fawzi and Michael Walter, editors, *18th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2023)*, volume 266 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 11:1–11:25. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.TQC.2023.11.
- 10 Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald de Wolf. Quantum lower bounds by polynomials. *J. ACM*, 48(4):778–797, 2001. doi:10.1145/502090.502097.
- 11 Aleksandrs Belovs. *Applications of Adversary Method in Quantum Query Algorithms*. PhD thesis, University of Latvia, 2014. URL: <https://dspace.lu.lv/dspace/handle/7/4854>.
- 12 Aleksandrs Belovs. Variations on quantum adversary, 2015. arXiv:1504.06943v1.
- 13 Aleksandrs Belovs and Duyal Yolcu. One-way ticket to Las Vegas and the quantum adversary, 2023. arXiv:2301.02003v1.

- 14 Charles H. Bennett. Logical reversibility of computation. *IBM Journal of Research and Development*, 17(6):525–532, 1973. doi:10.1147/rd.176.0525.
- 15 Charles H. Bennett. The thermodynamics of computation—a review. *International Journal of Theoretical Physics*, 21:905–940, 1982. doi:10.1007/bf02084158.
- 16 Charles H. Bennett. Time/space trade-offs for reversible computation. *SIAM Journal on Computing*, 18(4):766–776, 1989. doi:10.1137/0218053.
- 17 Charles H. Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and weaknesses of quantum computing. *SIAM Journal on Computing*, 26(5):1510–1523, 1997. doi:10.1137/S0097539796300933.
- 18 Ethan Bernstein and Umesh Vazirani. Quantum complexity theory. *SIAM Journal on Computing*, 26(5):1411–1473, 1997. doi:10.1137/S0097539796300921.
- 19 Adam Bouland and Tudor Giurgica-Tiron. Efficient universal quantum compilation: An inverse-free Solovay-Kitaev algorithm, 2021. arXiv:2112.02040v1.
- 20 David Deutsch. Quantum theory, the Church–Turing principle and the universal quantum computer. *Proc. R. Soc. Lond. A*, 400(1818):97–117, 1985. doi:10.1098/rspa.1985.0070.
- 21 David Deutsch and Richard Jozsa. Rapid solution of problems by quantum computation. *Proc. R. Soc. Lond. A*, 439(1907):553–558, 1992. doi:10.1098/rspa.1992.0167.
- 22 Bill Fefferman and Shelby Kimmel. Quantum vs. classical proofs and subset verification. In Igor Potapov, Paul Spirakis, and James Worrell, editors, *43rd International Symposium on Mathematical Foundations of Computer Science (MFCS 2018)*, volume 117 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 22:1–22:23. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.MFCS.2018.22.
- 23 Richard P. Feynman. Quantum mechanical computers. *Foundations of Physics*, 16(6):507–531, 1986. doi:10.1007/BF01886518.
- 24 Jingliang Gao. Quantum union bounds for sequential projective measurements. *Phys. Rev. A*, 92(5):052331, 2015. doi:10.1103/PhysRevA.92.052331.
- 25 Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, pages 212–219. ACM, 1996. doi:10.1145/237814.237866.
- 26 Aram Harrow and David Rosenbaum. Uslessness for an oracle model with internal randomness. *Quantum Info. Comput.*, 14(7&8):608–624, 2013. doi:10.26421/QIC14.7–8–5.
- 27 Atsuya Hasegawa and François Le Gall. An optimal oracle separation of classical and quantum hybrid schemes. In Sang Won Bae and Heejin Park, editors, *33rd International Symposium on Algorithms and Computation (ISAAC 2022)*, volume 248 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 6:1–6:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.ISAAC.2022.6.
- 28 Elham Kashefi, Adrian Kent, Vlatko Vedral, and Konrad Banaszek. Comparison of quantum oracles. *Phys. Rev. A*, 65:050304, 2002. doi:10.1103/PhysRevA.65.050304.
- 29 R. Landauer. Irreversibility and heat generation in the computing process. *IBM Journal of Research and Development*, 5(3):183–191, 1961. doi:10.1147/rd.53.0183.
- 30 Cedric Yen-Yu Lin and Han-Hsuan Lin. Upper bounds on quantum query complexity inspired by the Elitzur–Vaidman bomb tester. *Theory of Computing*, 12(18):1–35, 2016. doi:10.4086/toc.2016.v012a018.
- 31 Nathan Lindzey and Ansis Rosmanis. A tight lower bound for non-coherent index erasure. In Thomas Vidick, editor, *11th Innovations in Theoretical Computer Science Conference (ITCS 2020)*, volume 151 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 59:1–59:37. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.ITCS.2020.59.
- 32 David Rasmussen Lolck, Małinska, and Manaswi Paraashar. Quantum advantage with a faulty oracle, 2024. arXiv:2411.04931v1.
- 33 Anand Natarajan and Chinmay Nirkhe. A distribution testing oracle separation between QMA and QCMA. *Quantum*, 8:1377, 2024. doi:10.22331/q-2024-06-17-1377.

- 34 Ashwin Nayak. Inverting a permutation is as hard as unordered search. *Theory of Computing*, 7(2):19–25, 2011. doi:10.4086/toc.2011.v007a002.
- 35 Ryan O’Donnell and Ramgopal Venkateswaran. The quantum union bound made easy. In *2022 Symposium on Simplicity in Algorithms (SOSA)*, pages 314–320. SIAM, 2022. doi:10.1137/1.9781611977066.25.
- 36 Asher Peres. Reversible logic and quantum computers. *Phys. Rev. A*, 32:3266–3276, 1985. doi:10.1103/PhysRevA.32.3266.
- 37 Oded Regev. Impossibility of a quantum speed-up with a faulty oracle. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfssdóttir, and Igor Walukiewicz, editors, *Automata, Languages and Programming (ICALP)*, pages 773–781. Springer Berlin Heidelberg, 2008. doi:10.1007/978-3-540-70575-8_63.
- 38 Ansis Rosmanis. Tight bounds for inverting permutations via compressed oracle arguments, 2022. arXiv:2103.08975v2.
- 39 Ansis Rosmanis. Quantum search with noisy oracle, 2023. arXiv:2309.14944v1.
- 40 Yaoyun Shi. Quantum lower bounds for the collision and the element distinctness problems, 2001. arXiv:quant-ph/0112086v1.
- 41 Kristan Temme. Runtime of unstructured search with a faulty Hamiltonian oracle. *Phys. Rev. A*, 90:022310, 2014. doi:10.1103/PhysRevA.90.022310.