Permutation Superposition Oracles for Quantum Query Lower Bounds

Christian Majenz[*] Giulio Malavolta[†‡] Michael Walter[§]

Abstract

We propose a generalization of Zhandry's compressed oracle method to random permutations, where an algorithm can query both the permutation and its inverse. We show how to use the resulting oracle simulation to bound the success probability of an algorithm for any predicate on input-output pairs, a key feature of Zhandry's technique that had hitherto resisted attempts at generalization to random permutations. One key technical ingredient is to use *strictly monotone factorizations* to represent the permutation in the oracle's database. As an application of our framework, we show that the one-round sponge construction is unconditionally preimage resistant in the random permutation model. This proves a conjecture by Unruh.

Contents

[*] Department of Applied Mathematics and Computer Science, Technical University of Denmark. *Email:* chmaj@dtu.dk
[†] Department of Computing Sciences, Bocconi University, Italy. *Email:* giulio.malavolta@unibocconi.it
[‡] Max Planck Institute for Security and Privacy, Germany.
[§] Faculty of Computer Science, Ruhr-Universität Bochum, Germany. *Email:* michael.walter@rub.de

1 Introduction

The random oracle model [BR93] is a popular heuristic in the analysis of cryptographic protocols, that abstracts cryptographic objects as random functions and provides oracle access to other algorithms. From a theoretical standpoint, the random oracle model allows one to prove unconditional statements about cryptographic protocols, in a clean and well-defined model. On the practical side, the random oracle model enables efficient cryptographic schemes and essentially every construction (be it a digital signature or a public-key encryption) used in practice relies, in one way or another, on this heuristic in order to analyze security.[1] When considering security against quantum algorithms, it is natural to extend this model to allow the algorithms to query the random function on a superposition of inputs. This is commonly referred to as the quantum random oracle model (QROM) [BDF$^+$11]. However, many of the techniques (and proofs) developed in the (classical) random oracle model do not immediately carry over to the QROM. To illustrate the difference, it suffices to note that in the classical settings the reduction can read the queries of the algorithm, whereas the same action in the QROM may arbitrarily disturb the state of the algorithm.

To cope with this, the community has developed a series of new techniques to analyze quantum algorithm in these settings. An important method in these settings is the *compressed oracle* technique [Zha19]. Conceptually, this technique is the quantum analogue of the classical lazy-sampling method, which allows the reduction to define the random function only on the inputs queried by the attacker. At a more technical level, the technique considers a purified version of the random function, that allows the reduction to directly inspect the internal state of the compressed oracle simulation (the so-called *database*), in order to gain partial/approximate knowledge about the queries made by the algorithm. An extremely useful property in this context is that the compressed oracle simulation stores (a superposition of) a list of input-output pairs, so to learn something about the value $H(x)$, and whether $H(x)$ is known to the adversary, it is only necessary to inspect one register. This technique has proven extremely successful in analyzing indifferentiability of cryptographic schemes [Zha19], security reductions for the Fiat-Shamir transformation [LZ19b, DFMS21, DFMS21] and the Fujisaki-Okamoto transformation [BHH$^+$19, DFMS21, HHM22], and even new lower bounds on the query complexity of quantum algorithms [Zha19, LZ19a, CFHL21] and space-time trade-offs [HM23].

The Random Permutation Model. In the random permutation model, algorithms are given oracle access to a uniformly sampled permutation $\pi \in S_N$, as well as its inverse π^{-1}. This variant of the random oracle model is motivated by cryptographic schemes, such as the Feistel construction for pseudorandom permutations [LR88] or the industry-standard SHA-3 hash function [Dwo15], where an attacker has access to both the permutation and its inverse. When considering quantum attackers, it is therefore equally natural to assume that such a permutation can be implemented on a quantum computer (as it is publicly known), and hence queried in superposition. Accordingly, it is natural to model this situation by a *quantum-accessible random permutation oracle*, where one considers the unitary[2]

$$U^\pi |x\rangle |y\rangle = |x\rangle |y \oplus \pi(x)\rangle$$

and one gives the adversary query access to U^π and $U^{\pi^{-1}}$. Classically, the random oracle and the random permutation model are essentially equivalent. This is in stark contrast to the quantum setting where no such connection is known. So far, it has proven difficult to repeat the success of Zhandry's compressed oracle technique for analyzing quantum query access to a uniformly random permutation: Despite several attempts to come up with a full-fledged compressed permutation oracle [CMSZ19, Unr21, Cza21] the problem is still open. On the other hand, few existing results rely on a bare-bones, "un-compressed" superposition oracle for permutations [ABKM22, ABK$^+$, ABPS23], whereas recent works [Ros21, Unr23] have made partial progress on this problem but without being able to apply the formalism towards new query bounds.

Arguably, the main reason for the lack of progress is, that the compressed oracle technique relies on the statistical independence of the output values of a random function, but the output values of a random permutation are, of course, not independent (more discussion on this later). The purpose of this work is to make progress on this front, and expand our technical toolkit in the analysis of random permutation oracles.

[1] Note however that the ROM (and QROM) are both fundamentally uninstantiable [CGH04].

[2] Alternatively, one can consider the *in-place* permutation V^π defined as $V^\pi |x\rangle = |\pi(x)\rangle$, which is well defined since π is a bijection, and give the adversary query access to V^π and $V^{\pi^{-1}} = (V^\pi)^\dagger$. For the purposes of query bounds both models are equivalent, since either can be simulated using two queries to the other. We discuss this in more detail in Section 4.

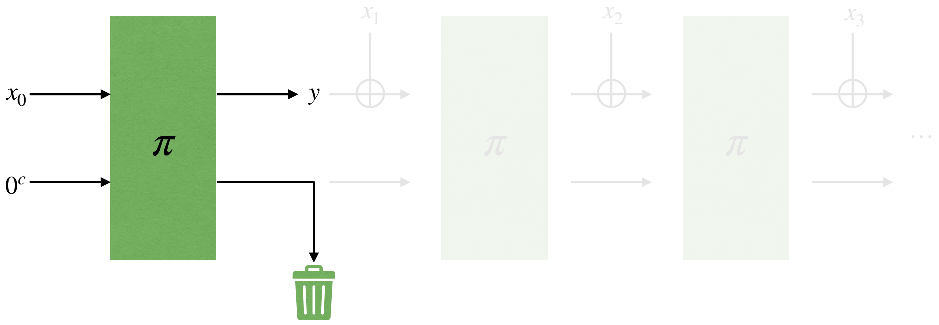

Figure 1: The 1-round sponge.

1.1 Summary of Contributions

In this work we propose a new approach to analyze quantumly-accessible permutation and prove query lower bounds in this setting. Our main ingredient is a new analysis of a representation of a permutation, known as its *strictly monotone factorization*, which may be of independent interest. At a technical level, we prove a series of lemmas that facilitate proofs of query lower bound in the random permutation model. Specifically:

- We present a *permutation* oracle (Section 4), that is the permutation analogue of Zhandry's compressed oracle technique. Our oracle makes crucial use of a particular representation of a permutation, known as its strictly monotone factorization (Section 3).
- We prove a *fundamental lemma* for permutation oracles (Section 5), where we describe a procedure to (approximately) determine whether an input was queried by the adversary, and we give a bound on the precision of its output.
- We propose a *progress measure* for permutation oracles (Section 6), that bounds the success probability of an algorithm after q queries to find an input/output pair that satisfies a given relation.
- We prove, as our *main theorem*, a general bound (Theorem 6.11) for any adversary to produce an input-output pair satisfying some relation R, for any R. This bound states the following:

Theorem 1.1 (Informal)**.** *Let $\mathcal{A}$ be an algorithm with quantum query access to a random permutation $\pi \in S_N$ and its inverse π^{-1}, and let R be a relation. If $\mathcal{A}$ makes at most q queries and outputs x, then*

$$\Pr[(x, \pi(x)) \in R] \leq O\left(\frac{q^3 r_{\max} \ln(N)}{N}\right),$$

where $r_{\max} = \max\{\max_x |R_x|, \max_y |R_y^{\mathrm{inv}}|\}$, with $R_x = \{y : (x,y) \in R\}$ and $R_y^{\mathrm{inv}} = \{x : (x,y) \in R\}$.

Illustrating the power of the new approach, we obtain as special cases the pre-image resistance of the sponge construction for the special case of one absorption round (see Fig. 1), and the double-sided zero-search conjecture [Unr23]. The former is the problem of finding an x such that $\pi(x\|0^c) = y\|0^c$ for some y; it was the original motivation of our work. Thus (Section 7):

Corollary 1.2 (One-Round Sponge, informal)**.** *Let $\mathcal{A}$ be an algorithm with quantum query access to a random permutation $\pi \in S_{\{0,1\}^n}$ and its inverse π^{-1}, let $c \in [n]$, and let $y \in \{0,1\}^{n-c}$. If $\mathcal{A}$ makes at most q queries and outputs x, then*

$$\Pr[\exists y' \in \{0,1\}^c : \pi(x\|0^c) = y\|y'] \leq O\left(\frac{q^3 n}{2^{\min(c,n-c)}}\right).$$

Corollary 1.3 (Double-Sided Zero-Search, informal)**.** *Let $\mathcal{A}$ be an algorithm with quantum query access to a random permutation $\pi \in S_{\{0,1\}^n}$ and its inverse π^{-1} and let $c \in [n]$. If $\mathcal{A}$ makes at most q queries and outputs x, then*

$$\Pr[\exists y \in \{0,1\}^{n-c} : \pi(x\|0^c) = y\|0^c] \leq O\left(\frac{q^3 n}{2^c}\right).$$

1.2 Key Challenges and Techniques

To understand the main challenges of extending Zhandry's method to permutations, and how we overcome these to prove our results, it is useful to recall the key properties of the compressed oracle method, and see why they fail for the case of permutation. The output values of a random function are independent as random variables. The crucial implication of this fact is that there exists a list of random variables that are

(i) independent, and
(ii) learning one output requires looking at only one of the random variables, and
(iii) each variable stores only information about one output.

It is easy to find a representation of a random permutation as a list of random variables that has some of the properties, but it is manifestly impossible to find one that has all, as the outputs are not independent. For instance, by knowing that $\pi(x) = y$ we can also infer that $\pi(x') \neq y$, for all $x' \neq x$.

One can therefore parameterize the set of approaches by which of the properties is given up on. In [Unr21], for example, a formalization using lazy sampling of permutations as partial functions is used, giving up on independence. In [Ros21], an analysis via the representation theory of the permutation group is conducted, which yields independent data via the decomposition into irreducible representation and symmetries, but gives up at least partially on locality.

Random Permutations from Independent Transpositions. Our main idea is to use the well-known fact that any permutation $\pi \in S_N$ admits a unique decomposition

$$\pi = (N\, t_N)\,(N{-}1\; t_{N-1}) \cdots (2\, t_2)\,(1\, t_1) \tag{1.1}$$

where we denote by $(k\, t_k)$ the transposition that sends k to t_k and viceversa. If one leaves out trivial transpositions (i.e., the factors with $k = t_k$), one obtains a so-called *strictly monotone* factorization of a permutation. It is well-known that any permutation $\pi \in S_N$ has a unique strictly monotone factorization, with number of terms equal to the Cayley distance between π and the identity permutation, that is, the minimum number of transpositions (of arbitrary type) in any factorization of π.

A useful property of this representation is that the transpositions making up a uniformly random permutation via the strictly monotone factorization are independent; t_k is uniformly chosen from the set $\{1, \ldots, k\}$. The main, less easy-to-see, property of this decomposition that we are going to use, is the fact that we can "track" the set of transpositions that act non-trivially on a given input x (which we refer to as being *active* for x). Crucial to our analysis is the fact that, for any given input x, the expected number of active transpositions is *small*, i.e., at most about $\ln(N)$. To compute the random permutation, only a small amount of information about non-active transpositions is retrieved, namely that they are not active. On the other hand, the output sensitively depends on the value t_k for an active location k, in the sense that changing t_k to any other value changes the output. This gives us a way to quantify the *sparsity* of the (quantum analogue of the) list of transposition that has been read. Although this means we only have an approximate *and* relaxed variant of the second property above, we will be able to show that this quantity is small enough to emulate the functionality of the compressed oracle method for the case of permutations.

The Permutation Oracle. With the above discussion in mind, let us now describe (a simplified version of) our permutation oracle. The simulation initializes a *database* of N registers $D = D_1 \ldots D_N$ and the k-th register D_k is initialized with the state

$$|+_k\rangle = \frac{1}{\sqrt{k}} \sum_{t=1}^{k} |t\rangle$$

which will be interpreted as the uniform superposition over all possible values t_k in Eq. (1.1). Since any given permutation $\pi \in S_N$ uniquely determines a basis state $|\pi\rangle = |t_1, \ldots, t_N\rangle$, it is easy to see that the initial state of the database corresponds to a uniform superposition over all possible permutations:

$$\bigotimes_{k=1}^{N} |+_k\rangle_{D_k} = \frac{1}{\sqrt{N!}} \sum_{\pi \in S_N} |\pi\rangle_D \,.$$

We can then define the forward query oracle (O^{SPO}_{XYD}) and the inverse query oracle ($O^{\mathsf{SPO},\mathrm{inv}}_{XYD}$) provided to the adversary by their actions of the basis states $|x\rangle_X$, $|y\rangle_Y$, and $|\pi\rangle_D$ as:

$$\begin{aligned} O^{\mathsf{SPO}}_{XYD} |x, y, \pi\rangle_{XYD} &= |x, y \oplus \pi(x), \pi\rangle_{XYD} \,, \\ O^{\mathsf{SPO},\mathrm{inv}}_{XYD} |x, y, \pi\rangle_{XYD} &= |x, y \oplus \pi^{-1}(x), \pi\rangle_{XYD} \,. \end{aligned}$$

In other words, the oracles apply a series of transpositions controlled on the current state of the database.

A Fundamental Lemma for Permutation Oracles. What makes the Zhandry's framework so useful is the ability to read off the database information about the adversary's state.[3] Thus our next task is to develop the necessary machinery for doing so. The first step is to prove a *fundamental lemma*, a statement akin to the existing bound for random functions [Zha19, CFHL21]. Fix an input x and an output y, simplifying a bit, the content of the fundamental lemma is the approximate equivalence of the following two experiments:

Exp. I: Read the value $\pi(x)$ off the database state and accept if $\pi(x) = y$, reject otherwise.

Exp. II: Check if the adversary has queried x, and reject if this is not the case. Else proceed as above.

The (approximate) equivalence of these two experiments is useful to implement a somewhat "gentle" measurement on the database state, for a given input x, since if we detect that the adversary never queried x, there is no need to disturb the state any further.

Of course at this point it is not clear what we exactly mean by "the adversary has queried x", and so the next step is to make this notion more precise. We observe that querying the superposition oracle on a basis state $|x\rangle$ must have a non-trivial effect on database location D_x in the computational basis, since it is determining the value of the first transposition, which is always active for x. Thus, we can formally define this quantity via a binary-outcome projective measurement

$$\mathcal{M}_{D_x} := \{|+_x\rangle\langle +_x|_{D_x}, I - |+_x\rangle\langle +_x|_{D_x}\}$$

and conditioning on the second outcome occurring. As a sanity check, note that if the first outcome is observed instead, then the database state is in its initial condition. With some routine calculation, we can then derive a bound

$$|\Pr[\text{Exp. I accepts}] - \Pr[\text{Exp. II accepts}]| \leq \sqrt{\frac{1}{x}}$$

where the term $1/x$ comes from the non-commutativity of the measurement $\mathcal{M}_{D_x}$ and the standard basis measurement used to determine π.

Unfortunately this bound on its own is not very meaningful: To see why, simply take $x = 1$ where we obtain a trivial bound. The source of this problem is the *asymmetric* treatment of different registers in the representation of the permutation, where lower registers have a much smaller set of possible transpositions. To deal with this, we introduce our next idea.

Twirling the Oracle. We overcome the challenge by randomizing the order in which we apply the strictly monotone decomposition. We address this by pre- and post-composing, or "twirling", the permutation oracle with two random permutations τ and σ. In other words, we define the *twirled* version of the permutation oracle as

$$\begin{aligned} O^{\mathsf{TSPO}^{\sigma,\tau}}_{XYD} |x, y, \pi\rangle_{XYD} &= |x, y \oplus \tau^{-1}(\pi(\sigma(x))), \pi\rangle_{XYD} \\ O^{\mathsf{TSPO}^{\sigma,\tau},\mathrm{inv}}_{XYD} |x, y, \pi\rangle_{XYD} &= |x, y \oplus \sigma^{-1}(\pi^{-1}(\tau(x))), \pi\rangle_{XYD} \end{aligned}$$

for forward and inverse queries, respectively. This yields our final construction, which we call the *twirled superposition permutation oracle*. Note that the permutations τ and σ are treated differently than π in the

[3] Although the analogy with Zahndry's technique is helpful for understanding our framework, the direct comparison is somewhat inaccurate. In particular, contrary to Zhandry's method, we will not attempt to *compress* the database in any way. As a consequence of this, our simulation will not be computationally efficient, which is sufficient to prove query lower bounds. We leave the development for an efficient version of our framework as ground for future work.

simulation, since we do not require any special property from their representation and their sole purpose is to randomize the view of the adversary in the sense that the adversary does not know which inputs correspond to small values x in the untwirled oracle. Another way to describe the twirled superposition permutation oracle is that it is constructed based on the strictly monotone factorization *in a random order*. This technique renders the permutation actually stored in a quantum register independent from the view of the algorithm interacting with the twirled superposition permutation oracle.

Equipped with this oracle, our bound obtained in the analysis of the fundamental lemma translates into an *expectation* over the random choice of the register x. Considering the *square* of the difference between the success probability of the two experiments, and taking the expectation over x, we obtain the bound

$$\frac{1}{N}\sum_{x=1}^{N}\frac{1}{x} \leq \frac{\ln(N)+1}{N}$$

using a standard bound on the harmonic sum. Note that the final bound is independent of x.

The Progress Measure. Once we have established a procedure to read information off the database, what is left to be decided is *what* we want to read from the database. Due to the challenge described above, we cannot straightforwardly implement a predicate that checks whether there exist an input-output pair $(x, \pi(x))$ that was read by the adversary, such that $(x, \pi(x)) \in R$, for some relation R. Instead, for a given input x, our progress measure is defined in terms of the following two-step procedure:

(i) Apply the projective measurement $\mathcal{M}_x$ defined above, and reject unless the second outcome is obtained (intuitively, this rejects unless the permutation was queried on input x by the adversary).

(ii) Check if $(x, \pi(x))$ indeed satisfies the relation R, which can be implemented by the following predicate:

$$\Pi_D^{R,x} := \sum_{\pi\in S_N:(x,\pi(x))\in R} |\pi\rangle\langle\pi|_D.$$

This procedure can be summarized by the following measurement operator:

$$E_D^{R,x} := \Pi_D^{R,x}(I - |+_x\rangle\langle+_x|_{D_x}).$$

It is easy to bound the progress measure if the adversary makes no query, since the above projection is acting on a uniform superposition. The challenge, which is the technically most involved part of our work, is to track the bound on the progress measure as the adversary queries the oracles in the forward and inverse direction. At a very high-level, we achieve this by splitting the effect of the action of the oracle $Q_{XYD}^{\mathsf{SPO}} = \{O_{XYD}^{\mathsf{SPO}}, O_{XYD}^{\mathsf{SPO},\mathrm{inv}}\}$ in two terms

$$\begin{aligned}
&\left\|E_D^{R,x} Q_{XYD}^{\mathsf{SPO}}|\phi\rangle\right\| - \left\|E_D^{R,x}|\phi\rangle\right\| \\
&\leq \left\|E_D^{R,x} Q_{XYD}^{\mathsf{SPO}}(I - E_D^{R,x})|\phi\rangle\right\| \\
&\leq \left\|E_D^{R,x} Q_{XYD}^{\mathsf{SPO}}(I - \Pi_D^{R,x})(I - |+_x\rangle\langle+_x|_{D_x})|\phi\rangle\right\| + \left\|E_D^{R,x} Q_{XYD}^{\mathsf{SPO}}|+_x\rangle\langle+_x|_{D_x}|\phi\rangle\right\|
\end{aligned}$$

for any state $|\phi\rangle$. We bound the two summands separately.

The RHS summand is bounded with a delicate analysis on the effect of the query unitary on the joint database-adversary state. We refer the reader to the technical sections for the calculations and we only mention here that the bound that one obtains with such analysis will not be sufficient for our main theorem. Instead, we will once again use the *randomization* of the register x and the twirling of the permutation π to transform the worst-case bound in a much sharper *average-case* bound

$$\frac{1}{N}\sum_{x=1}^{N}\left\|E_D^{R,x} Q_{XYD}^{\mathsf{SPO}}|+_x\rangle\langle+_x|_{D_x}|\phi\rangle\right\|^2 \leq O\left(\frac{q^2 r_{\max}\ln(N)}{N^2}\right)$$

where $r_{\max}$ is the maximum number of y such that $(x, y) \in R$, for all x. We can then turn this inequality back to a *worst-case* bound over x with a pidgeonhole argument, at the cost of losing a factor N in the bound. Fortunately, the resulting term is still small enough to obtain a good bound. Next, we deal with the LHS summand of the above bound.

Sparsity Analysis. To bound the LHS of the summand, let us first manipulate the expression

$$\begin{aligned}\left\|E_D^{R,x} Q^{\mathsf{SPO}}_{XYD}(I-\Pi_D^{R,x})(I-|+_x\rangle\langle+_x|_{D_x})\,|\phi\rangle\right\| &= \left\|E_D^{R,x}(I-\Pi_D^{R,x})Q^{\mathsf{SPO}}_{XYD}(I-|+_x\rangle\langle+_x|_{D_x})\,|\phi\rangle\right\| \\ &\le \left\|E_D^{R,x}(I-\Pi_D^{R,x})Q^{\mathsf{SPO}}_{XYD}\right\| \cdot \left\|(I-|+_x\rangle\langle+_x|_{D_x})\,|\phi\rangle\right\| \\ &= \left\|E_D^{R,x}(I-\Pi_D^{R,x})\right\| \cdot \left\|(I-|+_x\rangle\langle+_x|_{D_x})\,|\phi\rangle\right\| \\ &\le \sqrt{\frac{r_{\max}}{x}}\left\|(I-|+_x\rangle\langle+_x|_{D_x})\,|\phi\rangle\right\|\end{aligned}$$

where the first inequality follows by the submultiplicativity of the operator norm and the second inequality is obtained by observing that

$$\left\|E_D^{R,x}(I-\Pi_D^{R,x})\right\| = \left\|\Pi_D^{R,x}(I-|+_x\rangle\!\langle+_x|_{D_x})(I-\Pi_D^{R,x})\right\| = \left\|\Pi_D^{R,x}\,|+_x\rangle\!\langle+_x|_{D_x}\,(I-\Pi_D^{R,x})\right\| \le \left\|\Pi_D^{R,x}\,|+_x\rangle\!\langle+_x|_{D_x}\right\|$$

which can be bound to $\sqrt{r_{\max}/x}$ using the same argument as in the fundamental lemma. Thus, bounding this term boils down to bounding the amount of locations read by the adversary, i.e., the number of *active* registers in the database. Once again, the number of active locations on the initial state of the database is zero, so bounding this term involves analyzing the effect of the query unitary Q_{XYD} on the joint database-adversary state. A delicate analysis leads to a bound of

$$\frac{1}{N}\sum_{x=1}^{N}\frac{|R|}{x}\left\|(I-|+_x\rangle\langle+_x|_{D_x})\,|\phi\rangle\right\|^2 \le O\left(\frac{q^3 r_{\max}\ln(N)}{N^2}\right)$$

in expectation over x, which we can once again turn into a worst-case bound on every x at the cost of an extra $1/N$ factor. We refer the reader to the technical sections for more details.

Putting Things Together. Overall, the above analysis allow us to bound, via our *progress measure*, how the success probability of the predicate $E^{R,x}$ evolves as the adversary performs more queries. The final but crucial observation is that if x is the output of the algorithm $\mathcal{A}$, the measurement $E^{R,x}$ accepts precisely if Exp. II accepts. Applying the fundamental lemma to this bound, we obtain a bound on the acceptance probability of Exp. I (as defined above), which is the quantity that we are interested in.

This outline ignores many subtle aspects of the proof, but it contains the main ideas. Putting the bounds on the two main terms together, we obtain Theorem 1.1 and hence Corollaries 1.2 and 1.3.

1.3 Concurrent Independent Work

A recent manuscript by Carolan and Poremba [CP24], developed concurrently and independently from our work, also shows a proof for the double-sided zero-search conjecture of Unruh and the one-wayness of the one-round sponge. The techniques used to prove the bound are quite different and their work achieves the result using a worst-case to average-case reduction for random permutation, then appealing to known bounds. The advantage of their approach is that the bound obtained is *tight*. In contrast, our paper proposes a new framework to analyze permutation oracles and a theorem that applies to arbitrary relations on (single) input-output pairs, promising opportunities for generalization.

2 Preliminaries

We abbreviate $[N] := \{1, 2, \ldots, N\}$. For convenience, we assume that $N = 2^n$, so we can identify $[N] \cong \{0,1\}^n$ and use $\oplus$ to mean bitwise addition. For a set S, a probability distribution μ and a (classical or quantum) algorithm $\mathcal{A}$, we write $x \leftarrow S$, $x \leftarrow D$, and $x \leftarrow \mathcal{A}$ for sampling a uniformly random element x from S, sampling x according to the distribution D, or running an algorithm $\mathcal{A}$ to produce an output x. If the output is quantum, we usually use upper-case letters, following our conventions for quantum registers discussed below.

Combinatorics. We will also use the fact that for the *harmonic numbers*

$$H_N := \sum_{k=1}^{N} \frac{1}{k}, \tag{2.1}$$

the quantity $H_N - \ln N$ is monotonically decreasing with N. In particular, it holds that

$$H_N \leq \ln(N) + 1 \tag{2.2}$$

for every $N \geq 1$.

Quantum Information. Here we provide some preliminary background on quantum mechanics and quantum information. For more in-depth accounts we refer the reader to [NC00, Wat18].

We will label quantum systems by A, B, X, Y, etc. Any quantum system A is characterized by a Hilbert space $\mathcal{H}_A$. When $\mathcal{H}_A = \mathbb{C}^{\Sigma_A}$ for some finite set Σ_A, we call A a *quantum register.* This means that $\mathcal{H}_A$ has an orthonormal *standard basis* $|a\rangle$ labeled by the elements $a \in \Sigma_A$. When $\Sigma = [N]$, we can identify $\mathcal{H} = \mathbb{C}^N$ with its standard basis $|a\rangle$ for $a \in \mathbb{Z}_N$. If we have quantum system composed of two registers, say A and B, then the corresponding Hilbert space is the tensor product of the individual Hilbert spaces $\mathcal{H}_{AB} = \mathcal{H}_A \otimes \mathcal{H}_B \cong \mathbb{C}^{\Sigma_{AB}}$, with $\Sigma_{AB} = \Sigma_A \times \Sigma_B$ labeling the standard product basis. Accordingly, we may think of the composite system $AB = (A, B)$ as a register, and similarly for any collection S of registers.

States of a quantum system are given by *density operators*, that is, positive semi-definite operators of trace one. We call a state *pure* if this operator is a rank-one orthogonal projector, i.e., equal to $|\phi\rangle\langle\phi|$, where $|\phi\rangle$ is a unit vector. We will often identify pure states with unit vectors. The *trace distance* between two states ρ and τ, denoted by $\mathsf{Td}(\rho, \tau)$ is defined as

$$\mathsf{Td}(\rho, \tau) = \frac{1}{2}\|\rho - \tau\|_1 = \frac{1}{2} \operatorname{tr}\left(\sqrt{(\rho - \tau)^\dagger(\rho - \tau)}\right).$$

The operational meaning of the trace distance is that $\frac{1}{2}(1 + \mathsf{Td}(\rho, \tau))$ is the maximal probability that two states ρ and τ can be distinguished by any (possibly unbounded) quantum channel, when given one or the other with equal probability.

There are two basic kinds of quantum operations. The first is to apply unitary operators, or *unitaries.* If U is a unitary on $\mathcal{H}$ and we apply it to a state ρ, the result is $U\rho U^\dagger$, which is again a state. We denote by $\mathcal{U}(\mathcal{H})$ the group of unitary operators on $\mathcal{H}$, and abbreviate $\mathcal{U}(N) = \mathcal{U}(\mathbb{C}^N)$. The second is to measure the quantum state. We will only require *projective measurements*, which are given by a family of orthogonal projections $\{P_\omega\}_{\omega\in\Omega}$, labeled by some finite index set Ω, such that $\sum_{\omega\in\Omega} P_\omega = I$. If one applies such a measurement to a system in state ρ, then the probability of seeing outcome $\omega \in \Omega$ is $p_\omega = \operatorname{tr}(P_\omega \rho)$, in which case the state changes to $P_\omega \rho P_\omega / p_\omega$. If $\{P_a\} = \{|a\rangle\langle a|\}$ consists of the projections onto the standard basis of some register, this is called a *standard basis measurement.*

We will use subscripts to denote the corresponding quantum system or tensor factors, e.g., ρ_{AB} denotes a density operator on $\mathcal{H}_{AB} = \mathcal{H}_A \otimes \mathcal{H}_B$, and $\rho_{AB} = |\Psi\rangle\langle\Psi|_{AB}$ in the case of a pure state, with $|\Psi\rangle_{AB} \in \mathcal{H}_{AB}$. Similarly, we write U_A in the case of a unitary on $\mathcal{H}_A$. In particular, I_A denotes the identity operator on $\mathcal{H}_A$.

For unitaries and measurement operators (but never for states), it will be useful to identify operators on some Hilbert space with operators on some any other Hilbert space which includes the former as a tensor factor, by tensoring with the identity operator. For example, we will often abbreviate the operator $U_A \otimes I_B$ on $\mathcal{H}_{AB}$ simply by U_A if no confusion can arise. This is useful if we want to quantum operations to a subset of registers. For example, if U_A is a unitary and ρ_{AB} a state, then $U_A \rho_{AB} U_A^\dagger = (U_A \otimes I_B)\rho_{AB}(U_A^\dagger \otimes I_B)$ is the result of applying the unitary U_A to the first register A when the overall system starts out in state ρ_{AB}, and similarly for measurements.

We also recall the gentle measurement lemma [Wil19, Lemma 9.4.2].

Fact 2.1 (Gentle measurement). *Let ρ be a quantum state, and let $\{P, I - P\}$ be any projective measurement with two outcomes. If $\operatorname{tr}(P\rho) \geq 1 - \delta$, the post-measurement state $\rho' := P\rho P / \operatorname{tr}(P\rho)$ satisfies*

$$\mathsf{Td}(\rho, \rho') \leq \sqrt{\delta}.$$

Quantum Algorithms. In this work we consider the query complexity of algorithms with quantum access to oracles. An oracle is modeled by one or more unitaries O operating on an input/output register Z and possibly some internal register D (which will always be initialized explicitly). A quantum algorithm $\mathcal{A}$ making queries to this oracle has, without loss of generality, two registers – the oracle's input/output register Z and an internal work register A. It takes the following form: First, the algorithm's registers are initialized in the initial state $|0\rangle_{AZ}$. Then the algorithm alternatingly applies oracle-independent unitaries and query unitaries:

$$U^{(q)}_{AZ} O_{ZD} U^{(q-1)}_{AZ} O_{ZD} \dots U^{(1)}_{AZ} O_{ZD} U^{(0)}_{AZ} \tag{2.3}$$

Note that $\mathcal{A}$ can be given access to any oracle with input/output register Z. Finally, some (sub)registers might be measured or returned directly to obtain the classical and quantum outcomes of the algorithm. We write $\mathcal{A}^O$ for such an application of a query algorithm to an oracle O. In the above situation, we say that the algorithm makes q queries to the oracle. Thus we only consider the query complexity of an algorithm, but not the time complexity of the unitaries $U^{(k)}$. In other words, these unitaries need not be efficient. In particular, any advice state can be placed in the adversary's quantum memory by using the first unitary $U^{(0)}$.

3 Random Permutations

We denote by S_N the permutation group on N elements, that is, the group of bijections of $[N]$. We have a chain of subgroups $S_1 \subset S_2 \subset \dots \subset S_N$, where for each $j \in [N-1]$ we think of S_j as those permutations in S_{j+1} that fix the element $j+1$. For $k, l \in [N]$, we denote by $\tau = (k\ l)$ the *transposition* that sends $\tau(k) = l$ and $\tau(l) = k$. It will be convenient to allow $k = l$, in which case $(k\ l) = (k\ k)$ is the identity permutation.

3.1 Random Permutations from Independent Transpositions

The starting point for our work is the following representation of permutations.

Lemma 3.1. *For any $\pi \in S_N$, there exist unique numbers $t_k \in [k]$ for $k \in [N]$ such that*

$$\pi = (N\ t_N)\,(N-1\ t_{N-1}) \cdots (2\ t_2)\,(1\ t_1)\,. \tag{3.1}$$

While we always have $t_1 = 1$, it is useful to include this term to obtain simpler formulas below.

Proof. Any permutation $\pi \in S_N$ has a unique decomposition of the form

$$\pi = (N\ t)\,\sigma, \tag{3.2}$$

where $t \in [N]$ and $\sigma \in S_{N-1}$. Indeed, for (3.2) to hold we must have $t = \pi(N)$ and hence $\sigma = (N\ t)\,\pi$, but these formulas always define a valid decomposition of the form of Eq. (3.2). The lemma follows by induction. □

If one leaves out trivial transpositions in Eq. (3.1) (i.e., the factors with $k = t_k$), one obtains a so-called *strictly monotone* factorization. It is well-known that any permutation $\pi \in S_N$ has a unique strictly monotone factorization, with number of terms equal to the Cayley distance between π to the identity permutation, that is, the minimum number of transpositions (of arbitrary type) in any factorization of π.

Corollary 3.2. *A random permutation $\pi \in S_N$ is uniformly random if and only if the numbers $t_k \in [k]$ for $k \in [N]$ in Eq. (3.1) are independent and uniformly random.*

Given a permutation in the form of Eq. (3.1), it is easy to compute the inverse:

$$\pi^{-1} = (1\ t_1)\,(2\ t_2) \cdots (N-1\ t_{N-1})\,(N\ t_N)\,. \tag{3.3}$$

Note however that this decomposition is in general *not* of the form of Eq. (3.1). It will also be convenient to introduce the following notation:

$$\begin{aligned} \pi_{>k} &= (N\ t_N)\,(N-1\ t_{N-1}) \cdots (k+1\ t_{k+1})\,, \\ \pi_{<k} &= (k-1\ t_{k-1}) \cdots (2\ t_2)\,(1\ t_1)\,. \end{aligned} \tag{3.4}$$

Note that $\pi = \pi_{>k}\,(k\; t_k)\,\pi_{<k}$ and $\pi^{-1} = (\pi_{<k})^{-1}\,(k\; t_k)\,(\pi_{>k})^{-1}$. We use the convention that the subset-of-transpositions subscripts take precedence before other operations modifying a permutation, e.g. $\pi^{-1}_{<k} := (\pi_{<k})^{-1}$.

The following lemma will be useful in Section 6.2.

Lemma 3.3. *Let $\pi \in S_N$ be uniformly random and $k \in [N]$.*

(i) If $\xi \in S_k$ is uniformly random and independent from π, then $\pi_{>k}\xi$ is uniformly random in S_N.
(ii) If $\xi \in S_k$ is uniformly random and independent from π, then $\xi\pi_{>k}$ is uniformly random in S_N.
(iii) For every $\ell \in \{k+1,\dots,N\}$, it holds that $\Pr(\pi_{>k}(k) = \ell) = \frac{1}{N}$.

Proof. (i) This is clear from Corollary 3.2.

(ii) Using the notation of Eq. (3.4), we have

$$\begin{aligned}\xi\pi_{>k} &= \xi\,(N\; t_N)\,\xi^{-1}\xi\,(N{-}1\; t_{N-1})\,\xi^{-1}\cdots\xi\,(k+1\; t_{k+1})\,\xi^{-1}\xi\\ &= (N\; \xi(t_N))\,(N{-}1\; \xi(t_{N-1}))\cdots(k+1\; \xi(t_{k+1}))\,\xi,\end{aligned}$$

where we used that $\xi \in S_k$ and hence it fixes $k+1,\dots,N$, Now, the $t_\ell \in [\ell]$ for $\ell > k$ are uniformly random and independent from ξ, so the same is true for the $\xi(t_\ell) \in [\ell]$ for $\ell > k$. Thus we see that the distribution of $\xi\pi_{>k}$ is the same as the distribution of $\pi_{>k}\xi$, and the claim follows from part (i).

(iii) This follows from the preceding:

$$\Pr(\pi_{>k}(k) = \ell) = \Pr((\xi\pi_{>k})(k) = \ell) = \frac{1}{N}.$$

with ξ as in part (ii).

□

3.2 Active Transpositions

In the following we characterize the transpositions that, in the above decomposition of a permutation, are relevant to determine the action of the permutation on a given element. The results of this section will not be needed to prove our main theorem, but we include them here for the purpose of building up an intuition.

Given a permutation $\pi \in S_N$, consider its unique decomposition as in Lemma 3.1:

$$\pi = (N\; t_N)\,(N{-}1\; t_{N-1})\cdots(2\; t_2)\,(1\; t_1)\,,$$

When does a given transposition $(k\; t_k)$ impact the computation of $\pi(x)$ for some given $x \in [N]$? To study this we introduce the following definitions:

Definition 3.4 (Active sets)**.** Given a permutation $\pi \in S_N$ and $x, k \in [N]$, we say k is *active for π and x* if $\pi_{<k}(x) \in \{k, t_k\}$. Similarly, for $y \in [N]$ we say that k is *inverse-active for π and y* if $(\pi_{>k})^{-1}(y) \in \{k, t_k\}$.[4] We denote by $A(\pi, x), A^{\mathrm{inv}}(\pi, y) \subseteq [N]$ the set of active and inverse-active k, respectively, defined as above.

If k is active for π and x then changing t_k to any other value will lead to a different $\pi(x)$. Similarly, if k is inverse-active for π and y then changing t_k (in the decomposition of π) to any other value will lead to a different $\pi^{-1}(y)$. (The converses of these statements are in general not true.) Note that $\pi_{<k}(x) = k$ if and only if $x = k$, since $\pi_{<k} \in S_{k-1}$.

It is clear that the action of a permutation or its inverse on some element only depends on the corresponding active set:

Lemma 3.5. *Let $\pi \in S_N$ be a permutation. For any $x \in [N]$, let $A(\pi, x) = \{k_1 < \cdots < k_\ell\}$ denote the active locations, sorted in increasing order. Then we have:*

$$\pi(x) = (k_\ell\; t_{k_\ell})\left(k_{\ell-1}\; t_{k_{\ell-1}}\right)\cdots(k_2\; t_{k_2})\,(k_1\; t_{k_1})\,(x).$$

Similarly, if $y \in [N]$ and $A^{\mathrm{inv}}(\pi, y) = \{k_1 < \cdots < k_\ell\}$ are the corresponding inverse-active locations, then:

$$\pi^{-1}(y) = (k_1\; t_{k_1})\,(k_2\; t_{k_2})\cdots\left(k_{\ell-1}\; t_{k_{\ell-1}}\right)(k_\ell\; t_{k_\ell})\,(y).$$

[4] Note that this is in general *not* equivalent to saying that k is active for π^{-1} and y, as the latter would refer to the decomposition of π^{-1} in the form of Eq. (3.1), rather than to Eq. (3.3).

We now show that the event that k is active for a random permutation (and fixed x) is independent of the numbers $t_1, \ldots, t_{k-1}$, and compute the probability with which this happens:

Lemma 3.6. *Let $x \in [N]$. Then, for a uniformly random $\pi \in S_N$, we have:*

$$\Pr(k \in A(\pi, x) \mid t_1, \ldots, t_{k-1}) = \begin{cases} \frac{1}{k} & \text{if } x < k, \\ 1 & \text{if } x = k, \\ 0 & \text{if } x > k, \end{cases}$$

hence this is also equal to $\Pr(k \in A(\pi, x))$. *In particular, the events $k \in A(\pi, x)$ for $k \in [N]$ are independent.*

Proof. Recall that $k \in A(\pi, x)$ means that $\pi_{<k}(x) \in \{k, t_k\}$. For $x < k$, we have $\pi_{<k}(x) \in [k-1]$, hence the event $\pi_{<k}(x) \in \{k, t_k\}$ is equivalent to $\pi_{<k}(x) = t_k$. For a uniformly random π, the numbers $t_1, t_2, \ldots, t_k$ are independent and uniformly random (by Corollary 3.2). Hence $t_k \in [k]$ is uniformly random given $t_1, \ldots, t_{k-1}$. As $\pi_{<k}(x) \in [k-1]$ only depends on the latter, it follows that it coincides with t_k with probability $\frac{1}{k}$. For $x \geq k$ we have $\pi_{<k}(x) = x$, hence the event $\pi_{<k}(x) \in \{k, t_k\}$ is equivalent to $x = k$. □

Next we are going to bound the expected number of active locations for a random choice of the permutation.

Lemma 3.7. *Let $x \in [N]$. Then, for a uniformly random $\pi \in S_N$, we have:* $\mathbf{E}|A(\pi, x)| \leq 1 + \ln(N/x)$.

Proof. Recall that we denote by $H_n = \sum_{k=1}^{n} \frac{1}{k}$ the harmonic numbers. Then, using Lemma 3.6, we have

$$\mathbf{E}|A(\pi, x)| = 1 + \sum_{k=x+1}^{N} \frac{1}{k} = 1 + H_N - H_x \leq 1 + \ln N - \ln x.$$

where the inequality holds as $H_n - \ln n$ is monotonically decreasing with n. This completes the proof. □

We also provide a bound on the expected number of inverse-active locations. Since this is not the same as the expected number of active locations for the inverse, the bound differs from Lemma 3.7.

Lemma 3.8. *Let $y \in [N]$. Then, for a uniformly random $\pi \in S_N$, we have:* $\mathbf{E}|A^{\text{inv}}(\pi, y)| \leq 1 + \frac{2y-2}{N} < 3$.

Proof. Suppose first that there is some $k \in [N]$ such that $t_k = y$. Let $k^* \in [N]$ denote the largest such k. Since $t_{k^*} \in [k^*]$, we clearly must have $k^* \geq y$. Then we have $(\pi_{>k})^{-1}(y) = y \notin \{k, t_k\}$ for all $k > k^*$, $(\pi_{>k^*})^{-1}(y) = y = t_{k^*}$, and $(\pi_{>k})^{-1}(y) = k^* \notin \{k, t_k\}$ for all $k < k^*$. Together, we see that $A^{\text{inv}}(\pi, y) = \{k^*\}$. On the other hand, if $t_k \neq y$ for all $k \in [N]$ then $(\pi_{>k})^{-1}(y) = y$ for all $k \geq y$ and hence $A^{\text{inv}}(\pi, y) = \{y\} \cup A^{\text{inv}}(\pi_{<y}, t_y)$. Combining the above observations, we find that

$$\begin{aligned} \mathbf{E}|A^{\text{inv}}(\pi, y)| &= 1 + \Pr(t_y \neq y, \ldots, t_N \neq y)\, \mathbf{E}\big[A^{\text{inv}}(\pi_{<y}, t_y) \mid t_y \neq y, \ldots, t_N \neq y\big] \\ &= 1 + \frac{y-1}{N}\, \mathbf{E}\big[A^{\text{inv}}(\pi_{<y}, t_y) \mid t_y \neq y\big] \\ &= 1 + \frac{1}{N} \sum_{z=1}^{y-1} \mathbf{E}\big[A^{\text{inv}}(\pi_{<y}, z)\big], \end{aligned}$$

where we first used that $\pi_{<y}$ is independent from $t_y, \ldots, t_N$, and the last step holds because t_y is uniformly random in $[y-1]$ conditional on $t_y \neq y$. Since $\pi_{<y}$ is uniformly random in S_{y-1}, we obtain the following recurrence for $e(N, y) := \mathbf{E}_{\pi \leftarrow S_N} |A^{\text{inv}}(\pi, y)|$:

$$e(N, y) = 1 + \frac{1}{N} \sum_{z=1}^{y-1} e(y-1, z) = 1 + \frac{1}{N} f(y-1), \tag{3.5}$$

where we introduced the notation $f(n) := \sum_{y=1}^{n} e(n, y)$, with $f(0) = 0$, which in turn satisfies the recurrence

$$f(n) = \sum_{y=1}^{n} \left(1 + \frac{1}{n} f(y-1)\right) = n + \frac{1}{n} \sum_{k=0}^{n-1} f(k)$$

for all $n > 0$. It is easy to see that $f(n) \leq 2n$ for all $n \geq 0$ by using induction. Indeed, this is clearly true for $n = 0$, and if it holds that $f(k) \leq 2k$ for all $k < n$ then also $f(n) \leq n + \frac{1}{n} \sum_{k=0}^{n-1} 2k = n + n - 1 \leq 2n$. Using this estimate in Eq. (3.5) we obtain the desired result:

$$\mathbf{E}|A^{\text{inv}}(\pi, y)| = e(N, y) = 1 + \frac{1}{N} f(y-1) \leq 1 + \frac{2y-2}{N}.$$ □

4 Quantum Random Permutations Oracles

The decomposition of a random permutation introduced in Section 3.1 provides a way of sampling a random permutation by sampling many *independent and smaller* random data, namely the individual transpositions that make up the permutation (Corollary 3.2). Importantly, typical input-output pairs of the random permutation only depend on a few of them (Lemmas 3.7 and 3.8). In this section we will use this idea to construct an oracle that exactly simulate a quantum-accessible random permutation, but has an internal state that can be used to analyze quantum query algorithms. We first introduce some notation. Given any permutation $\pi \in S_N$, we denote by U^π the corresponding permutation operator on $\mathbb{C}^N \otimes \mathbb{C}^N$, that is,

$$U^\pi |x, y\rangle = |x, y \oplus \pi(x)\rangle \qquad \forall x, y \in [N].$$

This defines a unitary representation of S_N on $\mathbb{C}^N \otimes \mathbb{C}^N$.

Definition 4.1 (Quantum-accessible random permutation)**.** A *quantum-accessible random permutation* consists of query access to U^π and to $U^{\pi^{-1}}$, for a permutation $\pi \in S_N$ chosen uniformly at random.

When $\mathcal{A}$ is a query algorithm that gets query access to two oracles that act on $\mathbb{C}^N \otimes \mathbb{C}^N$ and $\pi \in S_N$ is a permutation, we write $\mathcal{A}^{U^\pi, U^{\pi^{-1}}}$ to indicate that we use the unitaries U^π and $U^{\pi^{-1}}$ as the two oracles.

Above we defined U^π by the usual formula for a quantum oracle corresponding to a Boolean function, but because π is a bijection we could also instead work with oracles that modify their input in-place, that is,

$$V^\pi |x\rangle = |\pi(x)\rangle \qquad \forall x \in [N].$$

However, to prove a query lower bound we will be able to consider either type of oracles. This is because the standard and the in-place variants can simulate each other at the cost of doubling the number of queries, if one is given access to the permutation as well as its inverse: it holds that

$$\begin{aligned} U^\pi_{XY} &= V^{\pi^{-1}}_X \, \mathrm{CNOT}_{X\to Y} \, V^\pi_X, \\ U^{\pi^{-1}}_{XY} &= V^\pi_X \, \mathrm{CNOT}_{X\to Y} \, V^{\pi^{-1}}_X, \end{aligned}$$

as well as

$$\begin{aligned} V^\pi_X |0\rangle_Y &= U^{\pi^{-1}}_{XY} \, \mathrm{SWAP}_{X\leftrightarrow Y} \, U^\pi_{XY} |0\rangle_Y \,, \\ V^{\pi^{-1}}_X |0\rangle_Y &= U^\pi_{XY} \, \mathrm{SWAP}_{X\leftrightarrow Y} \, U^{\pi^{-1}}_{XY} |0\rangle_Y \,. \end{aligned}$$

4.1 Superposition Permutation Oracle

We first construct a *superposition oracle* for random permutations. Like those for random functions, it is obtained by replacing the random (classical) choice of permutation by a uniform (quantum) superposition. Our oracle is specified by an internal quantum state space, an initial state, query unitaries (one for the random permutation and one for its inverse), and a recovery routine. The query unitaries will be constructed by applying the transpositions $(k\ t_k)$ in the right order, with each t_j obtained from the internal state of the oracle. Generalizing the approach of [CFHL21], we propose the following definition.

Definition 4.2 (Superposition permutation oracle)**.** The *superposition permutation oracle* (SPO) is defined as follows:

- The state space, called the *database*, consists of N registers, $D = D_1 \cdots D_N$ with the k-th register D_k having dimension k and computational basis $|1\rangle, \ldots, |k\rangle$. Any permutation $\pi \in S_N$, determines a basis state $|\pi\rangle_D = |t_1, \ldots, t_N\rangle_D$, where the numbers $t_k \in [k]$ are chosen as in Eq. (3.1).
- The initialization routine $\mathsf{Init}^{\mathsf{SPO}}_D$ initializes each register in a uniform superposition over the basis states. That is, the initial state of the database is

$$|\Phi_{\mathsf{SPO}}\rangle_D = \frac{1}{\sqrt{N!}} \sum_{\pi \in S_N} |\pi\rangle_D = \bigotimes_{k=1}^N |+_k\rangle_{D_k}\,, \text{ where } |+_k\rangle_{D_k} = \frac{1}{\sqrt{k}} \sum_{t=1}^k |t\rangle_{D_k}\,.$$

- There are two query unitaries, O^{SPO}_{XYD} and $O^{\mathsf{SPO},\mathrm{inv}}_{XYD}$, that define the two interfaces available to the query algorithm and that simulate oracles for a random permutation and its inverse. They are defined as follows: For all $x, y \in [N]$ and $\pi \in S_N$,

$$O^{\mathsf{SPO}}_{XYD} |x, y, \pi\rangle_{XYD} = |x, y \oplus \pi(x), \pi\rangle_{XYD} \,,$$
$$O^{\mathsf{SPO},\mathrm{inv}}_{XYD} |x, y, \pi\rangle_{XYD} = |x, y \oplus \pi^{-1}(x), \pi\rangle_{XYD} \,.$$

- The recovery routine $\mathsf{Rec}^{\mathsf{SPO}}_D$ simplify measures all registers $D_1, \ldots, D_N$ in the computational basis to obtain $t_k \in [k]$ for $k \in [N]$. It outputs the corresponding permutation according to Eq. (3.1).

When $\mathcal{A}$ is a query algorithm that gets query access to two oracles acting on two N-dimensional register X and Y, and if D is the database register of a superposition permutation oracle, we write $\mathcal{A}^{\mathsf{SPO}_D}$ to indicate that we use the interfaces O^{SPO}_{XYD} and $O^{\mathsf{SPO},\mathrm{inv}}_{XYD}$, respectively, to implement the two types of oracles queries. It is straightforward to verify that the SPO then exactly simulates a quantum-accessible random permutation.

Lemma 4.3. *Let $\mathcal{A}$ be a query algorithm that gets query access to two oracles that act on $\mathbb{C}^N \otimes \mathbb{C}^N$. Then the joint state of the classical random variable π and the quantum register B is the same for the following two experiments:*

(i) Sample $\pi \leftarrow S_N$ uniformly at random and run $B \leftarrow \mathcal{A}^{U^\pi, U^{\pi^{-1}}}$.
(ii) Run $\mathsf{Init}^{\mathsf{SPO}}_D$, then $B \leftarrow \mathcal{A}^{\mathsf{SPO}_D}$, and finally $\pi \leftarrow \mathsf{Rec}^{\mathsf{SPO}}_D$.

Proof. We show that both experiments give rise to the same joint state as the following:

(iii) Run $\mathsf{Init}^{\mathsf{SPO}}_D$, then $\pi \leftarrow \mathsf{Rec}^{\mathsf{SPO}}_D$, and finally $B \leftarrow \mathcal{A}^{\mathsf{SPO}_D}$.

Indeed, (i) and (iii) are equivalent by Corollary 3.2 and the fact that measuring a uniform superposition yields a uniformly random sample, while (iii) and (ii) are equivalent because measuring in the computational basis commutes with unitaries that are controlled on this basis. □

A consequence of Lemma 3.1 is that $\pi(x) = \pi_{\geq x}(x)$, so in particular $\pi(x)$ does not depend on $t_{x'}$ for $x' < x$. For the SPO, this means that when a query is made with input x, the registers $D_{x'}$ for $x' < x$ are not used, i.e., the query operator acts as the identity on them.

Lemma 4.4. *The* SPO *query operator fulfils the equation*

$$O^{\mathsf{SPO}}_{XYD} |x\rangle_X |y\rangle_Y |\pi_{\geq x}\rangle_{D_{\geq x}} = |x\rangle_X |y \oplus \pi_{\geq}(x)\rangle_Y |\pi_{\geq x}\rangle_{D_{\geq x}} \otimes I_{D_{<x}}.$$

Proof. This follows directly from Definition 4.2 and Lemma 3.1. □

4.2 Twirled Superposition Permutation Oracle

Just like is the case for Zhandry's compressed oracle for random functions, we would like to be able to inspect the internal state of the oracle (that is, the database) to gain partial, approximate knowledge about the queries made by the algorithm. However, there are two important caveats in the permutation case.

First, Zhandry's compressed oracle satisfies the extremely useful property that the compressed oracle stores (superpositions of) input-output pairs. This means that in order to learn something about the value $H(x)$ of the random function H at some point $x \in [N]$, and whether this value is known to the adversary, it is only necessary to measure one register. In contrast, we represent permutations as a product of transpositions and hence the database of our permutation oracle stores the analogous information in a less localized fashion. If we want to determine the value $\pi(x)$ of the random permutation π at some point $x \in [N]$, in general we may need to inspect all registers D_k for $k \geq x$. On the other hand, recall that for any fixed x, typically only $\tilde{O}(1)$ permutations are active and suffice to determine $\pi(x)$ (see Lemmas 3.5, 3.7 and 3.8).

Second, Zhandry's compressed oracle has the desirable feature that one can "jointly measure" whether a query algorithm has accessed a register *and* what function value it holds, with only a small error. In our permutation oracle the analogous procedure is to apply the binary measurement $\mathcal{M}_k := \{I - |{+_k}\rangle\langle{+_k}|_{D_k}, |{+_k}\rangle\langle{+_k}|_{D_k}\}$ to learn whether the k-th transposition has been accessed and, if so, measure in the computational basis to learn what its value is. However, since the size of the support of the uniform superposition $|+_k\rangle_{D_k}$ depends on k, the error in this "joint measurement" depends on k. For example, suppose an algorithm managed through some combination of queries and measurements to learn t_k with certainty for some particular $k \in [N]$. Then, conditional on this event, the database register D_k is in state $|t_k\rangle$. Applying the measurement $\mathcal{M}_k$ in this state will, however, return outcome $|+_k\rangle_{D_k}$ with probability $|\langle +_k|t_k\rangle|^2 = \frac{1}{k}$, which need not be small!

Both challenges can be addressed by pre- and post-composing, or "twirling", the SPO with two random permutations. This yields our final construction, which we call the *twirled superposition permutation oracle*. For convenience we first define a version where the twirls are fixed.

Definition 4.5 (Twirled superposition permutation oracle)**.** For any two fixed permutations $\sigma, \tau \in S_N$, the (σ, τ)*-twirled superposition permutation oracle* ($\mathsf{TSPO}^{\sigma,\tau}$) is defined as follows:

- The state space consists of the same *database* D as the superposition permutation oracle.
- The initialization routine is the same as for the superposition permutation oracle and hence we will continue to denote it by $\mathsf{Init}_D^{\mathsf{SPO}}$.
- There are two query unitaries, $O_{XYD}^{\mathsf{TSPO}^{\sigma,\tau}}$ and $O_{XYD}^{\mathsf{TSPO}^{\sigma,\tau},\mathrm{inv}}$, that define the two interfaces available to the query algorithm and that simulate oracles for a random permutation and its inverse. They are defined as follows. For all $x, y \in [N]$ and $\pi \in S_N$

  $$O_{XYD}^{\mathsf{TSPO}^{\sigma,\tau}} |x, y, \pi\rangle_{XYD} = |x, y \oplus \tau^{-1}(\pi(\sigma(x))), \pi\rangle_{XYD}\,,$$
  $$O_{XYD}^{\mathsf{TSPO}^{\sigma,\tau},\mathrm{inv}} |x, y, \pi\rangle_{XYD} = |x, y \oplus \sigma^{-1}(\pi^{-1}(\tau(x))), \pi\rangle_{XYD}\,.$$

  where X and Y are the N-dimensional target register of the oracles.
- The recovery routine $\mathsf{Rec}_D^{\mathsf{TSPO}^{\sigma,\tau}}$ first applies the recovery routine $\mathsf{Rec}^{\mathsf{SPO}}$ to obtain a permutation $\pi \in S_N$, and then returns $\tau^{-1}\pi\sigma$.

When $\mathcal{A}$ is a query algorithm that gets query access to two oracles acting on two N-dimensional register X and Y, and if D is the database register of a superposition permutation oracle, we write $\mathcal{A}^{\mathsf{TSPO}_D^{\sigma,\tau}}$ to indicate that we use the interfaces $O_{XYD}^{\mathsf{TSPO}^{\sigma,\tau}}$ and $O_{XYD}^{\mathsf{TSPO}^{\sigma,\tau},\mathrm{inv}}$, respectively, to implement the two types of oracles queries.

It is easy to see that the twirled superposition permutation oracles exactly simulate the ordinary one (whether the twirls are fixed or randomly sampled). Hence it also exactly simulates a quantum-accessible random permutation.

Lemma 4.6. *Let $\mathcal{A}$ be a query algorithm that gets query access to two oracles that act on two N-dimensional registers, and let $\sigma_0, \tau_0 \in S_N$ be fixed permutations. Then the joint classical-quantum state of the random variable π and the register B is the same for the following three experiments:*

(i) Run $\mathsf{Init}_D^{\mathsf{SPO}}$, then $B \leftarrow \mathcal{A}^{\mathsf{SPO}_D}$, and finally $\pi \leftarrow \mathsf{Rec}_D^{\mathsf{SPO}}$.
(ii) Run $\mathsf{Init}_D^{\mathsf{SPO}}$, then $B \leftarrow \mathcal{A}^{\mathsf{TSPO}_D^{\sigma_0,\tau_0}}$, and finally $\pi \leftarrow \mathsf{Rec}_D^{\mathsf{TSPO}^{\sigma_0,\tau_0}}$.
(iii) Sample $\sigma \leftarrow S_N$ and $\tau \leftarrow S_N$, run $\mathsf{Init}_D^{\mathsf{SPO}}$, then $B \leftarrow \mathcal{A}^{\mathsf{TSPO}_D^{\sigma,\tau}}$, and finally $\pi \leftarrow \mathsf{Rec}_D^{\mathsf{TSPO}^{\sigma,\tau}}$.

Moreover, in part (iii), σ, τ, and (π, B) are independent, and the three permutations σ, τ, π are independent and uniformly distributed.

Proof. It suffices to argue that (i) and (ii) result in the same joint state. By using Lemma 4.3 twice, we see that it suffices to compare the following two experiments:

(i') Sample $\pi \leftarrow S_N$ uniformly at random and run $B \leftarrow \mathcal{A}^{U^\pi, U^{\pi^{-1}}}$.
(ii') Sample $\pi \leftarrow S_N$ uniformly at random, run $B \leftarrow \mathcal{A}^{U^{\tau_0^{-1}} U^\pi U^{\sigma_0}, U^{\sigma_0^{-1}} U^{\pi^{-1}} U^{\tau_0}} = \mathcal{A}^{U^{\tau_0^{-1}\pi\sigma_0}, U^{(\tau_0^{-1}\pi\sigma_0)^{-1}}}$, and update $\pi \leftarrow \sigma_0^{-1}\pi\sigma_0$.

These are indeed equivalent, since if $\pi \in S_N$ is uniformly random then so is $\tau_0^{-1}\pi\sigma_0$, for fixed $\sigma_0, \tau_0 \in S_N$. □

It will be convenient to relate the twirled superposition oracle to the untwirled one by viewing the twirling as an action on the database D. To this end, define the left and right actions of S_N on D as

$$L^{\tau} |\pi\rangle = |\tau\pi\rangle ,$$
$$R^{\sigma} |\pi\rangle = |\pi\sigma^{-1}\rangle .$$

Then we have the following lemma, which states that the superposition oracle can be expressed in terms of the untwirled one, sandwiched by a basis change implemented by the operators L_τ and R_σ as defined above.

Lemma 4.7. *For all $\sigma, \tau \in S_N$, it holds that*

$$O_{XYD}^{\mathsf{TSPO}^{\sigma,\tau}} = (L_D^{\tau} R_D^{\sigma}) O_{XYD}^{\mathsf{SPO}} (L_D^{\tau^{-1}} R_D^{\sigma^{-1}}),$$
$$O_{XYD}^{\mathsf{TSPO}^{\sigma,\tau},\mathrm{inv}} = (L_D^{\tau} R_D^{\sigma}) O_{XYD}^{\mathsf{SPO},\mathrm{inv}} (L_D^{\tau^{-1}} R_D^{\sigma^{-1}}).$$

Proof. For all $\pi \in S_N$ and $x, y \in [N]$, we have

$$\begin{aligned}(L_D^{\tau} R_D^{\sigma}) O_{XYD}^{\mathsf{SPO}} (L_D^{\tau^{-1}} R_D^{\sigma^{-1}}) |x, y, \pi\rangle &= (L_D^{\tau} R_D^{\sigma}) O_{XYD}^{\mathsf{SPO}} |x, y, \tau^{-1}\pi\sigma\rangle \\ &= (L_D^{\tau} R_D^{\sigma}) |x, y \oplus \tau^{-1}\pi\sigma(x), \tau^{-1}\pi\sigma\rangle \\ &= |x, y \oplus \tau^{-1}(\pi(\sigma(x))), \pi\rangle = O_{XYD}^{\mathsf{TSPO}^{\sigma,\tau}} |x, y, \pi\rangle ,\end{aligned}$$

which establishes the first equation. The second one is proved in the same way. □

Clearly, these two operations commute with each other, and they leave the initial state $|\Phi_{\mathsf{SPO}}\rangle_D$ of the oracle invariant:

$$L_\sigma |\Phi_{\mathsf{SPO}}\rangle = R_\sigma |\Phi_{\mathsf{SPO}}\rangle_D = |\Phi_{\mathsf{SPO}}\rangle_D . \tag{4.1}$$

Thus we obtain the following lemma that allows us to compare the behavior of an algorithm when using either the twirled or the ordinary oracle, strengthening Lemma 4.6.

Lemma 4.8. *Let $\mathcal{A}$ be a unitary query algorithm on registers AXY, where X and Y are N-dimensional registers, that gets query access to two oracles that each act on XY. For every $\sigma, \tau \in S_N$, let $|\phi^{\sigma,\tau}\rangle_{AXYD}$ be the joint state of algorithm and oracle defined by running $\mathsf{Init}_D^{\mathsf{SPO}}$ and then $\mathcal{A}^{\mathsf{TSPO}_D^{\sigma,\tau}}$. Moreover, let $|\phi\rangle_{AXYD}$ denote the joint state of algorithm and oracle defined by running $\mathsf{Init}_D^{\mathsf{SPO}}$ and then $\mathcal{A}^{\mathsf{SPO}_D}$. Then,*

$$|\phi^{\sigma,\tau}\rangle_{AXYD} = L_D^{\tau} R_D^{\sigma} |\phi\rangle_{AXYD} .$$

Proof. Without loss of generality, the quantum query algorithm takes the form (cf. Eq. (2.3))

$$U_{AXY}^{(q)} Q_{XYD}^{(q)} U_{AXY}^{(q-1)} Q_{XYD}^{(q-1)} \dots U_{AXY}^{(1)} Q_{XYD}^{(1)} U_{AXY}^{(0)},$$

where each $Q_{XYD}^{(j)}$ is either a forward or an inverse query, and is applied to the initial state $|0\rangle_{AXY} \otimes |\Phi_{\mathsf{SPO}}\rangle_D$. When expressing the twirled oracles in terms of the ordinary ones using Lemma 4.7, we see that the "twirls" $L_D^{\tau} R_D^{\sigma}$ and $L_D^{\tau^{-1}} R_D^{\sigma^{-1}}$ in-between any pair of queries cancel (note that they commute with the unitaries $U_{AXY}^{(j)}$). Moreover, by Eq. (4.1) the initial twirl leaves the initial state $|\Phi_{\mathsf{SPO}}\rangle_D$ of the database invariant. Accordingly, the output state in the two scenarios only differs by an application of $L_D^{\tau} R_D^{\sigma}$, as claimed. □

Lemma 4.7 simulates queries to the twirled superposition oracle (for known σ and τ) by a single query to the ordinary one but requires access to the database. This can also be achieved by acting on the input/output registers X and Y, but in this case more than one query is required. The following lemma shows that an algorithm can always be converted into a "standard form" such that an analogous replacement is possible, at the cost of doubling the number of queries, which will be useful in Section 6.

Lemma 4.9. *Let $\mathcal{A}$ be a unitary query algorithm on registers AXY, where X and Y are N-dimensional registers, that gets query access to two oracles that each act on XY. Then there exists a unitary query algorithm $\mathcal{C}$ on registers BXY, with $B = AZ$ and Z an N-dimensional register, that gets query access to* four *oracles acting on XY such that, for every $\sigma, \tau \in S_N$ (and for any initial state of the database register D), the following three experiments result in the same state of the registers $BXYD$:*

(i) Run $\mathcal{A}^{\mathsf{TSPO}_D^{\sigma,\tau}}$ *and initialize the register* Z *in state* $|0\rangle_Z$*.*

(ii) Run $\mathcal{B}^{\mathsf{TSPO}_D^{\sigma,\tau}}$*, where the query algorithm* $\mathcal{B}$ *gets access to two oracles on* XY *and is defined as follows:*

$$\mathcal{B}^{O_{XY},O_{XY}^{\mathrm{inv}}} := \mathcal{C}^{O_{XY},\,O_{XY},\,O_{XY}^{\mathrm{inv}},\,O_{XY}^{\mathrm{inv}}}.$$

(iii) Run $\mathcal{B}_{\sigma,\tau}^{\mathsf{SPO}_D}$*, where the query algorithm* $\mathcal{B}_{\sigma,\tau}$ *gets access to two oracles on* XY *and is defined as follows:*

$$\mathcal{B}_{\sigma,\tau}^{O_{XY},O_{XY}^{\mathrm{inv}}} := \mathcal{C}^{(V_X^{\sigma^{-1}}V_Y^{\tau^{-1}}O_{XY}V_X^{\sigma}),\,(V_X^{\sigma^{-1}}O_{XY}V_X^{\sigma}V_Y^{\tau}),\,(V_X^{\tau^{-1}}V_Y^{\sigma^{-1}}O_{XY}^{\mathrm{inv}}V_X^{\tau}),\,(V_X^{\tau^{-1}}O_{XY}^{\mathrm{inv}}V_X^{\tau}V_Y^{\sigma})}$$

Moreover, if $\mathcal{A}$ *makes in total* q *oracle queries then* $\mathcal{C}$ *(and hence* $\mathcal{B}$ *and* $\mathcal{B}_{\sigma,\tau}$*) makes in total* $2q$ *oracle queries.*

Proof. Note that

$$\begin{aligned} O_{XYD}^{\mathsf{TSPO}^{\sigma,\tau}}\,|0\rangle_Z &= \mathrm{SWAP}_{Y\leftrightarrow Z}\,O_{XYD}^{\mathsf{TSPO}^{\sigma,\tau}}\;\mathrm{CNOT}_{Y\to Z}\,O_{XYD}^{\mathsf{TSPO}^{\sigma,\tau}}\;\mathrm{SWAP}_{Y\leftrightarrow Z}\,|0\rangle_Z \\ &= \mathrm{SWAP}_{Y\leftrightarrow Z}\Big(V_X^{\sigma^{-1}}O_{XYD}^{\mathsf{SPO}}V_X^{\sigma}V_Y^{\tau}\Big)\,\mathrm{CNOT}_{Y\to Z}\Big(V_X^{\sigma^{-1}}V_Y^{\tau^{-1}}O_{XYD}^{\mathsf{SPO}}V_X^{\sigma}\Big)\,\mathrm{SWAP}_{Y\leftrightarrow Z}\,|0\rangle_Z \end{aligned}$$

and

$$\begin{aligned} O_{XYD}^{\mathsf{TSPO}^{\sigma,\tau},\mathrm{inv}}\,|0\rangle_Z &= \mathrm{SWAP}_{Y\leftrightarrow Z}\,O_{XYD}^{\mathsf{TSPO}^{\sigma,\tau},\mathrm{inv}}\;\mathrm{CNOT}_{Y\to Z}\,O_{XYD}^{\mathsf{TSPO}^{\sigma,\tau},\mathrm{inv}}\;\mathrm{SWAP}_{Y\leftrightarrow Z}\,|0\rangle_Z \\ &= \mathrm{SWAP}_{Y\leftrightarrow Z}\Big(V_X^{\tau^{-1}}O_{XYD}^{\mathsf{SPO},\mathrm{inv}}V_X^{\tau}V_Y^{\sigma}\Big)\,\mathrm{CNOT}_{Y\to Z}\Big(V_X^{\tau^{-1}}V_Y^{\sigma^{-1}}O_{XYD}^{\mathsf{SPO},\mathrm{inv}}V_X^{\tau}\Big)\,\mathrm{SWAP}_{Y\leftrightarrow Z}\,|0\rangle_Z \end{aligned}$$

Thus we see that if we define the query algorithm $\mathcal{C}$ as follows,

$$\begin{aligned} &\mathcal{C}^{O_{XY},O'_{XY},O_{XY}^{\mathrm{inv}},O'^{\mathrm{inv}}_{XY}} := \\ &\qquad \mathcal{A}^{(\mathrm{SWAP}_{Y\leftrightarrow Z}\,O'_{XY}\;\mathrm{CNOT}_{Y\to Z}\,O_{XY}\;\mathrm{SWAP}_{Y\leftrightarrow Z}),\,(\mathrm{SWAP}_{Y\leftrightarrow Z}\,O'^{\mathrm{inv}}_{XY}\;\mathrm{CNOT}_{Y\to Z}\,O_{XY}^{\mathrm{inv}}\;\mathrm{SWAP}_{Y\leftrightarrow Z})}\,|0\rangle_Z\,, \end{aligned}$$

then the claim follows. □

5 The Fundamental Lemma of the Permutation Oracle

We know from the preceding section that the superposition permutation oracles exactly simulate a quantum-accessible permutation, with the permutation being obtained by measuring the database in the computational basis. However, to learn about queries made by the adversary, we wish to also measure whether database registers are in the uniform superposition states. The following result, which we call the *Fundamental Lemma*, shows that this only slightly changes the statistics. It resembles [CFHL21, Corollary 4.2] which goes back to Zhandry, but our result applies to random permutations rather than random functions. We state and prove it for arbitrary relations involving a single input-output pair.

Lemma 5.1 (Fundamental Lemma of the Permutation Oracle)**.** *Let* $R \subseteq [N]\times[N]$ *be a relation. Let* $\mathcal{A}$ *be a quantum algorithm that gets query access to two oracles that each act on* $\mathbb{C}^N \otimes \mathbb{C}^N$*, and which returns a pair* $(x,y) \in [N]\times[N]$*. We consider the following two experiments:*

(i) Sample $\pi \leftarrow S_N$ *uniformly at random, and run* $(x,y) \leftarrow \mathcal{A}^{U^{\pi},U^{\pi^{-1}}}$*.*
If $\pi(x) = y$ *and* $(x,y) \in R$ *then return 1. Otherwise return 0.*

(ii) Sample $\sigma \leftarrow S_N$ *and* $\tau \leftarrow S_N$*, run* $\mathsf{Init}_D^{\mathsf{SPO}}$*, and then* $(x,y) \leftarrow \mathcal{A}^{\mathsf{TSPO}_D^{\sigma,\tau}}$*. Next, apply to register* $D_{\sigma(x)}$ *the projective measurement* $\{|+_{\sigma(x)}\rangle\langle+_{\sigma(x)}|, I - |+_{\sigma(x)}\rangle\langle+_{\sigma(x)}|\}$*. If the second outcome is observed, run* $\pi \leftarrow \mathsf{Rec}_D^{\mathsf{TSPO}^{\sigma,\tau}}$*. If* $\pi(x) = y$ *and* $(x,y) \in R$*, return 1. In all other cases, return 0.*

Let $p_{(i)}$*,* $p_{(ii)}$ *denote the probability that first or second experiment returns 1, respectively. Then:*

$$\sqrt{p_{(i)}} \le \sqrt{p_{(ii)}} + \sqrt{\frac{\ln(N)+1}{N}}.$$

Thus, if we want to upper bound the probability that the algorithm learned a pair (x, y) such that $\pi(x) = y$ satisfying some relation R, then we can just imagine first measuring whether $D_{\sigma(x)}$ is not $|+_{\sigma(x)}\rangle$, without significantly increasing the error – we will see that this typically yields a quantity that is easier to upper bound. This bound is essentially identical to the one known for random functions, except for the extra term $\ln(N) + 1$, which is due to the varying dimensions of the database registers.

The idea of the proof is to recall that, by Lemmas 4.3 and 4.6, the first experiment is exactly simulated by the following:

(i') Sample $\sigma \leftarrow S_N$ and $\tau \leftarrow S_N$, run $\mathsf{Init}_D^{\mathsf{SPO}}$, and then $(x, y) \leftarrow \mathcal{A}^{\mathsf{TSPO}_D^{\sigma,\tau}}$. Measure the entire database in the computational basis and interpret the outcome as a permutation $\pi \in S_N$. If $\pi(\sigma(x)) = \tau(y)$ and $(x, y) \in R$, then return 1. Otherwise return 0.

For comparison, expanding the definition of the recovery routine $\mathsf{Rec}_D^{\mathsf{TSPO}^{\sigma,\tau}}$, the second experiment can be written as follows:

(ii') Sample $\sigma \leftarrow S_N$ and $\tau \leftarrow S_N$, run $\mathsf{Init}_D^{\mathsf{SPO}}$, and then $(x, y) \leftarrow \mathcal{A}^{\mathsf{TSPO}_D^{\sigma,\tau}}$. Next, apply to register $D_{\sigma(x)}$ the projective measurement $\{|+_{\sigma(x)}\rangle\langle +_{\sigma(x)}|, I - |+_{\sigma(x)}\rangle\langle +_{\sigma(x)}|\}$. If the second outcome is observed, measure the entire database in the computational basis and interpret the outcome as a permutation $\pi \in S_N$. If $\pi(\sigma(x)) = \tau(y)$ and $(x, y) \in R$, then return 1. In all other cases, return 0.

Note that (i') and (ii') only differ in that the latter contains the additional measurement of the register $D_{\sigma(x)}$ and subsequent check that the desired (second) outcome occurred. To prove the fundamental lemma, we therefore need to argue that this "postselection" does not impact the probability of acceptance much. We first state and prove a technical lemma that contains the core argument, and then use it establish Lemma 5.1.

Lemma 5.2. *Let $x \in [N]$ and $Y \subseteq [N]$. Then it holds that:*

$$\left\| \sum_{\pi \in S_N : \pi(x) \in Y} |\pi\rangle\langle\pi|_D - \sum_{\pi \in S_N : \pi(x) \in Y} |\pi\rangle\langle\pi|_D (I - |+_x\rangle\langle +_x|_{D_x}) \right\| = \left\| \sum_{\pi \in S_N : \pi(x) \in Y} |\pi\rangle\langle\pi|_D \, |+_x\rangle_{D_x} \right\| \le \sqrt{\frac{|Y|}{x}}.$$

Proof. The first equality is clear. Now, for any operator X the operator norm can be computed as $\|X\| = \max_{\|\phi\|=1} \|X\,|\phi\rangle\|$. Thus there exists a vector $|\Delta\rangle \in \mathcal{H}_{D_{x^c}} = \bigotimes_{k=1:k\neq x}^{n} \mathcal{H}_{D_k}$ such that

$$\begin{aligned} \left\| \sum_{\pi \in S_N : \pi(x) \in Y} |\pi\rangle\langle\pi|_D \, |+_x\rangle_{D_x} \right\|^2 &= \left\| \sum_{\pi \in S_N : \pi(x) \in Y} |\pi\rangle\langle\pi|_D \Big(|+_x\rangle_{D_x} \otimes |\Delta\rangle_{D_{x^c}} \Big) \right\|^2 \\ &= \sum_{\pi \in S_N : \pi(x) \in Y} \left| \langle\pi|_D \left(|+_x\rangle_{D_x} \otimes |\Delta\rangle_{D_{x^c}} \right) \right|^2. \end{aligned}$$

This can be upper bounded by

$$\Pr\big(\pi(x) \in Y \mid t_x \leftarrow [x],\ (t_1, \ldots, t_{x-1}, t_{x+1}, \ldots, t_N) \leftarrow Q\big),$$

where the permutation $\pi \in S_N$ is defined in terms of the numbers $t_1, \ldots, t_N$ via Eq. (3.1), and where Q is some probability distribution on $\prod_{k=1:k\neq x}^{n} [k]$ (namely the distribution obtained by measuring $|\Delta\rangle_{D_{x^c}}$ in the standard basis). Now,

$$\pi(x) \in Y \quad \Leftrightarrow \quad (N\ t_N)\,(N{-}1\ t_{N-1}) \cdots (2\ t_2)\,(1\ t_1)\,(x) \in Y \quad \Leftrightarrow \quad \pi_{>x}(t_x) \in Y \quad \Leftrightarrow \quad t_x \in \pi_{>x}^{-1}(Y),$$

where we recall the notation $\pi_{>x} = (N\ t_N)\,(N{-}1\ t_{N-1}) \cdots (x+1\ t_{x+1})$. Thus, whatever $\pi_{>x}$, there are at most $|Y|$ choices of $t_x \in [x]$ such that $\pi(x) \in Y$. Since $\pi_{>x}$ and t_x are independent and the latter is chosen uniformly at random in $[x]$, we have

$$\Pr\big(\pi(x) \in Y \bigm| t_x \leftarrow [x],\ (t_1, \ldots, t_{x-1}, t_{x+1}, \ldots, t_N) \leftarrow Q\big) \le \frac{|Y|}{x}.$$

This concludes the proof. □

We now prove the fundamental lemma.

Proof of Lemma 5.1. As discussed it suffices to compare the two experiments (i') and (ii'). Let $p(\sigma,\tau,x,y)$ denote the joint distribution of the uniformly random choices of $\sigma,\tau\in S_N$ and the output (x,y) of $\mathcal{A}^{\mathsf{TSPO}_D^{\sigma,\tau}}$ and choose, for each σ,τ,x,y, a purification $|\Delta(\sigma,\tau,x,y)\rangle_{DE}$ of the corresponding state of the database. Then:

$$p_{\text{(i')}} = \sum_{\sigma,\tau\in S_N,(x,y)\in R} p(\sigma,\tau,x,y)\, p_{\text{(i')}}(\sigma,\tau,x,y) \qquad \text{and} \qquad p_{\text{(ii')}} = \sum_{\sigma,\tau\in S_N,(x,y)\in R} p(\sigma,\tau,x,y)\, p_{\text{(ii')}}(\sigma,\tau,x,y),$$

where

$$\begin{aligned}
p_{\text{(i')}}(\sigma,\tau,x,y) &:= \left\| \sum_{\pi\in S_N:\pi(\sigma(x))=\tau(y)} |\pi\rangle\langle\pi|_D\, |\Delta(\sigma,\tau,x,y)\rangle_{DE} \right\|^2, \\
p_{\text{(ii')}}(\sigma,\tau,x,y) &:= \left\| \sum_{\pi\in S_N:\pi(\sigma(x))=\tau(y)} |\pi\rangle\langle\pi|_D \Big(I - |+_{\sigma(x)}\rangle\langle+_{\sigma(x)}|_{D_{\sigma(x)}}\Big)\, |\Delta(\sigma,\tau,x,y)\rangle_{DE} \right\|^2.
\end{aligned}$$

Using Lemma 5.2 (with $Y=\{\tau(y)\}$) and the Cauchy-Schwarz inequality, it follows that

$$\begin{aligned}
p_{\text{(i')}} &\le \sum_{\sigma,\tau\in S_N,(x,y)\in R} p(\sigma,\tau,x,y)\left(\sqrt{p_{\text{(ii')}}(\sigma,\tau,x,y)} + \sqrt{\frac{1}{\sigma(x)}}\right)^2 \\
&= p_{\text{(ii')}} + 2\sum_{\sigma,\tau\in S_N,(x,y)\in R} p(\sigma,\tau,x,y)\sqrt{p_{\text{(ii')}}(\sigma,\tau,x,y)}\sqrt{\frac{1}{\sigma(x)}} + \sum_{\sigma,\tau\in S_N,(x,y)\in R} p(\sigma,\tau,x,y)\frac{1}{\sigma(x)} \\
&\le p_{\text{(ii')}} + 2\sqrt{p_{\text{(ii')}}}\sqrt{\sum_{\sigma,\tau\in S_N,(x,y)\in R} p(\sigma,\tau,x,y)\frac{1}{\sigma(x)}} + \sum_{\sigma,\tau\in S_N,(x,y)\in R} p(\sigma,\tau,x,y)\frac{1}{\sigma(x)}.
\end{aligned}$$

Thus,

$$\sqrt{p_{\text{(i')}}} \le \sqrt{p_{\text{(ii')}}} + \sqrt{\sum_{\sigma,\tau\in S_N,(x,y)\in R} p(\sigma,\tau,x,y)\frac{1}{\sigma(x)}}.$$

Finally, we note that as a consequence of Lemma 4.6, σ, τ, and (x,y) are independent with respect to the distribution $p(\sigma,\tau,x,y)$. Thus,

$$\sum_{\sigma,\tau\in S_N,(x,y)\in R} p(\sigma,\tau,x,y)\frac{1}{\sigma(x)} \le \frac{1}{N}\sum_{k=1}^{N}\frac{1}{k} \le \frac{\ln(N)+1}{N}.$$

□

In order to apply the fundamental lemma, it is useful to upper bound the probability $p_{\text{(ii)}}$ in way that only refers to the state of the database. This is achieved by the following lemma.

Lemma 5.3. *Let $R\subseteq[N]\times[N]$ be a relation. Let $\mathcal{A}$ be a quantum algorithm that gets query access to two oracles that each act on $\mathbb{C}^N\otimes\mathbb{C}^N$, and which returns a pair $(x,y)\in[N]\times[N]$. For $\sigma,\tau\in S_N$, let $|\phi^{\sigma,\tau}\rangle$ denote a purification of the state of the database after running $\mathsf{Init}_D^{\mathsf{SPO}}$ and then $\mathcal{A}^{\mathsf{TSPO}_D^{\sigma,\tau}}$. Then, the quantity $p_{(ii)}$ in Lemma 5.1 can be upper bounded as*

$$p_{(ii)} \le \mathop{\mathbf{E}}_{\sigma,\tau\leftarrow S_N} \sum_{(x,y)\in R} \sum_{\substack{\pi\in S_N \text{ s.th.} \\ \tau^{-1}(\pi(\sigma(x)))=y}} \left\| \langle\pi|_D \left(I - |+_{\sigma(x)}\rangle\langle+_{\sigma(x)}|_{D_{\sigma(x)}}\right) |\phi^{\sigma,\tau}\rangle \right\|^2.$$

Proof. Without loss of generality we can assume that $\mathcal{A}$ is a unitary query algorithm on registers AXY such that the classical outcomes x and y can be obtained by measuring the X and Y registers. For every $\sigma, \tau \in S_N$, let $|\phi^{\sigma,\tau}\rangle_{AXYD}$ be the joint state of algorithm and oracle defined by running $\mathsf{Init}_D^{\mathsf{SPO}}$ and then $\mathcal{A}^{\mathsf{TSPO}_D^{\sigma,\tau}}$. Then:

$$p_{\text{(ii)}} = \mathop{\mathbf{E}}_{\sigma,\tau \leftarrow S_N} \sum_{(x,y)\in R} \sum_{\substack{\pi \in S_N \text{ s.th.} \\ \tau^{-1}(\pi(\sigma(x)))=y}} \left\| \langle \pi |_D \left(I - |+_{\sigma(x)}\rangle\langle +_{\sigma(x)}|_{D_{\sigma(x)}} \right) \langle xy|_{XY} \, |\phi^{\sigma,\tau}\rangle_{AXYD} \right\|^2.$$

Because the projection $\langle xy|$ commutes with the operators on D and never increases the norm, we can upper bound the above as

$$p_{\text{(ii)}} \leq \mathop{\mathbf{E}}_{\sigma,\tau \leftarrow S_N} \sum_{(x,y)\in R} \sum_{\substack{\pi \in S_N \text{ s.th.} \\ \tau^{-1}(\pi(\sigma(x)))=y}} \left\| \langle \pi |_D \left(I - |+_{\sigma(x)}\rangle\langle +_{\sigma(x)}|_{D_{\sigma(x)}} \right) |\phi^{\sigma,\tau}\rangle_{AXYD} \right\|^2.$$

Since this expression only depends on the reduced state on D, the claim follows. □

6 Bounding the Success Probability for Search

In this section we generalize Zhandry's compressed oracle technique [Zha19] to the case of random permutations, but using our (twirled) permutation oracle. The high-level strategy for proving such theorems is as follows: For the compressed oracle for a random function H, it can be approximately determined whether a certain input x has been queried, and if so whether x, together with the corresponding function output $y = H(x)$, fulfils a certain relation. This test is performed by a measurement acting on the database register D_x only. For different inputs x, these tests commute, and hence there exists a projective measurement answering the question whether there *exists* an input x such that the described test would trigger. The probability that this measurement outputs yes after q queries then serves as a convenient *progress measure*. This is reminiscent to the progress measures used in the so-called "hybrid method" that was introduced earlier to prove the query lower bound for the unstructured search problem [BBBV97].

We would like to generalize this technique to the case of random permutations. We follow a similar strategy, and begin by devising a generalization of the test for a single input. To test whether a certain input x has been queried in forward direction or output in an inverse query, and if so, whether, together with the corresponding function output $y = \pi(x)$, it fulfils a certain relation $R \subseteq [N] \times [N]$, we define the following operators:

$$\Pi_D^{R,x} := \sum_{\pi \in S_N : \pi(x) \in R_x} |\pi\rangle\langle\pi|_D, \qquad E_D^{R,x} := \Pi_D^{R,x}(I - |+_x\rangle\langle +_x|_{D_x}).$$

where we have defined $R_x := \{y \in [N] : (x,y) \in R\}$. For later use, we also set $R_y^{\text{inv}} := \{x \in [N] : (x,y) \in R\}$ and

$$r_{\max} := \max\left\{ \max_{x\in[N]} |R_x|, \max_{y \in [N]} |R_y^{\text{inv}}| \right\}.$$

Unfortunately, the operators $E^{R,x}$ for different x do not commute, so we cannot simply use these to construct a measurement answering the existence question. Instead, we will check whether a *random* x has this property. To lift the worst-case bounds established in the previous section to the average case, we further consider running the query algorithm with the *twirled* permutation oracle for uniformly random $\sigma, \tau \in S_N$. Because the permutation in the database is now twirled as compared to the action of the oracle, we must also consider the twirled relation $R^{\sigma,\tau}$ that is defined as follows in terms of R:

$$(x,y) \in R^{\sigma,\tau} :\Longleftrightarrow (\sigma^{-1}(x), \tau^{-1}(y)) \in R. \tag{6.1}$$

Thus we are led to consider the following natural progress measure:

$$\mathop{\mathbf{E}}_{\substack{x \leftarrow [N], \\ \sigma,\tau \leftarrow S_N}} \left\| E_D^{R^{\sigma,\tau},x} |\phi^{\sigma,\tau}\rangle \right\|^2,$$

where $|\phi^{\sigma,\tau}\rangle$ denotes the joint state of (a unitary realization of) the algorithm and database obtained by running the algorithm with the twirled permutation oracle $\mathsf{TSPO}^{\sigma,\tau}$. Remarkably, this progress measure not only has an intuitive operational interpretation, but it is also directly related to the upper bound furnished by the fundamental lemma (cf. Lemma 5.3):

Lemma 6.1. *For any relation $R \subseteq [N] \times [N]$, it holds that*

$$N \underset{\substack{x \leftarrow [N], \\ \sigma,\tau \leftarrow S_N}}{\mathbf{E}} \left\| E_D^{R^{\sigma,\tau},x} |\phi^{\sigma,\tau}\rangle \right\|^2 = \underset{\sigma,\tau \leftarrow S_N}{\mathbf{E}} \sum_{(x,y)\in R} \sum_{\substack{\pi \in S_N \text{ s.th.} \\ \tau^{-1}(\pi(\sigma(x)))=y}} \left\| \langle \pi|_D \left(I - |+_{\sigma(x)}\rangle\langle +_{\sigma(x)}|_{D_{\sigma(x)}} \right) |\phi^{\sigma,\tau}\rangle \right\|^2.$$

Proof. We calculate:

$$\begin{aligned}
&\underset{\sigma,\tau \leftarrow S_N}{\mathbf{E}} \sum_{(x,y)\in R} \sum_{\substack{\pi \in S_N \text{ s.th.} \\ \tau^{-1}(\pi(\sigma(x)))=y}} \left\| \langle \pi|_D \left(I - |+_{\sigma(x)}\rangle\langle +_{\sigma(x)}|_{D_{\sigma(x)}} \right) |\phi^{\sigma,\tau}\rangle \right\|^2 \\
&= \underset{\sigma,\tau \leftarrow S_N}{\mathbf{E}} \sum_{x=1}^{N} \sum_{\substack{\pi \in S_N \text{ s.th.} \\ (\sigma(x),\pi(\sigma(x)))\in R^{\sigma,\tau}}} \left\| \langle \pi|_D \left(I - |+_{\sigma(x)}\rangle\langle +_{\sigma(x)}|_{D_{\sigma(x)}} \right) |\phi^{\sigma,\tau}\rangle \right\|^2 \\
&= \underset{\sigma,\tau \leftarrow S_N}{\mathbf{E}} \sum_{x=1}^{N} \sum_{\substack{\pi \in S_N \text{ s.th.} \\ (x,\pi(x))\in R^{\sigma,\tau}}} \left\| \langle \pi|_D \left(I - |+_{x}\rangle\langle +_{x}|_{D_{x}} \right) |\phi^{\sigma,\tau}\rangle \right\|^2 \\
&= N \underset{\substack{x \leftarrow [N], \\ \sigma,\tau \leftarrow S_N}}{\mathbf{E}} \left\| E_D^{R^{\sigma,\tau},x} |\phi^{\sigma,\tau}\rangle \right\|^2.
\end{aligned}$$

□

We can now summarize the plan for the remainder of this section:

(i) We bound the effect of a single query on any fixed location x of the database in Section 6.1.
(ii) We derive a bound on the success probability of the adversary in Section 6.2, but in expectation over the random choice of the database location and the choice of the twirling.
(iii) We bound the (weighted) average probability that a database register is no longer in the uniform state, which is necessary to complete the proof, in Section 6.3.
(iv) We combine (ii) and (iii) with the fundamental lemma to obtain our main theorem in Section 6.4.

6.1 Bounding the Success Probability in the Worst Case

We start with a useful lemma that follows readily from Lemma 5.2.

Lemma 6.2. *Let $Q_{XYD}^{\mathsf{SPO}} \in \{O_{XYD}^{\mathsf{SPO}}, O_{XYD}^{\mathsf{SPO},\mathrm{inv}}\}$, and $x \in [N]$. Then:*

(i) $\left\| E_D^{R,x} Q_{XYD}^{\mathsf{SPO}} (I - \Pi_D^{R,x}) \right\| \leq \sqrt{\frac{|R_x|}{x}}$.

(ii) $\left\| E_D^{R,x} Q_{XYD}^{\mathsf{SPO}} |\phi\rangle \right\| - \left\| E_D^{R,x} |\phi\rangle \right\| \leq \sqrt{\frac{|R_x|}{x}} \left\| (I - |+_x\rangle\langle +_x|_{D_x}) |\phi\rangle \right\| + \left\| E_D^{R,x} Q_{XYD}^{\mathsf{SPO}} |+_x\rangle\langle +_x|_{D_x} |\phi\rangle \right\|$ *for any pure state* $|\phi\rangle_{AXYD}$.

(iii) $\left\| E_D^{R,x} Q_{YD}^{\mathsf{SPO},z} |+_x\rangle\langle +_x|_{D_x} \right\| \leq 2\sqrt{\frac{|R_x|}{x}}$ *for all $z \in [N]$, where* $Q_{YD}^{\mathsf{SPO},z} := \langle z|_X \, Q_{XYD}^{\mathsf{SPO}} \, |z\rangle_X$.

Proof. (i) Since Q_{XYD}^{SPO} is a unitary controlled on register D, it commutes with computational basis projections, and hence with $\Pi_D^{R,x}$. Therefore, and using unitary invariance, we obtain

$$\begin{aligned}
\left\| E_D^{R,x} Q_{XYD}^{\mathsf{SPO}} (I - \Pi_D^{R,x}) \right\| &= \left\| E_D^{R,x} (I - \Pi_D^{R,x}) Q_{XYD}^{\mathsf{SPO}} \right\| \\
&= \left\| \Pi_D^{R,x} (I - |+_x\rangle\langle +_x|_{D_x}) (I - \Pi_D^{R,x}) \right\| \\
&= \left\| \Pi_D^{R,x} |+_x\rangle\langle +_x|_{D_x} (I - \Pi_D^{R,x}) \right\|
\end{aligned}$$

$$\leq \left\| \Pi_D^{R,x} \left|+_x\right\rangle_{D_x} \right\| \leq \sqrt{\frac{|R_x|}{x}}.$$

The last inequality holds due to Lemma 5.2.

(ii) Since

$$E_D^{R,x} Q_{XYD}^{\mathsf{SPO}} \left|\phi\right\rangle = E_D^{R,x} Q_{XYD}^{\mathsf{SPO}} E_D^{R,x} \left|\phi\right\rangle + E_D^{R,x} Q_{XYD}^{\mathsf{SPO}} \left(I - E_D^{R,x}\right) \left|\phi\right\rangle,$$

and $\|E_D^{R,x} Q_{XYD}^{\mathsf{SPO}} E_D^{R,x} \left|\phi\right\rangle\| \leq \|E_D^{R,x} \left|\phi\right\rangle\|$, we have, by the triangle inequality,

$$\left\| E_D^{R,x} Q_{XYD}^{\mathsf{SPO}} \left|\phi\right\rangle \right\| - \left\| E_D^{R,x} \left|\phi\right\rangle \right\| \leq \left\| E_D^{R,x} Q_{XYD}^{\mathsf{SPO}} \left(I - E_D^{R,x}\right) \left|\phi\right\rangle \right\|.$$

Since $I - E_D^{R,x} = (I - \Pi_D^{R,x})(I - |+_x\rangle\langle +_x|_{D_x}) + |+_x\rangle\langle +_x|_{D_x}$, we can bound the right-hand side by

$$\begin{aligned} &\left\| E_D^{R,x} Q_{XYD}^{\mathsf{SPO}} \left(I - E_D^{R,x}\right) \left|\phi\right\rangle \right\| \\ &\leq \left\| E_D^{R,x} Q_{XYD}^{\mathsf{SPO}} (I - \Pi_D^{R,x})(I - |+_x\rangle\langle +_x|_{D_x}) \left|\phi\right\rangle \right\| + \left\| E_D^{R,x} Q_{XYD}^{\mathsf{SPO}} |+_x\rangle\langle +_x|_{D_x} \left|\phi\right\rangle \right\| \\ &\leq \sqrt{\frac{|R_x|}{x}} \left\| (I - |+_x\rangle\langle +_x|_{D_x}) \left|\phi\right\rangle \right\| + \left\| E_D^{R,x} Q_{XYD}^{\mathsf{SPO}} |+_x\rangle\langle +_x|_{D_x} \left|\phi\right\rangle \right\|, \end{aligned}$$

using another triangle inequality and, in the last step, part (i).

(iii) We can proceed similarly as in part (i):

$$\begin{aligned} \left\| E_D^{R,x} Q_{YD}^{\mathsf{SPO},z} |+_x\rangle\langle +_x|_{D_x} \right\| &= \left\| \Pi_D^{R,x} (I - |+_x\rangle\langle +_x|_{D_x}) Q_{YD}^{\mathsf{SPO},z} |+_x\rangle\langle +_x|_{D_x} \right\| \\ &\leq \left\| \Pi_D^{R,x} Q_{YD}^{\mathsf{SPO},z} |+_x\rangle\langle +_x|_{D_x} \right\| + \left\| \Pi_D^{R,x} |+_x\rangle\langle +_x|_{D_x} Q_{YD}^{\mathsf{SPO},z} |+_x\rangle\langle +_x|_{D_x} \right\| \\ &= \left\| Q_{YD}^{\mathsf{SPO},z} \Pi_D^{R,x} |+_x\rangle\langle +_x|_{D_x} \right\| + \left\| \Pi_D^{R,x} |+_x\rangle\langle +_x|_{D_x} Q_{YD}^{\mathsf{SPO},z} |+_x\rangle\langle +_x|_{D_x} \right\| \\ &\leq 2 \| \Pi_D^{R,x} \left|+_x\right\rangle_{D_x} \| \leq 2\sqrt{\frac{|R_x|}{x}}. \end{aligned}$$

□

We now bound the effect of a single forward query on the probability amplitude.

Lemma 6.3 (Forward query). *For any pure state $\left|\phi\right\rangle_{AXYD}$ such that $\|\langle \pi|_D \left|\phi\right\rangle_{AXYD}\|^2 = \frac{1}{N!}$ for all $\pi \in S_N$, and for any $x \in [N]$, we have*

$$\left\| E_D^{R,x} O_{XYD}^{\mathsf{SPO}} \left|\phi\right\rangle \right\| - \left\| E_D^{R,x} \left|\phi\right\rangle \right\| \leq \sqrt{\frac{|R_x|}{x}} \left\| (I - |+_x\rangle\langle +_x|_{D_x}) \left|\phi\right\rangle \right\| + 2\sqrt{\zeta_{|\phi\rangle,x}}$$

where $\left|\phi_x\right\rangle_{AYD} := \langle x|_X \left|\phi\right\rangle_{AXYD}$ and

$$\zeta_{|\phi\rangle,x} := \frac{|R_x|}{x} \| \left|\phi_x\right\rangle \|^2 + \frac{|R_x|}{x^2 N} + \frac{1}{x} \sum_{z=1}^{x-1} \sum_{\pi_{x^c} : \pi_{x^c}(z) \in R_x} \left\| \langle \pi_{x^c}|_{D_{x^c}} \left|\phi_z\right\rangle \right\|^2$$

Proof. By Lemma 6.2 (ii), we have

$$\left\| E_D^{R,x} O_{XYD}^{\mathsf{SPO}} \left|\phi\right\rangle \right\| - \| E_D^{R,x} \left|\phi\right\rangle \| \leq \sqrt{\frac{|R_x|}{x}} \left\| (I - |+_x\rangle\langle +_x|_{D_x}) \left|\phi\right\rangle \right\| + \| E_D^{R,x} O_{XYD}^{\mathsf{SPO}} |+_x\rangle\langle +_x|_{D_x} \left|\phi\right\rangle \|, \tag{6.2}$$

To upper bound the right-hand side norm, we compute its square as follows:

$$\left\| E_D^{R,x} O_{XYD}^{\mathsf{SPO}} |+_x\rangle\langle +_x|_{D_x} \left|\phi\right\rangle \right\|^2 = \sum_{z \in [N]} \left\| \langle z|_X \, E_D^{R,x} O_{XYD}^{\mathsf{SPO}} |+_x\rangle\langle +_x|_{D_x} \left|\phi\right\rangle \right\|^2 = \sum_{z \in [N]} \left\| E_D^{R,x} O_{YD}^{\mathsf{SPO},z} |+_x\rangle\langle +_x|_{D_x} \left|\phi_z\right\rangle \right\|^2,$$

where we defined $O_{YD}^{\mathsf{SPO},z} := \langle z|_X \, O_{XYD}^{\mathsf{SPO}} \left|z\right\rangle_X$. We analyze the summands and distinguish three cases:

(i) For $z > x$, $O_{YD}^{\mathsf{SPO},z}$ acts trivially on D_x by Lemma 4.4, and hence commutes with $|+_x\rangle\langle +_x|_{D_x}$. As $E_D^{R,x}|+_x\rangle\langle +_x|_{D_x} = 0$, we see that the corresponding summands vanish.

(ii) For $z = x$, then we have the following bound from Lemma 6.2 (iii):

$$\left\| E_D^{R,x} O_{YD}^{\mathsf{SPO},x} |+_x\rangle\langle +_x|_{D_x} |\phi_x\rangle \right\|^2 \leq 4\frac{|R_x|}{x} \, \| |\phi_x\rangle \|^2.$$

(iii) For $z < x$, the argument is more involved. We begin by computing the action of $O_{YD}^{\mathsf{SPO},z}$ on computational basis states. For $\pi \in S_N$, let us write $\pi = \pi_{>x}\,(x\;t)\,\pi_{<x}$ as in Section 3.2, with $t \in [x]$, and define $\pi_{x^c} := \pi_{>x}\pi_{<x}$. We may identify π_{x^c} with the indices t_k for $k \neq x$; accordingly we shall write $|\pi\rangle_D = |t\rangle_{D_x} |\pi_{x^c}\rangle_{D_{x^c}}$. Note that if $\pi_{<x}(z) = t$ then $\pi(z) = \pi_{>x}(x)$, while otherwise $\pi(z) = \pi_{x^c}(z)$. Thus, for every $\pi \in S_N$ and $y \in [N]$, we have

$$O_{YD}^{\mathsf{SPO},z} |y, \pi\rangle_{YD} = O_{YD}^{\mathsf{SPO},z} |y, t, \pi_{x^c}\rangle_{YD_xD_{x^c}} = \begin{cases} |y \oplus \pi_{>x}(x), t, \pi_{x^c}\rangle_{YD_xD_{x^c}} & \text{if } \pi_{<x}(z) = t, \\ |y \oplus \pi_{x^c}(z), t, \pi_{x^c}\rangle_{YD_xD_{x^c}} & \text{otherwise.} \end{cases}$$

It follows that

$$\begin{aligned} O_{YD}^{\mathsf{SPO},z} |y\rangle_Y |+_x\rangle_{D_x} |\pi_{x^c}\rangle_{D_{x^c}} &= |y \oplus \pi_{x^c}(z)\rangle_Y |+_x\rangle_{D_x} |\pi_{x^c}\rangle_{D_{x^c}} \\ &\quad + \frac{1}{\sqrt{x}} \Big(|y \oplus \pi_{>x}(x)\rangle_Y - |y \oplus \pi_{x^c}(z)\rangle_Y \Big) |\pi_{<x}(z)\rangle_{D_x} |\pi_{x^c}\rangle_{D_{x^c}}, \end{aligned}$$

hence

$$\begin{aligned} (I - |+_x\rangle\langle +_x|_{D_x}) O_{YD}^{\mathsf{SPO},z} |y\rangle_Y |+_x\rangle_{D_x} |\pi_{x^c}\rangle_{D_{x^c}} &= \frac{1}{\sqrt{x}} \Big(|y \oplus \pi_{>x}(x)\rangle_Y - |y \oplus \pi_{x^c}(z)\rangle_Y \Big) \\ &\quad \otimes \left(|\pi_{<x}(z)\rangle_{D_x} - \frac{1}{\sqrt{x}} |+_x\rangle_{D_x} \right) \otimes |\pi_{x^c}\rangle_{D_{x^c}} \end{aligned}$$

and finally

$$\begin{aligned} &E_D^{R,x} O_{YD}^{\mathsf{SPO},z} |y\rangle_Y |+_x\rangle_{D_x} |\pi_{x^c}\rangle_{D_{x^c}} \\ &= \frac{1}{\sqrt{x}} \Big(|y \oplus \pi_{>x}(x)\rangle_Y - |y \oplus \pi_{x^c}(z)\rangle_Y \Big) \otimes \Bigg(\mathbf{1}_{\pi_{x^c}(z) \in R_x} |\pi_{<x}(z)\rangle_{D_x} - \frac{1}{x} \sum_{t \in [x] \cap \pi_{>x}^{-1}(R_x)} |t\rangle_{D_x} \Bigg) \otimes |\pi_{x^c}\rangle_{D_{x^c}} \\ &= M_{YD_{x^c}}^{(z)} \left(\frac{1}{\sqrt{x}} |y\rangle_Y \otimes \Bigg(\mathbf{1}_{\pi_{x^c}(z) \in R_x} |\pi_{<x}(z)\rangle_{D_x} - \frac{1}{x} \sum_{t \in [x] \cap \pi_{>x}^{-1}(R_x)} |t\rangle_{D_x} \Bigg) \otimes |\pi_{x^c}\rangle_{D_{x^c}} \right), \end{aligned}$$

where the operator $M_{YD_{x^c}}^{(z)}$ is defined by $M_{YD_{x^c}}^{(z)} |y\rangle_Y |\pi_{x^c}\rangle_{D_{x^c}} = (|y \oplus \pi_{>x}(x)\rangle_Y - |y \oplus \pi_{x^c}(z)\rangle_Y) |\pi_{x^c}\rangle_{D_{x^c}}$ and has operator norm $\leq \sqrt{2}$. Thus:

$$E_D^{R,x} O_{YD}^{\mathsf{SPO},z} |+_x\rangle_{D_x} |\pi_{x^c}\rangle_{D_{x^c}} = M_{YD_{x^c}}^{(z)} \frac{1}{\sqrt{x}} \Bigg(\mathbf{1}_{\pi_{x^c}(z) \in R_x} |\pi_{<x}(z)\rangle_{D_x} - \frac{1}{x} \sum_{t \in [x] \cap \pi_{>x}^{-1}(R_x)} |t\rangle_{D_x} \Bigg) |\pi_{x^c}\rangle_{D_{x^c}}.$$

We can now bound the desired norm:

$$\begin{aligned} &\left\| E_D^{R,x} O_{YD}^{\mathsf{SPO},z} |+_x\rangle\langle +_x|_{D_x} |\phi_z\rangle \right\|^2 \\ &= \left\| E_D^{R,x} O_{YD}^{\mathsf{SPO},z} |+_x\rangle\langle +_x|_{D_x} \sum_{\pi_{x^c}} |\pi_{x^c}\rangle\langle \pi_{x^c}|_{D_{x^c}} |\phi_z\rangle \right\|^2 \\ &= \left\| \sum_{\pi_{x^c}} E_D^{R,x} O_{YD}^{\mathsf{SPO},z} |+_x\rangle_{D_x} |\pi_{x^c}\rangle_{D_{x^c}} \langle +_x|_{D_x} \langle \pi_{x^c}|_{D_{x^c}} |\phi_z\rangle \right\|^2 \\ &= \left\| M_{YD_{x^c}}^{(z)} \sum_{\pi_{x^c}} \frac{1}{\sqrt{x}} \Bigg(\mathbf{1}_{\pi_{x^c}(z) \in R_x} |\pi_{<x}(z)\rangle_{D_x} - \frac{1}{x} \sum_{t \in [x] \cap \pi_{>x}^{-1}(R_x)} |t\rangle_{D_x} \Bigg) |\pi_{x^c}\rangle_{D_{x^c}} \langle +_x|_{D_x} \langle \pi_{x^c}|_{D_{x^c}} |\phi_z\rangle \right\|^2 \end{aligned}$$

$$
\begin{aligned}
&\le \frac{2}{x}\sum_{\pi_{x^c}}\Big\|\Big(\mathbf{1}_{\pi_{x^c}(z)\in R_x}\,|\pi_{<x}(z)\rangle_{D_x} - \frac{1}{x}\sum_{t\in[x]\cap\pi_{>x}^{-1}(R_x)}|t\rangle_{D_x}\Big)\,\langle +_x|_{D_x}\,\langle\pi_{x^c}|_{D_{x^c}}\,|\phi_z\rangle\Big\|^2\\
&\le \frac{4}{x}\sum_{\pi_{x^c}}\bigg(\mathbf{1}_{\pi_{x^c}(z)\in R_x}\Big\|\langle +_x|_{D_x}\,\langle\pi_{x^c}|_{D_{x^c}}\,|\phi_z\rangle\Big\|^2+\frac{1}{x^2}\Big\|\sum_{t\in[x]\cap\pi_{>x}^{-1}(R_x)}|t\rangle_{D_x}\,\langle +_x|_{D_x}\,\langle\pi_{x^c}|_{D_{x^c}}\,|\phi_z\rangle\Big\|^2\bigg)\\
&= \frac{4}{x}\sum_{\pi_{x^c}:\pi_{x^c}(z)\in R_x}\Big\|\langle +_x|_{D_x}\,\langle\pi_{x^c}|_{D_{x^c}}\,|\phi_z\rangle\Big\|^2+\frac{4}{x^3}\sum_{\pi_{x^c}}\sum_{t\in[x]\cap\pi_{>x}^{-1}(R_x)}\big\|\langle +_x|_{D_x}\,\langle\pi_{x^c}|_{D_{x^c}}\,|\phi_z\rangle\big\|^2\\
&\le \frac{4}{x}\sum_{\pi_{x^c}:\pi_{x^c}(z)\in R_x}\Big\|\langle\pi_{x^c}|_{D_{x^c}}\,|\phi_z\rangle\Big\|^2+\frac{4}{x^3}\sum_{\pi_{x^c}}\sum_{t\in[x]\cap\pi_{>x}^{-1}(R_x)}\big\|\langle\pi_{x^c}|_{D_{x^c}}\,|\phi_z\rangle\big\|^2.
\end{aligned}
$$

Here the first inequality follows from the bound on the operator norm of $M^{(z)}_{YD_{x^c}}$, and the fact that all π_{x^c} are orthogonal, the second inequality follows by Cauchy-Schwarz, and the last inequality follows by the fact that $|+_x\rangle$ is a unit vector. It follows that

$$
\begin{aligned}
&\sum_{z=1}^{x-1}\Big\|E_D^{R,x}O_{YD}^{\mathsf{SPO},z}|+_x\rangle\langle +_x|_{D_x}\,|\phi_z\rangle\Big\|^2\\
&\le \frac{4}{x}\sum_{z=1}^{x-1}\sum_{\pi_{x^c}:\pi_{x^c}(z)\in R_x}\Big\|\langle\pi_{x^c}|_{D_{x^c}}\,|\phi_z\rangle\Big\|^2+\frac{4}{x^3}\sum_{z=1}^{x-1}\sum_{\pi_{x^c}}\sum_{t\in[x]\cap\pi_{>x}^{-1}(R_x)}\big\|\langle\pi_{x^c}|_{D_{x^c}}\,|\phi_z\rangle\big\|^2\\
&\le \frac{4}{x}\sum_{z=1}^{x-1}\sum_{\pi_{x^c}:\pi_{x^c}(z)\in R_x}\Big\|\langle\pi_{x^c}|_{D_{x^c}}\,|\phi_z\rangle\Big\|^2+\frac{4}{x^3}\sum_{\pi_{x^c}}\sum_{t\in[x]\cap\pi_{>x}^{-1}(R_x)}\big\|\langle\pi_{x^c}|_{D_{x^c}}\,|\phi\rangle\big\|^2\\
&= \frac{4}{x}\sum_{z=1}^{x-1}\sum_{\pi_{x^c}:\pi_{x^c}(z)\in R_x}\Big\|\langle\pi_{x^c}|_{D_{x^c}}\,|\phi_z\rangle\Big\|^2+\frac{4}{x^2}\sum_{\pi_{x^c}}\sum_{t\in[x]\cap\pi_{>x}^{-1}(R_x)}\frac{1}{N!},
\end{aligned}
$$

where we used that $\|\langle\pi|_D\,|\phi\rangle\|^2=\frac{1}{N!}$ for all $\pi\in S_N$ and hence $\|\langle\pi_{x^c}|_{D_{x^c}}\,|\phi\rangle\|^2=\frac{x}{N!}$. Since

$$
\sum_{\pi_{x^c}}\sum_{t\in[x]\cap\pi_{>x}^{-1}(R_x)}\frac{1}{N!}=\frac{1}{N!}\sum_{\pi\in S_N:\pi(x)\in R_x}1=\Pr\big(\pi(x)\in R_x\mid\pi\leftarrow S_N\big)\le\frac{|R_x|}{N},
$$

we find that

$$
\sum_{z=1}^{x-1}\Big\|E_D^{R,x}O_{YD}^{\mathsf{SPO},z}|+_x\rangle\langle +_x|_{D_x}\,|\phi_z\rangle\Big\|^2\le\frac{4}{x}\sum_{z=1}^{x-1}\sum_{\pi_{x^c}:\pi_{x^c}(z)\in R_x}\Big\|\langle\pi_{x^c}|_{D_{x^c}}\,|\phi_z\rangle\Big\|^2+\frac{4}{x^2}\frac{|R_x|}{N}.
$$

Altogether, we obtain the following bound for the right-hand side term in Eq. (6.2):

$$
\begin{aligned}
\Big\|E_D^{R,x}O_{XYD}^{\mathsf{SPO}}|+_x\rangle\langle +_x|_{D_x}\,|\phi\rangle\Big\|&\le\sqrt{4\frac{|R_x|}{x}\,\||\phi_x\rangle\|^2+\frac{4}{x}\sum_{z=1}^{x-1}\sum_{\pi_{x^c}:\pi_{x^c}(z)\in R_x}\Big\|\langle\pi_{x^c}|_{D_{x^c}}\,|\phi_z\rangle\Big\|^2+\frac{4}{x^2}\frac{|R_x|}{N}}\\
&=2\sqrt{\frac{|R_x|}{x}\,\||\phi_x\rangle\|^2+\frac{|R_x|}{x^2N}+\frac{1}{x}\sum_{z=1}^{x-1}\sum_{\pi_{x^c}:\pi_{x^c}(z)\in R_x}\Big\|\langle\pi_{x^c}|_{D_{x^c}}\,|\phi_z\rangle\Big\|^2}
\end{aligned}
$$

and the lemma follows. □

Next, we now bound the effect of an inverse query on the probability amplitude.

Lemma 6.4 (Inverse query)**.** *For any pure state $|\phi\rangle_{AXYD}$ such that $\|\langle\pi|_D\,|\phi\rangle_{AXYD}\|^2=\frac{1}{N!}$, and for any $x\in[N]$, we have*

$$
\Big\|E_D^{R,x}O_{XYD}^{\mathsf{SPO},\mathrm{inv}}\,|\phi\rangle\Big\|-\Big\|E_D^{R,x}\,|\phi\rangle\Big\|\le\sqrt{\frac{|R_x|}{x}}\big\|\big(I-|+_x\rangle\langle +_x|_{D_x}\big)\,|\phi\rangle\big\|+4\sqrt{\zeta^{\mathrm{inv}}_{|\phi\rangle,x}},
$$

where $|\phi_x\rangle_{AYD} := \langle x|_X \, |\phi\rangle_{AXYD}$ *and*

$$\zeta^{\mathrm{inv}}_{|\phi\rangle,x} := \frac{|R_x|}{x} \, \| |\phi_x\rangle \|^2 + \frac{|R_x|}{x^2 N} + \frac{1}{x} \sum_{z \in R_x} \sum_{\pi_{x^c} : \pi^{-1}_{>x}(z) < x} \left\| \langle \pi_{x^c}|_{D_{x^c}} \, |\phi_z\rangle \right\|^2$$
$$+ \frac{1}{x} \sum_{z=x+1}^{N} \sum_{\pi_{x^c} : \pi^{-1}_{>x}(z) = x} \left| [x] \cap \pi^{-1}_{>x}(R_x) \right| \left\| \langle \pi_{x^c}|_{D_{x^c}} \, |\phi_z\rangle \right\|^2$$

Proof. By Lemma 6.2 (ii), we have

$$\left\| E_D^{R,x} O^{\mathsf{SPO},\mathrm{inv}}_{XYD} \, |\phi\rangle \right\| - \left\| E_D^{R,x} \, |\phi\rangle \right\| \leq \sqrt{\frac{|R_x|}{x}} \left\| \left(I - |{+_x}\rangle\langle{+_x}|_{D_x} \right) |\phi\rangle \right\| + \left\| E_D^{R,x} O^{\mathsf{SPO},\mathrm{inv}}_{XYD} |{+_x}\rangle\langle{+_x}|_{D_x} \, |\phi\rangle \right\|, \quad (6.3)$$

and we need to bound the right-hand side term, which we can rewrite as

$$\left\| E_D^{R,x} O^{\mathsf{SPO},\mathrm{inv}}_{XYD} |{+_x}\rangle\langle{+_x}|_{D_x} \, |\phi\rangle \right\|^2 = \sum_{z \in [N]} \left\| E_D^{R,x} O^{\mathsf{SPO},\mathrm{inv},z}_{YD} |{+_x}\rangle\langle{+_x}|_{D_x} \, |\phi_z\rangle \right\|^2,$$

where $O^{\mathsf{SPO},\mathrm{inv},z}_{YD} := \langle z|_X \, O^{\mathsf{SPO},\mathrm{inv}}_{XYD} \, |z\rangle_X$. We analyze the the right-hand side summands and distinguish three cases:

(i) For $z = x$, we have the following bound from Lemma 6.2 (iii):

$$\left\| E_D^{R,x} O^{\mathsf{SPO},\mathrm{inv},z}_{YD} |{+_x}\rangle\langle{+_x}|_{D_x} \, |\phi_x\rangle \right\|^2 \leq 4 \frac{|R_x|}{x} \, \| |\phi_x\rangle \|^2 .$$

(ii) For $z < x$, we proceed similarly as in the corresponding case of Lemma 6.3. For $\pi \in S_N$, let us again write $\pi = \pi_{>x} \, (x \; t) \, \pi_{<x}$ as in Section 3.2, with $t \in [x]$, define $\pi_{x^c} := \pi_{>x} \pi_{<x}$, identify π_{x^c} with the indices t_k for $k \neq x$, and write $|\pi\rangle_D = |t\rangle_{D_x} \, |\pi_{x^c}\rangle_{D_{x^c}}$. Let

$$S_{x,z} := \left\{ \pi_{x^c} \; : \; \pi^{-1}_{>x}(z) \leq x \right\} = \left\{ \pi_{x^c} \; : \; \pi^{-1}_{>x}(z) < x \right\} = \left\{ \pi_{x^c} \; : \; \pi^{-1}_{>x}(z) = z \right\} = \left\{ \pi_{x^c} \; : \; t_w \neq z \; \forall w > x \right\}.$$

If $\pi_{x^c} \notin S_{x,z}$, then $\pi^{-1}(z) = z$ for any choice of t. Hence $O^{\mathsf{SPO},\mathrm{inv},z}_{YD}$ acts trivially on D_x and we have

$$E_D^{R,x} O^{\mathsf{SPO},\mathrm{inv},z}_{YD} \, |{+_x}\rangle_{D_x} \, |\pi_{x^c}\rangle_{D_{x^c}} = E_D^{R,x} \, |{+_x}\rangle_{D_x} \, O^{\mathsf{SPO},\mathrm{inv},z}_{YD} \, |\pi_{x^c}\rangle_{D_{x^c}} = 0. \quad (6.4)$$

Now suppose that $\pi_{x^c} \in S_{x,z}$. Then,

$$O^{\mathsf{SPO},\mathrm{inv},z}_{YD} \, |y, t, \pi_{x^c}\rangle_{Y D_x D_{x^c}} = |y \oplus \pi^{-1}(z), t, \pi_{x^c}\rangle_{Y D_x D_{x^c}} = \begin{cases} |y \oplus x, t, \pi_{x^c}\rangle_{Y D_x D_{x^c}} & \text{if } t = z, \\ |y \oplus \pi^{-1}_{<x}(z), t, \pi_{x^c}\rangle_{Y D_x D_{x^c}} & \text{otherwise.} \end{cases}$$

It follows that

$$O^{\mathsf{SPO},\mathrm{inv},z}_{YD} \, |y\rangle_Y \, |{+_x}\rangle_{D_x} \, |\pi_{x^c}\rangle_{D_{x^c}} = |y \oplus \pi^{-1}_{<x}(z)\rangle_Y \, |{+_x}\rangle_{D_x} \, |\pi_{x^c}\rangle_{D_{x^c}}$$
$$+ \frac{1}{\sqrt{x}} \Big(|y \oplus x\rangle_Y - |y \oplus \pi^{-1}_{<x}(z)\rangle_Y \Big) \, |z\rangle_{D_x} \, |\pi_{x^c}\rangle_{D_{x^c}} ,$$

hence

$$\left(I - |{+_x}\rangle\langle{+_x}|_{D_x} \right) O^{\mathsf{SPO},\mathrm{inv},z}_{YD} \, |y\rangle_Y \, |{+_x}\rangle_{D_x} \, |\pi_{x^c}\rangle_{D_{x^c}} = \frac{1}{\sqrt{x}} \Big(|y \oplus x\rangle_Y - |y \oplus \pi^{-1}_{<x}(z)\rangle_Y \Big)$$
$$\otimes \left(|z\rangle_{D_x} - \frac{1}{\sqrt{x}} \, |{+_x}\rangle_{D_x} \right) \otimes |\pi_{x^c}\rangle_{D_{x^c}}$$

and finally

$$E_D^{R,x} O^{\mathsf{SPO},\mathrm{inv},z}_{YD} \, |y\rangle_Y \, |{+_x}\rangle_{D_x} \, |\pi_{x^c}\rangle_{D_{x^c}}$$

$$
\begin{aligned}
&= \frac{1}{\sqrt{x}}\Big(|y\oplus x\rangle_Y - |y\oplus \pi^{-1}_{<x}(z)\rangle_Y\Big)\otimes\Big(\mathbf{1}_{z\in R_x}|z\rangle_{D_x} - \frac{1}{x}\sum_{t\in[x]\cap\pi^{-1}_{>x}(R_x)}|t\rangle_{D_x}\Big)\otimes|\pi_{x^c}\rangle_{D_{x^c}}\\
&= M^{(z)}_{YD_{x^c}}\left(\frac{1}{\sqrt{x}}|y\rangle_Y\otimes\Big(\mathbf{1}_{z\in R_x}|z\rangle_{D_x} - \frac{1}{x}\sum_{t\in[x]\cap\pi^{-1}_{>x}(R_x)}|t\rangle_{D_x}\Big)\otimes|\pi_{x^c}\rangle_{D_{x^c}}\right),
\end{aligned}
$$

where the operator $M^{(z)}_{YD_{x^c}}$ is defined by $M^{(z)}_{YD_{x^c}}|y\rangle_Y|\pi_{x^c}\rangle_{YD_{x^c}} = \big(|y\oplus x\rangle_Y - |y\oplus\pi^{-1}_{<x}(z)\rangle_Y\big)|\pi_{x^c}\rangle_{D_{x^c}}$ and has operator norm $\leq\sqrt{2}$. Thus:

$$
E^{R,x}_D O^{\mathsf{SPO},\mathrm{inv},z}_{YD}|+_x\rangle_{D_x}|\pi_{x^c}\rangle_{D_{x^c}} = M^{(z)}_{YD_{x^c}}\frac{1}{\sqrt{x}}\Big(\mathbf{1}_{z\in R_x}|z\rangle_{D_x} - \frac{1}{x}\sum_{t\in[x]\cap\pi^{-1}_{>x}(R_x)}|t\rangle_{D_x}\Big)\otimes|\pi_{x^c}\rangle_{D_{x^c}}
$$

Using the above and Eq. (6.4), we can now bound the desired norm:

$$
\begin{aligned}
&\Big\|E^{R,x}_D O^{\mathsf{SPO},\mathrm{inv},z}_{YD}|+_x\rangle\langle+_x|_{D_x}|\phi_z\rangle\Big\|^2\\
&= \Big\|E^{R,x}_D O^{\mathsf{SPO},\mathrm{inv},z}_{YD}|+_x\rangle\langle+_x|_{D_x}\sum_{\pi_{x^c}}|\pi_{x^c}\rangle\langle\pi_{x^c}|_{D_{x^c}}|\phi_z\rangle\Big\|^2\\
&= \Big\|\sum_{\pi_{x^c}\in S_{x,z}} E^{R,x}_D O^{\mathsf{SPO},\mathrm{inv},z}_{YD}|+_x\rangle_{D_x}|\pi_{x^c}\rangle_{D_{x^c}}\langle+_x|_{D_x}\langle\pi_{x^c}|_{D_{x^c}}|\phi_z\rangle\Big\|^2\\
&= \Big\|M^{(z)}_{YD_{x^c}}\sum_{\pi_{x^c}\in S_{x,z}}\frac{1}{\sqrt{x}}\Big(\mathbf{1}_{z\in R_x}|z\rangle_{D_x} - \frac{1}{x}\sum_{t\in[x]\cap\pi^{-1}_{>x}(R_x)}|t\rangle_{D_x}\Big)|\pi_{x^c}\rangle_{D_{x^c}}\langle+_x|_{D_x}\langle\pi_{x^c}|_{D_{x^c}}|\phi_z\rangle\Big\|^2\\
&\leq \frac{2}{x}\sum_{\pi_{x^c}\in S_{x,z}}\Big\|\Big(\mathbf{1}_{z\in R_x}|z\rangle_{D_x} - \frac{1}{x}\sum_{t\in[x]\cap\pi^{-1}_{>x}(R_x)}|t\rangle_{D_x}\Big)\langle+_x|_{D_x}\langle\pi_{x^c}|_{D_{x^c}}|\phi_z\rangle\Big\|^2\\
&\leq \frac{4}{x}\sum_{\pi_{x^c}\in S_{x,z}}\Big(\mathbf{1}_{z\in R_x}\Big\|\langle+_x|_{D_x}\langle\pi_{x^c}|_{D_{x^c}}|\phi_z\rangle\Big\|^2 + \frac{1}{x^2}\Big\|\sum_{t\in[x]\cap\pi^{-1}_{>x}(R_x)}|t\rangle_{D_x}\langle+_x|_{D_x}\langle\pi_{x^c}|_{D_{x^c}}|\phi_z\rangle\Big\|^2\Big)\\
&= \frac{4}{x}\sum_{\pi_{x^c}\in S_{x,z}}\mathbf{1}_{z\in R_x}\Big\|\langle+_x|_{D_x}\langle\pi_{x^c}|_{D_{x^c}}|\phi_z\rangle\Big\|^2 + \frac{4}{x^3}\sum_{\pi_{x^c}\in S_{x,z}}\sum_{t\in[x]\cap\pi^{-1}_{>x}(R_x)}\Big\|\langle+_x|_{D_x}\langle\pi_{x^c}|_{D_{x^c}}|\phi_z\rangle\Big\|^2\\
&\leq \frac{4}{x}\sum_{\pi_{x^c}\in S_{x,z}}\mathbf{1}_{z\in R_x}\Big\|\langle\pi_{x^c}|_{D_{x^c}}|\phi_z\rangle\Big\|^2 + \frac{4}{x^3}\sum_{\pi_{x^c}\in S_{x,z}}\sum_{t\in[x]\cap\pi^{-1}_{>x}(R_x)}\Big\|\langle\pi_{x^c}|_{D_{x^c}}|\phi_z\rangle\Big\|^2
\end{aligned}
$$

following the same reasoning as the proof of Lemma 6.3. It follows that

$$
\begin{aligned}
&\sum_{z=1}^{x-1}\Big\|E^{R,x}_D O^{\mathsf{SPO},\mathrm{inv},z}_{YD}|+_x\rangle\langle+_x|_{D_x}|\phi_z\rangle\Big\|^2\\
&\leq \frac{4}{x}\sum_{z=1}^{x-1}\sum_{\pi_{x^c}\in S_{x,z}}\mathbf{1}_{z\in R_x}\Big\|\langle\pi_{x^c}|_{D_{x^c}}|\phi_z\rangle\Big\|^2 + \frac{4}{x^3}\sum_{z=1}^{x-1}\sum_{\pi_{x^c}\in S_{x,z}}\sum_{t\in[x]\cap\pi^{-1}_{>x}(R_x)}\Big\|\langle\pi_{x^c}|_{D_{x^c}}|\phi_z\rangle\Big\|^2\\
&\leq \frac{4}{x}\sum_{z=1}^{x-1}\sum_{\pi_{x^c}\in S_{x,z}}\mathbf{1}_{z\in R_x}\Big\|\langle\pi_{x^c}|_{D_{x^c}}|\phi_z\rangle\Big\|^2 + \frac{4}{x^3}\sum_{\pi_{x^c}}\sum_{t\in[x]\cap\pi^{-1}_{>x}(R_x)}\Big\|\langle\pi_{x^c}|_{D_{x^c}}|\phi\rangle\Big\|^2\\
&\leq \frac{4}{x}\sum_{z=1}^{x-1}\sum_{\pi_{x^c}\in S_{x,z}}\mathbf{1}_{z\in R_x}\Big\|\langle\pi_{x^c}|_{D_{x^c}}|\phi_z\rangle\Big\|^2 + \frac{4}{x^2}\frac{|R_x|}{N}\\
&= \frac{4}{x}\sum_{z=1}^{x-1}\sum_{\pi_{x^c}:\pi^{-1}_{>x}(z)<x}\mathbf{1}_{z\in R_x}\Big\|\langle\pi_{x^c}|_{D_{x^c}}|\phi_z\rangle\Big\|^2 + \frac{4}{x^2}\frac{|R_x|}{N},
\end{aligned}
$$

where the last inequality follows from the same argument as in the proof of Lemma 6.3 and is using the assumption that $\|\langle\pi_{x^c}|_{D_{x^c}}|\phi\rangle\|^2 = \frac{x}{N!}$ for all $\pi\in S_N$.

(iii) For $z > x$, we use the same notation as above, but instead of $S_{x,z}$ we consider

$$S'_{x,z} := \big\{\pi_{x^c} \ : \ \pi^{-1}_{>x}(z) < x\big\},$$
$$S''_{x,z} := \big\{\pi_{x^c} \ : \ \pi^{-1}_{>x}(z) = x\big\}.$$

If $\pi_{x^c} \notin S'_{x,z} \cup S''_{x,z}$, then we see as above that $\pi^{-1}(z)$ does not depend on the choice of t and hence

$$E^{R,x}_D O^{\mathsf{SPO},\mathrm{inv},z}_{YD} \left|+_x\right\rangle_{D_x} \left|\pi_{x^c}\right\rangle_{D_{x^c}} = E^{R,x}_D \left|+_x\right\rangle_{D_x} O^{\mathsf{SPO},\mathrm{inv},z}_{YD} \left|\pi_{x^c}\right\rangle_{D_{x^c}} = 0. \tag{6.5}$$

Next, suppose that $\pi_{x^c} \in S'_{x,z}$. Then,

$$O^{\mathsf{SPO},\mathrm{inv},z}_{YD} \left|y, t, \pi_{x^c}\right\rangle_{YD_xD_{x^c}} = \begin{cases} \left|y \oplus x, t, \pi_{x^c}\right\rangle_{YD_xD_{x^c}} & \text{if } t = \pi^{-1}_{>x}(z), \\ \left|y \oplus \pi^{-1}_{x^c}(z), t, \pi_{x^c}\right\rangle_{YD_xD_{x^c}} & \text{otherwise.} \end{cases}$$

It follows that

$$\begin{aligned} O^{\mathsf{SPO},\mathrm{inv},z}_{YD} \left|y\right\rangle_Y \left|+_x\right\rangle_{D_x} \left|\pi_{x^c}\right\rangle_{D_{x^c}} &= \left|y \oplus \pi^{-1}_{x^c}(z)\right\rangle_Y \left|+_x\right\rangle_{D_x} \left|\pi_{x^c}\right\rangle_{D_{x^c}} \\ &\quad + \frac{1}{\sqrt{x}}\Big(\left|y \oplus x\right\rangle_Y - \left|y \oplus \pi^{-1}_{x^c}(z)\right\rangle_Y\Big) \left|\pi^{-1}_{>x}(z)\right\rangle_{D_x} \left|\pi_{x^c}\right\rangle_{D_{x^c}}, \end{aligned}$$

hence

$$\begin{aligned} \big(I - \left|+_x\right\rangle\!\left\langle+_x\right|_{D_x}\big) O^{\mathsf{SPO},\mathrm{inv},z}_{YD} \left|y\right\rangle_Y \left|+_x\right\rangle_{D_x} \left|\pi_{x^c}\right\rangle_{D_{x^c}} &= \frac{1}{\sqrt{x}}\Big(\left|y \oplus x\right\rangle_Y - \left|y \oplus \pi^{-1}_{x^c}(z)\right\rangle_Y\Big) \\ &\quad \otimes \left(\left|\pi^{-1}_{>x}(z)\right\rangle_{D_x} - \frac{1}{\sqrt{x}}\left|+_x\right\rangle_{D_x}\right) \otimes \left|\pi_{x^c}\right\rangle_{D_{x^c}} \end{aligned}$$

and finally

$$\begin{aligned} &E^{R,x}_D O^{\mathsf{SPO},\mathrm{inv},z}_{YD} \left|y\right\rangle_Y \left|+_x\right\rangle_{D_x} \left|\pi_{x^c}\right\rangle_{D_{x^c}} \\ &= \frac{1}{\sqrt{x}}\Big(\left|y \oplus x\right\rangle_Y - \left|y \oplus \pi^{-1}_{x^c}(z)\right\rangle_Y\Big) \otimes \Bigg(\mathbf{1}_{z \in R_x} \left|\pi^{-1}_{>x}(z)\right\rangle_{D_x} - \frac{1}{x} \sum_{t \in [x] \cap \pi^{-1}_{>x}(R_x)} \left|t\right\rangle_{D_x}\Bigg) \otimes \left|\pi_{x^c}\right\rangle_{D_{x^c}} \\ &= M^{(z)}_{YD_{x^c}} \left(\frac{1}{\sqrt{x}} \left|y\right\rangle_Y \otimes \Bigg(\mathbf{1}_{z \in R_x} \left|\pi^{-1}_{>x}(z)\right\rangle_{D_x} - \frac{1}{x} \sum_{t \in [x] \cap \pi^{-1}_{>x}(R_x)} \left|t\right\rangle_{D_x}\Bigg) \otimes \left|\pi_{x^c}\right\rangle_{D_{x^c}}\right), \end{aligned}$$

where the operator $M'_{YD_{x^c},z}$ is defined by $M'_{YD_{x^c},z} \left|y\right\rangle_Y \left|\pi_{x^c}\right\rangle_{YD_{x^c}} = \big(\left|y \oplus x\right\rangle_Y - \left|y \oplus \pi^{-1}_{x^c}(z)\right\rangle_Y\big) \left|\pi_{x^c}\right\rangle_{D_{x^c}}$ and has operator norm $\leq \sqrt{2}$. Thus:

$$E^{R,x}_D O^{\mathsf{SPO},\mathrm{inv},z}_{YD} \left|+_x\right\rangle_{D_x} \left|\pi_{x^c}\right\rangle_{D_{x^c}} = M'_{YD_{x^c},z} \frac{1}{\sqrt{x}}\Bigg(\mathbf{1}_{z \in R_x} \left|\pi^{-1}_{>x}(z)\right\rangle_{D_x} - \frac{1}{x} \sum_{t \in [x] \cap \pi^{-1}_{>x}(R_x)} \left|t\right\rangle_{D_x}\Bigg) \otimes \left|\pi_{x^c}\right\rangle_{D_{x^c}}. \tag{6.6}$$

Finally, suppose that $\pi_{x^c} \in S''_{x,z}$. Then,

$$O^{\mathsf{SPO},\mathrm{inv},z}_{YD} \left|y, t, \pi_{x^c}\right\rangle_{YD_xD_{x^c}} = \left|y \oplus \pi^{-1}_{<x}(t), t, \pi_{x^c}\right\rangle_{YD_xD_{x^c}},$$

hence

$$\begin{aligned} \big(I - \left|+_x\right\rangle\!\left\langle+_x\right|_{D_x}\big) O^{\mathsf{SPO},\mathrm{inv},z}_{YD} \left|y\right\rangle_Y \left|+_x\right\rangle_{D_x} \left|\pi_{x^c}\right\rangle_{D_{x^c}} &= \frac{1}{\sqrt{x}} \sum_{t=1}^{x} \left|y \oplus \pi^{-1}_{<x}(t)\right\rangle_Y \\ &\quad \otimes \left(\left|t\right\rangle_{D_x} - \frac{1}{\sqrt{x}}\left|+_x\right\rangle_{D_x}\right) \otimes \left|\pi_{x^c}\right\rangle_{D_{x^c}}, \end{aligned}$$

and finally

$$E^{R,x}_D O^{\mathsf{SPO},\mathrm{inv},z}_{YD} \left|y\right\rangle_Y \left|+_x\right\rangle_{D_x} \left|\pi_{x^c}\right\rangle_{D_{x^c}}$$

$$
\begin{aligned}
&= \frac{1}{\sqrt{x}} \sum_{t=1}^{x} |y \oplus \pi^{-1}_{<x}(t)\rangle_Y \otimes \left(\mathbf{1}_{\pi_{>x}(t)\in R_x} |t\rangle_{D_x} - \frac{1}{x} \sum_{t'\in[x]\cap\pi^{-1}_{>x}(R_x)} |t'\rangle_{D_x} \right) \otimes |\pi_{x^c}\rangle_{D_{x^c}} \\
&= M''_{YD} \left(\frac{1}{\sqrt{x}} |y\rangle_Y \otimes \sum_{t\in[x]\cap\pi^{-1}_{>x}(R_x)} |t\rangle_{D_x} \otimes |\pi_{x^c}\rangle_{D_{x^c}} \right),
\end{aligned}
$$

with $M''_{YD}\,|y,t,\pi_{x^c}\rangle := (|y\oplus\pi^{-1}_{<x}(t)\rangle_Y - \frac{1}{x}\sum_{t'=1}^{x}|y\oplus t'\rangle_Y)\,|t\rangle_{D_x}\,|\pi_{x^c}\rangle_{D_{x^c}}$, an operator of norm ≤ 2, hence

$$
E^{R,x}_{D} O^{\mathsf{SPO},\mathrm{inv},z}_{YD} \, |{+_x}\rangle_{D_x} |\pi_{x^c}\rangle_{D_{x^c}} = M''_{YD} \frac{1}{\sqrt{x}} \sum_{t\in[x]\cap\pi^{-1}_{>x}(R_x)} |t\rangle_{D_x} \otimes |\pi_{x^c}\rangle_{D_{x^c}} \,.
$$

Together with Eqs. (6.5) and (6.6), we obtain

$$
\begin{aligned}
&\left\| E^{R,x}_{D} O^{\mathsf{SPO},\mathrm{inv},z}_{YD} |{+_x}\rangle\langle{+_x}|_{D_x} |\phi_z\rangle \right\|^2 \\
&= \left\| E^{R,x}_{D} O^{\mathsf{SPO},\mathrm{inv},z}_{YD} |{+_x}\rangle\langle{+_x}|_{D_x} \sum_{\pi_{x^c}} |\pi_{x^c}\rangle\langle\pi_{x^c}|_{D_{x^c}} |\phi_z\rangle \right\|^2 \\
&= \left\| \sum_{\pi_{x^c}\in S'_{x,z}\cup S''_{x,z}} E^{R,x}_{D} O^{\mathsf{SPO},\mathrm{inv},z}_{YD} \, |{+_x}\rangle_{D_x} |\pi_{x^c}\rangle_{D_{x^c}} \langle{+_x}|_{D_x} \langle\pi_{x^c}|_{D_{x^c}} |\phi_z\rangle \right\|^2 \\
&\leq 2 \left\| \sum_{\pi_{x^c}\in S'_{x,z}} E^{R,x}_{D} O^{\mathsf{SPO},\mathrm{inv},z}_{YD} \, |{+_x}\rangle_{D_x} |\pi_{x^c}\rangle_{D_{x^c}} \langle{+_x}|_{D_x} \langle\pi_{x^c}|_{D_{x^c}} |\phi_z\rangle \right\|^2 \\
&+ 2 \left\| \sum_{\pi_{x^c}\in S''_{x,z}} E^{R,x}_{D} O^{\mathsf{SPO},\mathrm{inv},z}_{YD} \, |{+_x}\rangle_{D_x} |\pi_{x^c}\rangle_{D_{x^c}} \langle{+_x}|_{D_x} \langle\pi_{x^c}|_{D_{x^c}} |\phi_z\rangle \right\|^2 \\
&= 2 \left\| M'_{YD_{x^c},z} \sum_{\pi_{x^c}\in S'_{x,z}} \frac{1}{\sqrt{x}} \Big(\mathbf{1}_{z\in R_x} |\pi^{-1}_{>x}(z)\rangle_{D_x} - \frac{1}{x} \sum_{t\in[x]\cap\pi^{-1}_{>x}(R_x)} |t\rangle_{D_x} \Big) |\pi_{x^c}\rangle_{D_{x^c}} \langle{+_x}|_{D_x} \langle\pi_{x^c}|_{D_{x^c}} |\phi_z\rangle \right\|^2 \\
&+ 2 \left\| M''_{YD} \sum_{\pi_{x^c}\in S''_{x,z}} \frac{1}{\sqrt{x}} \sum_{t\in[x]\cap\pi^{-1}_{>x}(R_x)} |t\rangle_{D_x} |\pi_{x^c}\rangle_{D_{x^c}} \langle{+_x}|_{D_x} \langle\pi_{x^c}|_{D_{x^c}} |\phi_z\rangle \right\|^2 \\
&\leq \frac{4}{x} \sum_{\pi_{x^c}\in S'_{x,z}} \left\| \Big(\mathbf{1}_{z\in R_x} |\pi^{-1}_{>x}(z)\rangle_{D_x} - \frac{1}{x} \sum_{t\in[x]\cap\pi^{-1}_{>x}(R_x)} |t\rangle_{D_x} \Big) \langle{+_x}|_{D_x} \langle\pi_{x^c}|_{D_{x^c}} |\phi_z\rangle \right\|^2 \\
&+ \frac{8}{x} \sum_{\pi_{x^c}\in S''_{x,z}} \left\| \sum_{t\in[x]\cap\pi^{-1}_{>x}(R_x)} |t\rangle_{D_x} \langle{+_x}|_{D_x} \langle\pi_{x^c}|_{D_{x^c}} |\phi_z\rangle \right\|^2 \\
&\leq \frac{8}{x} \sum_{\pi_{x^c}\in S'_{x,z}} \mathbf{1}_{z\in R_x} \left\| \langle{+_x}|_{D_x} \langle\pi_{x^c}|_{D_{x^c}} |\phi_z\rangle \right\|^2 + \frac{8}{x^3} \sum_{\pi_{x^c}\in S'_{x,z}} \sum_{t\in[x]\cap\pi^{-1}_{>x}(R_x)} \left\| \langle{+_x}|_{D_x} \langle\pi_{x^c}|_{D_{x^c}} |\phi_z\rangle \right\|^2 \\
&+ \frac{8}{x} \sum_{\pi_{x^c}\in S''_{x,z}} \sum_{t\in[x]\cap\pi^{-1}_{>x}(R_x)} \left\| \langle{+_x}|_{D_x} \langle\pi_{x^c}|_{D_{x^c}} |\phi_z\rangle \right\|^2 \\
&\leq \frac{8}{x} \sum_{\pi_{x^c}\in S'_{x,z}} \mathbf{1}_{z\in R_x} \left\| \langle\pi_{x^c}|_{D_{x^c}} |\phi_z\rangle \right\|^2 + \frac{8}{x^3} \sum_{\pi_{x^c}\in S'_{x,z}} \sum_{t\in[x]\cap\pi^{-1}_{>x}(R_x)} \left\| \langle\pi_{x^c}|_{D_{x^c}} |\phi_z\rangle \right\|^2 \\
&+ \frac{8}{x} \sum_{\pi_{x^c}\in S''_{x,z}} \sum_{t\in[x]\cap\pi^{-1}_{>x}(R_x)} \left\| \langle\pi_{x^c}|_{D_{x^c}} |\phi_z\rangle \right\|^2.
\end{aligned}
$$

By summing the above estimate over all $z > x$, we obtain

$$
\sum_{z=x+1}^{N} \left\| E^{R,x}_{D} O^{\mathsf{SPO},\mathrm{inv},z}_{YD} |{+_x}\rangle\langle{+_x}|_{D_x} |\phi_z\rangle \right\|^2
$$

$$
\begin{aligned}
&\leq \frac{8}{x} \sum_{z=x+1}^{N} \sum_{\pi_{x^c} \in S'_{x,z}} \mathbf{1}_{z\in R_x} \left\| \langle \pi_{x^c}|_{D_{x^c}} |\phi_z\rangle \right\|^2 + \frac{8}{x^3} \sum_{z=x+1}^{N} \sum_{\pi_{x^c} \in S'_{x,z}} \sum_{t \in [x] \cap \pi^{-1}_{>x}(R_x)} \left\| \langle \pi_{x^c}|_{D_{x^c}} |\phi_z\rangle \right\|^2 \\
&+ \frac{8}{x} \sum_{z=x+1}^{N} \sum_{\pi_{x^c} \in S''_{x,z}} \sum_{t \in [x] \cap \pi^{-1}_{>x}(R_x)} \left\| \langle \pi_{x^c}|_{D_{x^c}} |\phi_z\rangle \right\|^2 \\
&\leq \frac{8}{x} \sum_{z=x+1}^{N} \sum_{\pi_{x^c} \in S'_{x,z}} \mathbf{1}_{z\in R_x} \left\| \langle \pi_{x^c}|_{D_{x^c}} |\phi_z\rangle \right\|^2 + \frac{8}{x^2}\frac{|R_x|}{N} + \frac{8}{x} \sum_{z=x+1}^{N} \sum_{\pi_{x^c} \in S''_{x,z}} \sum_{t \in [x] \cap \pi^{-1}_{>x}(R_x)} \left\| \langle \pi_{x^c}|_{D_{x^c}} |\phi_z\rangle \right\|^2 \\
&= \frac{8}{x} \sum_{z=x+1}^{N} \sum_{\pi_{x^c} : \pi^{-1}_{>x}(z) < x} \mathbf{1}_{z\in R_x} \left\| \langle \pi_{x^c}|_{D_{x^c}} |\phi_z\rangle \right\|^2 + \frac{8}{x^2}\frac{|R_x|}{N} \\
&+ \frac{8}{x} \sum_{z=x+1}^{N} \sum_{\pi_{x^c} : \pi^{-1}_{>x}(z) = x} \left|[x] \cap \pi^{-1}_{>x}(R_x)\right| \left\| \langle \pi_{x^c}|_{D_{x^c}} |\phi_z\rangle \right\|^2 .
\end{aligned}
$$

The last estimate follows as in part (ii).

By combining the results of (i), (ii), and (iii), we obtain the following bound on the right-hand side Eq. (6.3):

$$
\begin{aligned}
\left\| E_D^{R,x} O_{XYD}^{\mathsf{SPO},\mathrm{inv}} |+_x\rangle\langle +_x|_{D_x} |\phi\rangle \right\| &\leq \sqrt{\begin{aligned} &4\frac{|R_x|}{x} \left\| |\phi_x\rangle \right\|^2 + \frac{4}{x} \sum_{z=1}^{x-1} \sum_{\pi_{x^c} : \pi^{-1}_{>x}(z) < x} \mathbf{1}_{z\in R_x} \left\| \langle \pi_{x^c}|_{D_{x^c}} |\phi_z\rangle \right\|^2 + \frac{4}{x^2}\frac{|R_x|}{N} \\ &+ \frac{8}{x} \sum_{z=x+1}^{N} \sum_{\pi_{x^c} : \pi^{-1}_{>x}(z) < x} \mathbf{1}_{z\in R_x} \left\| \langle \pi_{x^c}|_{D_{x^c}} |\phi_z\rangle \right\|^2 + \frac{8}{x^2}\frac{|R_x|}{N} \\ &+ \frac{8}{x} \sum_{z=x+1}^{N} \sum_{\pi_{x^c} : \pi^{-1}_{>x}(z) = x} \left|[x] \cap \pi^{-1}_{>x}(R_x)\right| \left\| \langle \pi_{x^c}|_{D_{x^c}} |\phi_z\rangle \right\|^2 \end{aligned}} \\
&\leq 4 \sqrt{\begin{aligned} &\frac{|R_x|}{x} \left\| |\phi_x\rangle \right\|^2 + \frac{1}{x^2}\frac{|R_x|}{N} + \frac{1}{x} \sum_{z=1}^{N} \sum_{\pi_{x^c} : \pi^{-1}_{>x}(z) < x} \mathbf{1}_{z\in R_x} \left\| \langle \pi_{x^c}|_{D_{x^c}} |\phi_z\rangle \right\|^2 \\ &+ \frac{1}{x} \sum_{z=x+1}^{N} \sum_{\pi_{x^c} : \pi^{-1}_{>x}(z) = x} \left|[x] \cap \pi^{-1}_{>x}(R_x)\right| \left\| \langle \pi_{x^c}|_{D_{x^c}} |\phi_z\rangle \right\|^2 \end{aligned}} \\
&\leq 4 \sqrt{\begin{aligned} &\frac{|R_x|}{x} \left\| |\phi_x\rangle \right\|^2 + \frac{1}{x^2}\frac{|R_x|}{N} + \frac{1}{x} \sum_{z\in R_x} \sum_{\pi_{x^c} : \pi^{-1}_{>x}(z) < x} \left\| \langle \pi_{x^c}|_{D_{x^c}} |\phi_z\rangle \right\|^2 \\ &+ \frac{1}{x} \sum_{z=x+1}^{N} \sum_{\pi_{x^c} : \pi^{-1}_{>x}(z) = x} \left|[x] \cap \pi^{-1}_{>x}(R_x)\right| \left\| \langle \pi_{x^c}|_{D_{x^c}} |\phi_z\rangle \right\|^2 \end{aligned}} .
\end{aligned}
$$

Now the lemma follows. □

As a consequence of the preceding lemmas, we obtain a bound that holds for any query algorithm that makes q queries.

Proposition 6.5. *Let $\mathcal{A}$ be a unitary query algorithm on registers AXY, where X and Y are N-dimensional registers, that gets query access to two oracles that each act on XY. Let $|\phi\rangle_{AXYD}$ be the joint state of algorithm and oracle defined by running $\mathsf{Init}_D^{\mathsf{SPO}}$ and then $\mathcal{A}^{\mathsf{SPO}_D}$. Suppose that $\mathcal{A}$ makes in total q queries to its oracles. Then for any $x \in [N]$,*

$$
\left\| E_D^{R,x} |\phi\rangle \right\| \leq \sum_{j=1}^{q} \left(\sqrt{\frac{|R_x|}{x}} \left\| \left(I - |+_x\rangle\langle +_x|_{D_x}\right) |\phi^{(j)}\rangle \right\| + 4\sqrt{\zeta_x^{(j)}} \right)
$$

$$\leq \sqrt{2q \sum_{j=1}^{q} \left(\frac{|R_x|}{x} \left\| \left(I - |+_x\rangle\langle +_x|_{D_x}\right) |\phi^{(j)}\rangle \right\|^2 + 16\zeta_x^{(j)} \right)},$$

where $|\phi^{(j)}\rangle_{AXYD}$ denotes the state right before the j-th query of $\mathcal{A}^{\mathsf{SPO}_D}$ and where we defined $\zeta_x^{(j)} := \zeta_{|\phi^{(j)}\rangle,x}$ as in Lemma 6.3 if the j-th query is a forward query and otherwise $\zeta_x^{(j)} := \zeta^{\mathrm{inv}}_{|\phi^{(j)}\rangle,x}$, as in Lemma 6.4.

Proof. We will prove the first inequality, since the second follows directly using the Cauchy-Schwartz inequality. To this end, we first observe that $\| \langle \pi|_D |\phi^{(j)}\rangle_{AXYD} \|^2 = \frac{1}{N!}$ for every $j \in [q]$ and $\pi \in S_N$. This is because the database is initialized in a uniform superposition of the basis states $|\pi\rangle_D$, the query unitaries O^{SPO}_{XYD} and $O^{\mathsf{SPO},\mathrm{inv}}_{XYD}$ are controlled on D in this basis and hence commute with a basis measurement, and all other unitaries applied by $\mathcal{A}$ only act on registers AXY. Thus the states $|\phi^{(j)}\rangle_{AXYD}$ satisfy the requirements of Lemmas 6.3 and 6.4 and hence we see that, for $Q^{\mathsf{SPO}} \in \{O^{\mathsf{SPO}}, O^{\mathsf{SPO},\mathrm{inv}}\}$ and any $x \in [N]$,

$$\left\| E_D^{R,x} Q^{\mathsf{SPO}}_{XYD} |\phi^{(j)}\rangle \right\| - \left\| E_D^{R,x} |\phi^{(j)}\rangle \right\| \leq \sqrt{\frac{|R_x|}{x}} \left\| \left(I - |+_x\rangle\langle +_x|_{D_x}\right) |\phi^{(j)}\rangle \right\| + 4\sqrt{\zeta_x^{(j)}} \tag{6.7}$$

We will now show the following inequality for every $k \in \{0, 1, \ldots, q\}$, with $|\phi^{(q+1)}\rangle := |\phi\rangle$:

$$\left\| E_D^{R,x} |\phi^{(k+1)}\rangle \right\| \leq \sum_{j=1}^{k} \left(\sqrt{\frac{|R_x|}{x}} \left\| \left(I - |+_x\rangle\langle +_x|_{D_x}\right) |\phi^{(j)}\rangle \right\| + 4\sqrt{\zeta_x^{(j)}} \right), \tag{6.8}$$

This will conclude the proof, since for $k = q$ it is the desired inequality. We use induction over k. For $k = 0$, the inequality holds trivially, since all database registers are initialized in a uniform superposition and hence $(I - |+_x\rangle\langle +_x|_{D_x}) |\phi\rangle = 0$ and $E_D^{R,x} |\phi\rangle = 0$. For the induction step, note that for any $k > 0$ we can write

$$E_D^{R,x} |\phi^{(k+1)}\rangle_{AXYD} = E_D^{R,x} U_{AXY} Q^{\mathsf{SPO}}_{XYD} |\phi^{(k)}\rangle_{AXYD} = U_{AXY} E_D^{R,x} Q^{\mathsf{SPO}}_{XYD} |\phi^{(k)}\rangle_{AXYD},$$

for some unitary U_{AXY}, where $Q^{\mathsf{SPO}} \in \{O^{\mathsf{SPO}}, O^{\mathsf{SPO},\mathrm{inv}}\}$ depending on $\mathcal{A}$'s choice for the k-th query. Here we have used that the operators U_{AXY} and $E_D^{R,x}$ act on disjoint registers. Using first the unitary invariance of the norm, then Eq. (6.7), and finally the induction hypothesis, we get

$$\begin{aligned} \left\| E_D^{R,x} |\phi^{(k+1)}\rangle \right\| &= \left\| E_D^{R,x} Q^{\mathsf{SPO}}_{XYD} |\phi^{(k)}\rangle_{AXYD} \right\| \\ &\leq \left\| E_D^{R,x} |\phi^{(k)}\rangle \right\| + \sqrt{\frac{|R_x|}{x}} \left\| \left(I - |+_x\rangle\langle +_x|_{D_x}\right) |\phi^{(k)}\rangle \right\| + 4\sqrt{\zeta^{(k)}_{|\phi^{(k)}\rangle,x}} \\ &\leq \sum_{j=1}^{k} \left(\sqrt{\frac{|R_x|}{x}} \left\| \left(I - |+_x\rangle\langle +_x|_{D_x}\right) |\phi^{(j)}\rangle \right\| + 4\sqrt{\zeta_x^{(j)}} \right). \end{aligned}$$

This concludes the proof of Eq. (6.8) and hence proof of the corollary. □

6.2 Bounding the Success Probability in Expectation

As discussed at the beginning of the section we will now lift the worst-case bounds established in the previous section to the average case by running the query algorithm with the *twirled* permutation oracle for uniformly random $\sigma, \tau \in S_N$ and also averaging over the choice of $x \in [N]$, corresponding to the progress measure

$$\mathop{\mathbf{E}}_{\substack{x \leftarrow [N], \\ \sigma,\tau \leftarrow S_N}} \left\| E_D^{R^{\sigma,\tau},x} |\phi^{\sigma,\tau}\rangle \right\|^2,$$

where $R^{\sigma,\tau}$ is the twirled relation defined in (6.1). Note that $r_{\max} = \max\{\max_{x\in[N]} |R_x^{\sigma,\tau}|, \max_{y\in[N]} |(R^{\sigma,\tau})_y^{\mathrm{inv}}|\}$.

The main result of this section is the following. It shows that the randomized test for the twirled relation rarely succeeds for algorithms making not too many queries, up to an error term that captures the "sparsity" of the database and that will be bounded in Section 6.3.

Proposition 6.6. *Let $\mathcal{A}$ be a unitary query algorithm on registers AXY, where X and Y are N-dimensional registers, that gets query access to two oracles that each act on XY. Suppose that $\mathcal{A}$ makes in total q queries to its oracles. For every $\sigma, \tau \in S_N$, let $|\phi^{\sigma,\tau}\rangle_{AXYD}$ be the joint state of algorithm and oracle defined by running $\mathsf{Init}_D^{\mathsf{SPO}}$ and then $\mathcal{A}^{\mathsf{TSPO}_D^{\sigma,\tau}}$. Let $\mathcal{B}$ be the query algorithm on registers BXY as in Lemma 4.9, with $B = AZ$ and Z another N-dimensional register. Then,*

$$\mathop{\mathbf{E}}_{\substack{x \leftarrow [N],\\ \sigma,\tau \leftarrow S_N}} \left\| E_D^{R^{\sigma,\tau},x} |\phi^{\sigma,\tau}\rangle \right\|^2 \leq 384 \frac{q^2 r_{\max}\big(\ln(N)+2\big)}{N^2} + 4q r_{\max} \sum_{j=1}^{2q} \mathop{\mathbf{E}}_{\substack{x \leftarrow [N],\\ \sigma,\tau \leftarrow S_N}} \frac{\left\| \big(I - |+_x\rangle\langle +_x|_{D_x}\big) |\phi^{\sigma,\tau,(j)}\rangle \right\|^2}{x}$$

where $|\phi^{\sigma,\tau,(j)}\rangle_{BXYD}$ is the state given by running $\mathsf{Init}_D^{\mathsf{SPO}}$ and then $\mathcal{B}^{\mathsf{TSPO}_D^{\sigma,\tau}}$ until right before the j-th query.

Proof. By Lemma 4.9 for every $\sigma, \tau \in S_N$ it holds that

$$\mathcal{B}^{\mathsf{TSPO}_D^{\sigma,\tau}} = \mathcal{B}_{\sigma,\tau}^{\mathsf{SPO}_D}.$$

with $\mathcal{B}_{\sigma,\tau}$ defined in the statement of the lemma. Moreover, the joint state of the algorithm and oracle defined by running $\mathsf{Init}_D^{\mathsf{SPO}}$ and then either of these two algorithms is given by $|\chi^{\sigma,\tau}\rangle_{BXYD} = |\phi^{\sigma,\tau}\rangle_{AXYD} \otimes |0\rangle_Z$. Thus we can apply Proposition 6.5 with the relation $R^{\sigma,\tau}$ and the algorithm $\mathcal{B}_{\sigma,\tau}$, which makes $2q$ queries to the untwirled standard oracle, to obtain

$$\left\| E_D^{R^{\sigma,\tau},x} |\phi^{\sigma,\tau}\rangle \right\|^2 = \left\| E_D^{R^{\sigma,\tau},x} |\chi^{\sigma,\tau}\rangle \right\|^2 \leq 4q \sum_{j=1}^{2q} \left(\frac{|R_x^{\sigma,\tau}|}{x} \left\| \big(I - |+_x\rangle\langle +_x|_{D_x}\big) |\chi^{\sigma,\tau,(j)}\rangle \right\|^2 + 16 \zeta_x^{\sigma,\tau,(j)} \right), \tag{6.9}$$

where $|\chi^{\sigma,\tau,(j)}\rangle_{BXYD}$ denotes the state right before the j-th query of $\mathcal{B}_{\sigma,\tau}^{\mathsf{SPO}_D}$ and

$$\zeta_x^{\sigma,\tau,(j)} = \begin{cases} \zeta_{|\chi^{\sigma,\tau,(j)}\rangle,x} & \text{if the } j\text{-th query of } \mathcal{B}_{\sigma,\tau} \text{ is a forward query,} \\ \zeta^{\text{inv}}_{|\chi^{\sigma,\tau,(j)}\rangle,x} & \text{if the } j\text{-th query of } \mathcal{B}_{\sigma,\tau} \text{ is an inverse query;} \end{cases}$$

the right-hand side quantities are defined in Lemmas 6.3 and 6.4. In view of the relation between $\mathcal{B}_{\sigma,\tau}$ and $\mathcal{B}$ in Lemma 4.9, we can express the pre-query states of the former in terms of the pre-query states of the latter: we have $|\chi^{\sigma,\tau,(j)}\rangle \in \left\{ V_X^\sigma |\phi^{\sigma,\tau,(j)}\rangle, V_X^\sigma V_Y^\tau |\phi^{\sigma,\tau,(j)}\rangle \right\}$ if the j-th query of $\mathcal{B}_{\sigma,\tau}$ is a forward query, and otherwise $|\chi^{\sigma,\tau,(j)}\rangle \in \left\{ V_X^\tau |\phi^{\sigma,\tau,(j)}\rangle, V_X^\tau V_Y^\sigma |\phi^{\sigma,\tau,(j)}\rangle \right\}$. Using the unitary invariance of the norm, we see that

$$\left\| \big(I - |+_x\rangle\langle +_x|_{D_x}\big) |\chi^{\sigma,\tau,(j)}\rangle \right\| = \left\| \big(I - |+_x\rangle\langle +_x|_{D_x}\big) |\phi^{\sigma,\tau,(j)}\rangle \right\| \tag{6.10}$$

and

$$\zeta_x^{\sigma,\tau,(j)} = \begin{cases} \zeta_{V_X^\sigma |\phi^{\sigma,\tau,(j)}\rangle,x} & \text{if the } j\text{-th query of } \mathcal{B} \text{ is a forward query,} \\ \zeta^{\text{inv}}_{V_X^\tau |\phi^{\sigma,\tau,(j)}\rangle,x} & \text{if the } j\text{-th query of } \mathcal{B} \text{ is an inverse query.} \end{cases}$$

To prove the proposition, we need to upper bound $\mathbf{E}_{x,\sigma,\tau}\, \zeta^{\sigma,\tau,(j)}_{|\chi^{\sigma,\tau,(j)}\rangle,x}$. We distinguish the two cases:

(i) If the j-th query is a forward query, we have

$$\zeta_x^{\sigma,\tau,(j)} = \zeta_{V_X^\sigma |\phi^{\sigma,\tau,(j)}\rangle,x} = \frac{|R_x^{\sigma,\tau}|}{x} \| |\phi^{\sigma,\tau,(j)}_{\sigma^{-1}(x)}\rangle \|^2 + \frac{|R_x^{\sigma,\tau}|}{x^2 N} + \frac{1}{x} \sum_{z=1}^{x-1} \sum_{\pi_{x^c} : \pi_{x^c}(z) \in R_x^{\sigma,\tau}} \left\| \langle \pi_{x^c}|_{D_{x^c}} |\phi^{\sigma,\tau,(j)}_{\sigma^{-1}(z)}\rangle \right\|^2,$$

where $|\phi_\xi^{\sigma,\tau,(j)}\rangle_{BYD} := \langle \xi|_X |\phi^{\sigma,\tau,(j)}\rangle_{BXYD}$ (as in Lemma 6.3). We will now upper bound the average of the above term by term, beginning with the first term. Now, $p_\xi^{(j)} := \| |\phi_\xi^{\sigma,\tau,(j)}\rangle \|^2$ is a function of the reduced state of an algorithm that makes queries to the twirled standard oracle (the part of $\mathcal{B}$ right up to the j-th query), so it follows from Lemma 4.6 that this quantity is independent of $\sigma, \tau \in S_N$. Thus:

$$\mathop{\mathbf{E}}_{x,\sigma,\tau} \frac{|R_x^{\sigma,\tau}|}{x} \| |\phi^{\sigma,\tau,(j)}_{\sigma^{-1}(x)}\rangle \|^2 \leq \mathop{\mathbf{E}}_{x,\sigma,\tau} \frac{r_{\max}}{x} p^{(j)}_{\sigma^{-1}(x)} = \mathop{\mathbf{E}}_{x} \frac{r_{\max}}{x} \mathop{\mathbf{E}}_{\sigma,\tau} p^{(j)}_{\sigma^{-1}(x)} = \mathop{\mathbf{E}}_{x} \frac{r_{\max}}{x} \mathop{\mathbf{E}}_{x'} p^{(j)}_{x'}$$

$$\begin{aligned} &= r_{\max}\left(\frac{1}{N}\sum_{x=1}^{N}\frac{1}{x}\right)\left(\frac{1}{N}\sum_{x'=1}^{N}p_{x'}^{(j)}\right) \leq r_{\max}\frac{\ln(N)+1}{N}\frac{1}{N} \\ &= (\ln(N)+1)\frac{r_{\max}}{N^2} \end{aligned} \tag{6.11}$$

since $\sum_{x'=1}^{N} p_{x'}^{(j)} = \| |\phi^{\sigma,\tau,(j)}\rangle \|^2 = 1$. The second term can be bounded straightforwardly:

$$\mathop{\mathbf{E}}_{x,\sigma,\tau} \frac{|R_x^{\sigma,\tau}|}{x^2 N} \leq \frac{r_{\max}}{N} \mathop{\mathbf{E}}_{x} \frac{1}{x^2} = \frac{r_{\max}}{N^2}\sum_{x=1}^{N}\frac{1}{x^2} \leq \frac{\pi^2}{6}\frac{r_{\max}}{N^2} \leq 2\frac{r_{\max}}{N^2}. \tag{6.12}$$

We defer bounding the third term to Lemma 6.7 (i), where we get

$$\mathop{\mathbf{E}}_{x,\sigma,\tau} \frac{1}{x}\sum_{z=1}^{x-1} \sum_{\pi_{x^c}:\pi_{x^c}(z)\in R_x^{\sigma,\tau}} \left\| \langle \pi_{x^c}|_{D_{x^c}} |\phi^{\sigma,\tau,(j)}_{\sigma^{-1}(z)}\rangle \right\|^2 \leq (\ln(N)+3)\frac{r_{\max}}{N^2}. \tag{6.13}$$

Combining Eqs. (6.11) to (6.13), we obtain

$$\mathop{\mathbf{E}}_{x,\sigma,\tau} \zeta_x^{\sigma,\tau,(j)} \leq \big(2\ln(N)+6\big)\frac{r_{\max}}{N^2}. \tag{6.14}$$

(ii) If the j-th query is an inverse query, we have

$$\begin{aligned} \zeta_x^{\sigma,\tau,(j)} = \zeta^{\mathrm{inv}}_{V_X^{\tau}|\phi^{\sigma,\tau,(j)}\rangle,x} = &\frac{|R_x^{\sigma,\tau}|}{x} \| |\phi^{\sigma,\tau,(j)}_{\tau^{-1}(x)}\rangle\|^2 + \frac{|R_x^{\sigma,\tau}|}{x^2 N} + \frac{1}{x}\sum_{z\in R_x^{\sigma,\tau}} \sum_{\pi_{x^c}:\pi^{-1}_{>x}(z)<x} \left\| \langle \pi_{x^c}|_{D_{x^c}} |\phi^{\sigma,\tau,(j)}_{\tau^{-1}(z)}\rangle \right\|^2 \\ &+ \frac{1}{x}\sum_{z=x+1}^{N} \sum_{\pi_{x^c}:\pi^{-1}_{>x}(z)=x} |[x]\cap \pi^{-1}_{>x}(R_x^{\sigma,\tau})| \left\| \langle \pi_{x^c}|_{D_{x^c}} |\phi^{\sigma,\tau,(j)}_{\tau^{-1}(z)}\rangle \right\|^2, \end{aligned}$$

where $|\phi_\xi^{\sigma,\tau,(j)}\rangle_{BYD} := \langle \xi|_X |\phi^{\sigma,\tau,(j)}\rangle_{BXYD}$ (as in Lemma 6.4). The average can again be bounded term by term. For the first two terms we proceed as above and for the last two we use Lemma 6.7 (ii) and (iii). Altogether we obtain

$$\begin{aligned} \mathop{\mathbf{E}}_{x,\sigma,\tau} \zeta_x^{\sigma,\tau,(j)} &\leq (\ln(N)+1)\frac{r_{\max}}{N^2} + 2\frac{r_{\max}}{N^2} + (\ln(N)+1)\frac{r_{\max}}{N^2} + (\ln(N)+1)\frac{r_{\max}}{N^2} \\ &= \big(3\ln(N)+5\big)\frac{r_{\max}}{N^2}. \end{aligned} \tag{6.15}$$

From Eqs. (6.14) and (6.15) we see that in both the forward and the inverse case, we can bound

$$\mathop{\mathbf{E}}_{x,\sigma,\tau} \zeta_x^{\sigma,\tau,(j)} \leq 3\big(\ln(N)+2\big)\frac{r_{\max}}{N^2}$$

We can now use the above and Eq. (6.10) to further bound Eq. (6.9) and obtain

$$\begin{aligned} \mathop{\mathbf{E}}_{x,\sigma,\tau} \left\| E_D^{R^{\sigma,\tau},x} |\phi^{\sigma,\tau}\rangle \right\|^2 &\leq 64q\sum_{j=1}^{2q} \mathop{\mathbf{E}}_{x,\sigma,\tau} \zeta_x^{\sigma,\tau,(j)} + 4q\sum_{j=1}^{2q} \mathop{\mathbf{E}}_{x,\sigma,\tau} \frac{|R_x^{\sigma,\tau}|}{x} \left\| \big(I - |+_x\rangle\langle +_x|_{D_x}\big) |\chi^{\sigma,\tau,(j)}\rangle \right\|^2 \\ &\leq 384q^2\big(\ln(N)+2\big)\frac{r_{\max}}{N^2} + 4q\sum_{j=1}^{2q} \mathop{\mathbf{E}}_{x,\sigma,\tau} \frac{|R_x^{\sigma,\tau}|}{x} \left\| \big(I - |+_x\rangle\langle +_x|_{D_x}\big) |\phi^{\sigma,\tau,(j)}\rangle \right\|^2 \\ &= 384\frac{q^2 r_{\max}\big(\ln(N)+2\big)}{N^2} + 4q r_{\max}\sum_{j=1}^{2q} \mathop{\mathbf{E}}_{x,\sigma,\tau} \frac{\left\| \big(I - |+_x\rangle\langle +_x|_{D_x}\big) |\phi^{\sigma,\tau,(j)}\rangle \right\|^2}{x}, \end{aligned}$$

which is the desired result. □

Lemma 6.7. *In the situation of Proposition 6.6 and with* $|\phi_\xi^{\sigma,\tau,(j)}\rangle_{BYD} := \langle \xi|_X |\phi^{\sigma,\tau,(j)}\rangle_{BXYD}$, *we have:*

(i) $\mathbf{E}_{x \leftarrow [N],\, \sigma,\tau \leftarrow S_N} \frac{1}{x} \sum_{z=1}^{x-1} \sum_{\pi_{x^c} : \pi_{x^c}(z) \in R_x^{\sigma,\tau}} \left\| \langle \pi_{x^c} |_{D_{x^c}} | \phi^{\sigma,\tau,(j)}_{\sigma^{-1}(z)} \rangle \right\|^2 \leq (\ln(N) + 3) \frac{r_{\max}}{N^2}$.

(ii) $\mathbf{E}_{x \leftarrow [N],\, \sigma,\tau \leftarrow S_N} \frac{1}{x} \sum_{z \in R_x^{\sigma,\tau}} \sum_{\pi_{x^c} : \pi_{>x}^{-1}(z) < x} \left\| \langle \pi_{x^c} |_{D_{x^c}} | \phi^{\sigma,\tau,(j)}_{\tau^{-1}(z)} \rangle \right\|^2 \leq (\ln(N) + 1) \frac{r_{\max}}{N^2}$.

(iii) $\mathbf{E}_{x \leftarrow [N],\, \sigma,\tau \leftarrow S_N} \frac{1}{x} \sum_{z=x+1}^{N} \sum_{\pi_{x^c} : \pi_{>x}^{-1}(z) = x} |[x] \cap \pi_{>x}^{-1}(R_x^{\sigma,\tau})| \left\| \langle \pi_{x^c} |_{D_{x^c}} | \phi^{\sigma,\tau,(j)}_{\tau^{-1}(z)} \rangle \right\|^2 \leq (\ln(N) + 1) \frac{r_{\max}}{N^2}$.

Proof. Note that the quantity

$$q_{\omega,\xi} := \left\| \langle \tau\omega\sigma^{-1} |_D | \phi^{\sigma,\tau,(j)}_{\xi} \rangle \right\|^2 = \left\| \langle \tau\omega\sigma^{-1} |_D \langle \xi |_X | \phi^{\sigma,\tau,(j)} \rangle \right\|^2$$

can be interpreted as the joint probability of the outcomes of the following procedure: initialize the database, run an algorithm (namely, $\mathcal{B}$ up to right before its j-th query) that makes queries to the twirled standard oracle, measure the X register to obtain an outcome $\xi \in [N]$, and also apply the recovery operation to the database to obtain an outcome $\omega \in S_N$. Accordingly, Lemma 4.6 shows that $q_{\omega,\xi}$ does not depend on the choice of $\sigma, \tau \in S_N$ (which justifies the notation) and that the marginal distribution of ω with respect to $q_{\omega,\xi}$ is uniform, i.e., $\sum_{\xi \in [N]} q_{\omega,\xi} = \frac{1}{N!}$ for all $\omega \in S_N$. This observation will be used to establish all three parts of the lemma.

(i) We start by writing

$$\mathop{\mathbf{E}}_{x,\sigma,\tau} \frac{1}{x} \sum_{z=1}^{x-1} \sum_{\pi_{x^c} : \pi_{x^c}(z) \in R_x^{\sigma,\tau}} \left\| \langle \pi_{x^c} |_{D_{x^c}} | \phi^{\sigma,\tau,(j)}_{\sigma^{-1}(z)} \rangle \right\|^2 = \mathop{\mathbf{E}}_{x,\sigma,\tau} \frac{1}{x} \sum_{z=1}^{x-1} \sum_{\pi : \pi_{x^c}(z) \in R_x^{\sigma,\tau}} q_{\tau^{-1}\pi\sigma, \sigma^{-1}(z)}. \tag{6.16}$$

Recall that $\pi = \pi_{>x}\,(x\ t_x)\,\pi_{>x}$ for some $t_x \in [x]$, as in Eqs. (3.1) and (3.4). We will write $t_x(\pi) := t_x$ to make explicit the dependency of t_x on the permutation. Because $z < x$, we have

$$\pi_{x^c}(z) = \begin{cases} \pi(z) & \text{if } \pi_{<x}(z) \neq t_x(\pi), \\ \pi(x) & \text{if } \pi_{<x}(z) = t_x(\pi), \end{cases}$$

so we can write the right-hand side of Eq. (6.16) as a sum of two terms,

$$\begin{aligned}
&\mathop{\mathbf{E}}_{x,\sigma,\tau} \frac{1}{x} \sum_{z=1}^{x-1} \sum_{\pi : \pi_{x^c}(z) \in R_x^{\sigma,\tau}} q_{\tau^{-1}\pi\sigma, \sigma^{-1}(z)} \\
&= \mathop{\mathbf{E}}_{x,\sigma,\tau} \frac{1}{x} \sum_{z=1}^{x-1} \sum_{\pi : \pi(z) \in R_x^{\sigma,\tau}} \mathbf{1}_{\pi_{<x}(z) \neq t_x(\pi)}\, q_{\tau^{-1}\pi\sigma, \sigma^{-1}(z)} + \mathop{\mathbf{E}}_{x,\sigma,\tau} \frac{1}{x} \sum_{z=1}^{x-1} \sum_{\pi : \pi(x) \in R_x^{\sigma,\tau}} \mathbf{1}_{\pi_{<x}(z) = t_x(\pi)}\, q_{\tau^{-1}\pi\sigma, \sigma^{-1}(z)} \\
&= \mathop{\mathbf{E}}_{x',\sigma,\tau} \frac{1}{\sigma(x')} \sum_{z' \in \sigma^{-1}([1,\sigma(x')])} \sum_{\pi' : \pi'(z') \in R_{x'}} \mathbf{1}_{\pi_{<\sigma(x')}(\sigma(z')) \neq t_{\sigma(x')}(\tau\pi'\sigma^{-1})}\, q_{\pi',z'} \\
&+ \mathop{\mathbf{E}}_{x',\sigma,\tau} \frac{1}{\sigma(x')} \sum_{z' \in \sigma^{-1}([1,\sigma(x')])} \sum_{\pi' : \pi'(x') \in R_{x'}} \mathbf{1}_{\pi_{<\sigma(x')}(\sigma(z')) = t_{\sigma(x')}(\tau\pi'\sigma^{-1})}\, q_{\pi',z'},
\end{aligned}$$

where the last step follows by substituting $x = \sigma(x')$, $z = \sigma(z')$, and $\pi = \tau\pi'\sigma^{-1}$, noting that (x', σ, τ) is still uniformly random, and using the relation $\pi(\xi) \in R_x^{\sigma,\tau} \Leftrightarrow \tau^{-1}(\pi(\xi)) \in R_{\sigma^{-1}(x)}$. As τ only appears in the indicator functions, we can rewrite and bound this as

$$\begin{aligned}
&\mathop{\mathbf{E}}_{x',\sigma} \frac{1}{\sigma(x')} \sum_{z' \in \sigma^{-1}([1,\sigma(x')])} \sum_{\pi' : \pi'(z') \in R_{x'}} \Pr_{\tau}\big(\pi_{<\sigma(x')}(\sigma(z')) \neq t_{\sigma(x')}(\tau\pi'\sigma^{-1})\big)\, q_{\pi',z'} \\
&+ \mathop{\mathbf{E}}_{x',\sigma} \frac{1}{\sigma(x')} \sum_{z' \in \sigma^{-1}([1,\sigma(x')])} \sum_{\pi' : \pi'(x') \in R_{x'}} \Pr_{\tau}\big(\pi_{<\sigma(x')}(\sigma(z')) = t_{\sigma(x')}(\tau\pi'\sigma^{-1})\big)\, q_{\pi',z'} \\
&= \mathop{\mathbf{E}}_{x',\sigma} \frac{1}{\sigma(x')} \sum_{z' \in \sigma^{-1}([1,\sigma(x')])} \sum_{\pi' : \pi'(z') \in R_{x'}} \Pr_{t' \leftarrow [\sigma(x')]} \big(\pi_{<\sigma(x')}(\sigma(z')) \neq t'\big)\, q_{\pi',z'}
\end{aligned}$$

$$\begin{aligned}
&+ \mathop{\mathbf{E}}_{x',\sigma} \frac{1}{\sigma(x')} \sum_{z'\in\sigma^{-1}([1,\sigma(x')])} \sum_{\pi':\pi'(x')\in R_{x'}} \Pr_{t'\leftarrow[\sigma(x')]}\big(\pi_{<\sigma(x')}(\sigma(z')) = t'\big)\, q_{\pi',z'} \\
&\le \mathop{\mathbf{E}}_{x',\sigma} \frac{1}{\sigma(x')} \sum_{z'\in\sigma^{-1}([1,\sigma(x')])} \sum_{\pi':\pi'(z')\in R_{x'}} q_{\pi',z'} + \mathop{\mathbf{E}}_{x',\sigma} \frac{1}{\sigma(x')} \sum_{z'\in\sigma^{-1}([1,\sigma(x')])} \sum_{\pi':\pi'(x')\in R_{x'}} \frac{1}{\sigma(x')} q_{\pi',z'} \\
&= \mathop{\mathbf{E}}_{x',\sigma} \frac{1}{\sigma(x')} \sum_{z'\in\sigma^{-1}([1,\sigma(x')])} \sum_{\pi':\pi'(z')\in R_{x'}} q_{\pi',z'} + \mathop{\mathbf{E}}_{x',\sigma} \frac{1}{(\sigma(x'))^2} \sum_{z'\in\sigma^{-1}([1,\sigma(x')])} \sum_{\pi':\pi'(x')\in R_{x'}} q_{\pi',z'},
\end{aligned}$$

since, for any fixed x', z', σ, π', the permutation $\tau\pi'\sigma^{-1}$ is uniformly random in S_N, so $t_{\sigma(x')}(\tau\pi'\sigma^{-1})$ is uniformly random in $[\sigma(x')]$ (by Corollary 3.2) and hence equal to any fixed integer in this interval with probability $\frac{1}{\sigma(x')}$. We can finally upper bound the above by

$$\begin{aligned}
&\mathop{\mathbf{E}}_{x',\sigma} \frac{1}{\sigma(x')} \sum_{z'\in\sigma^{-1}([1,\sigma(x')])} \sum_{\pi':\pi'(z')\in R_{x'}} q_{\pi',z'} + \mathop{\mathbf{E}}_{x',\sigma} \frac{1}{(\sigma(x'))^2} \sum_{z'\in\sigma^{-1}([1,\sigma(x')])} \sum_{\pi':\pi'(x')\in R_{x'}} q_{\pi',z'} \\
&\le \mathop{\mathbf{E}}_{x',\sigma} \frac{1}{\sigma(x')} \sum_{z'=1}^{N} \sum_{\pi':\pi'(z')\in R_{x'}} q_{\pi',z'} + \mathop{\mathbf{E}}_{x',\sigma} \frac{1}{(\sigma(x'))^2} \sum_{z'=1}^{N} \sum_{\pi':\pi'(x')\in R_{x'}} q_{\pi',z'} \\
&= \mathop{\mathbf{E}}_{x'}\left(\mathop{\mathbf{E}}_{\sigma} \frac{1}{\sigma(x')}\right) \sum_{z'=1}^{N} \sum_{\pi':\pi'(z')\in R_{x'}} q_{\pi',z'} + \mathop{\mathbf{E}}_{x'}\left(\mathop{\mathbf{E}}_{\sigma} \frac{1}{\sigma(x')^2}\right) \sum_{z'=1}^{N} \sum_{\pi':\pi'(x')\in R_{x'}} q_{\pi',z'} \\
&= \mathop{\mathbf{E}}_{x'}\left(\mathop{\mathbf{E}}_{\sigma} \frac{1}{\sigma(x')}\right) \sum_{z'=1}^{N} \sum_{\pi':\pi'(z')\in R_{x'}} q_{\pi',z'} + \mathop{\mathbf{E}}_{x'}\left(\mathop{\mathbf{E}}_{\sigma} \frac{1}{\sigma(x')^2}\right) \sum_{z'=1}^{N} \sum_{\pi':\pi'(x')\in R_{x'}} q_{\pi',z'} \\
&\le \frac{\ln(N)+1}{N} \mathop{\mathbf{E}}_{x'} \sum_{z'=1}^{N} \sum_{\pi':\pi'(z')\in R_{x'}} q_{\pi',z'} + \frac{\pi^2}{6}\frac{1}{N} \mathop{\mathbf{E}}_{x'} \sum_{z'=1}^{N} \sum_{\pi':\pi'(x')\in R_{x'}} q_{\pi',z'} \\
&= \frac{\ln(N)+1}{N} \sum_{\pi'\in S_N} \sum_{z'=1}^{N} \Pr_{x'}\Big(x' \in R^{\mathrm{inv}}_{\pi'(z)}\Big) q_{\pi',z'} + \frac{\pi^2}{6}\frac{1}{N} \mathop{\mathbf{E}}_{x'} \Pr_{\pi'}(\pi'(x') \in R_{x'}) \\
&\le \frac{\ln(N)+1}{N}\frac{r_{\max}}{N} + \frac{\pi^2}{6}\frac{1}{N}\frac{r_{\max}}{N} \\
&\le (\ln(N)+3)\frac{r_{\max}}{N^2},
\end{aligned}$$

where we first enlarging the sum over z' to all of $[N]$, then we bounded the expectation over σ by using that $\sigma(x') \in [N]$ is uniformly random for any fixed x'; in the last equality we also used that the marginal distribution of π' with respect to $q_{\pi',z'}$ is uniform as discussed above.

(ii) Similarly as above, we begin by writing

$$\mathop{\mathbf{E}}_{x,\sigma,\tau} \frac{1}{x} \sum_{z\in R_x^{\sigma,\tau}} \sum_{\pi_{x^c}:\pi_{>x}^{-1}(z)<x} \left\| \langle \pi_{x^c}|_{D_{x^c}} |\phi^{\sigma,\tau,(j)}_{\tau^{-1}(z)}\rangle \right\|^2 = \mathop{\mathbf{E}}_{x,\sigma,\tau} \frac{1}{x} \sum_{z\in R_x^{\sigma,\tau}} \sum_{\pi:\pi_{>x}^{-1}(z)<x} q_{\tau^{-1}\pi\sigma,\tau^{-1}(z)}$$

We can upper-bound this by omitting the constraint on π, which gives the bound

$$\begin{aligned}
\mathop{\mathbf{E}}_{x,\sigma,\tau} \frac{1}{x} \sum_{z\in R_x^{\sigma,\tau}} \sum_{\pi:\pi_{>x}^{-1}(z)<x} q_{\tau^{-1}\pi\sigma,\tau^{-1}(z)} &\le \mathop{\mathbf{E}}_{x,\sigma,\tau} \frac{1}{x} \sum_{z\in R_x^{\sigma,\tau}} q_{\tau^{-1}(z)} = \mathop{\mathbf{E}}_{x',\sigma,\tau} \frac{1}{\sigma(x')} \sum_{z'\in R_{x'}} q_{z'} \\
&= \mathop{\mathbf{E}}_{x'}\left(\mathop{\mathbf{E}}_{\sigma} \frac{1}{\sigma(x')}\right) \sum_{z'\in R_{x'}} q_{z'} \le \frac{\ln(N)+1}{N} \mathop{\mathbf{E}}_{x'} \sum_{z'\in R_{x'}} q_{z'} \\
&= \frac{\ln(N)+1}{N} \sum_{z'=1}^{N} \Pr_{x'}\big(x' \in R^{\mathrm{inv}}_{z'}\big) q_{z'} \le (\ln(N)+1)\frac{r_{\max}}{N^2},
\end{aligned}$$

where we use the notation $q_\xi := \sum_{\omega\in S_N} q_{\omega,\xi}$ for the marginal distribution of ξ with respect to $q_{\omega,\xi}$; the second step follows by substituting $x = \sigma(x')$ and $z = \tau(z')$.

(iii) Again we begin with

$$\begin{aligned}
&\mathop{\mathbf{E}}_{x,\sigma,\tau} \frac{1}{x} \sum_{z=x+1}^{N} \sum_{\pi_{x^c}:\pi_{>x}^{-1}(z)=x} \left|[x] \cap \pi_{>x}^{-1}(R_x^{\sigma,\tau})\right| \left\| \langle \pi_{x^c}|_{D_{x^c}} \, |\phi_{\tau^{-1}(z)}^{\sigma,\tau,(j)}\rangle \right\|^2 \\
&= \mathop{\mathbf{E}}_{x,\sigma,\tau} \frac{1}{x} \sum_{z=x+1}^{N} \sum_{\pi:\pi_{>x}^{-1}(z)=x} \left|[x] \cap \pi_{>x}^{-1}(R_x^{\sigma,\tau})\right| q_{\tau^{-1}\pi\sigma,\tau^{-1}(z)} \\
&\leq r_{\max} \mathop{\mathbf{E}}_{x,\sigma,\tau} \frac{1}{x} \sum_{z=x+1}^{N} \sum_{\pi:\pi_{>x}^{-1}(z)=x} q_{\tau^{-1}\pi\sigma,\tau^{-1}(z)} \\
&= r_{\max} \mathop{\mathbf{E}}_{x',\sigma,\tau} \frac{1}{\tau(x')} \sum_{z'\in\tau^{-1}(\{\tau(x')+1,\dots,N\})} \sum_{\pi'} \mathbf{1}_{((\tau\pi'\sigma^{-1})_{>\tau(x')})^{-1}(\tau(z'))=\tau(x')} q_{\pi',z'},
\end{aligned}$$

where the last step follows by substituting $x = \tau(x')$, $z = \tau(z')$, and $\pi = \tau\pi'\sigma^{-1}$. As σ only occurs in the indicator function, we can rewrite and bound this as

$$\begin{aligned}
&r_{\max} \mathop{\mathbf{E}}_{x',\tau} \frac{1}{\tau(x')} \sum_{z'\in\tau^{-1}(\{\tau(x')+1,\dots,N\})} \sum_{\pi'} \Pr_{\sigma}\Big(((\tau\pi'\sigma^{-1})_{>\tau(x')})^{-1}(\tau(z')) = \tau(x')\Big) q_{\pi',z'} \\
&= r_{\max} \mathop{\mathbf{E}}_{x',\tau} \frac{1}{\tau(x')} \sum_{z'\in\tau^{-1}(\{\tau(x')+1,\dots,N\})} \sum_{\pi'} \Pr_{\sigma}\Big(\sigma_{>\tau(x')}(\tau(x')) = \tau(z')\Big) q_{\pi',z'} \qquad (6.17)
\end{aligned}$$

since, for any fixed τ and π', the permutation $\tau\pi'\sigma^{-1}$ is again uniformly random. Using part (iii) of Lemma 3.3, we see that the inner probability is simply equal to $\frac{1}{N}$. Hence the above is equal to

$$\frac{r_{\max}}{N} \mathop{\mathbf{E}}_{x',\tau} \frac{1}{\tau(x')} \sum_{z'\in\tau^{-1}(\{\tau(x')+1,\dots,N\})} \sum_{\pi'} q_{\pi',z'} \leq \frac{r_{\max}}{N} \mathop{\mathbf{E}}_{x',\tau} \frac{1}{\tau(x')} = \frac{r_{\max}}{N} \mathop{\mathbf{E}}_{x'} \frac{1}{x'} \leq (\ln(N)+1)\frac{r_{\max}}{N^2}.$$

□

6.3 Sparsity Analysis

The goal of this section is to upper bound the term

$$\mathop{\mathbf{E}}_{\substack{x\leftarrow[N],\\ \sigma,\tau\leftarrow S_N}} \frac{\left\| \left(I - |+_x\rangle\langle +_x|_{D_x}\right) |\phi^{\sigma,\tau,(j)}\rangle \right\|^2}{x}, \qquad (6.18)$$

which remains to be estimated in the right-hand side of Proposition 6.6. Intuitively, this quantifies the extent to which a random database register D_x has been queried by the algorithm, weighted by $1/x$-.

To analyze Eq. (6.18), recall that $|\phi^{\sigma,\tau,(j)}\rangle_{AXYD}$ denotes the joint state of the algorithm and database right before the j-th query when run with the twirled oracle. By Lemma 4.8, we have

$$|\phi^{\sigma,\tau,(j)}\rangle_{AXYD} = L_D^{\tau} R_D^{\sigma} \, |\phi^{(j)}\rangle_{AXYD},$$

where $|\phi^{(j)}\rangle$ denotes the state right before the j-th query when the same algorithm is run with the *untwirled* oracle. We can thus express Eq. (6.18) as follows:

$$\mathop{\mathbf{E}}_{\substack{x\leftarrow[N],\\ \sigma,\tau\leftarrow S_N}} \frac{\left\| \left(I - |+_x\rangle\langle +_x|_{D_x}\right) |\phi^{\sigma,\tau,(j)}\rangle \right\|^2}{x} = \langle \phi^{(j)} | \, \Gamma_D \, |\phi^{(j)}\rangle, \qquad (6.19)$$

where we have introduced the operator

$$\Gamma_D := \mathop{\mathbf{E}}_{x\leftarrow[N]} \frac{1}{x} \mathop{\mathbf{E}}_{\sigma,\tau\leftarrow S_N} (L_D^{\tau} R_D^{\sigma})^{\dagger} \left(I - |+_x\rangle\langle +_x|_{D_x}\right) (L_D^{\tau} R_D^{\sigma}) \qquad (6.20)$$

where we recall that H_N denotes the harmonic numbers, see Eq. (2.1).

To upper bound the quantity of interest, we now observe that that we can upper bound its growth with each additional query as follows, in terms of the norm of a commutator:

$$\begin{aligned}\langle\phi^{(j+1)}|\Gamma_D|\phi^{(j+1)}\rangle - \langle\phi^{(j)}|\Gamma_D|\phi^{(j)}\rangle &= \langle\phi^{(j)}|Q^\dagger_{XYD}\Gamma_D Q_{XYD} - \Gamma_D|\phi^{(j)}\rangle \\ &= \langle\phi^{(j)}|Q^\dagger_{XYD}[\Gamma_D, Q_{XYD}]|\phi^{(j)}\rangle \\ &\leq \|[\Gamma_D, Q_{XYD}]\|\end{aligned} \tag{6.21}$$

where $Q_{XYD} \in \{O^{\mathsf{SPO}}_{XYD}, O^{\mathsf{SPO,inv}}_{XYD}\}$, depending on whether the j-th query is a forward or an inverse query. The first equality holds because the unitary that the algorithm performs inbetween the two queries does not act on the oracle's database register D.

We now calculate the operator Γ_D explicitly and use the result to estimate the norm of the commutator.

Lemma 6.8. *We have*

$$\Gamma_D = \frac{H_N - H_N^{(2)}}{N} I_D - 2\frac{H_N^{(2)} - H_N^{(3)}}{N} W_D^{(2)} - \frac{H_N - 3H_N^{(2)} + 2H_N^{(3)}}{N} W_D^{(3)},$$

where we denote $H_N^{(\ell)} := \sum_{\ell=1}^N \frac{1}{x^\ell}$ *and* $W^{(\ell)} := \mathbf{E}_{\gamma\ \ell\text{-cycle}} R^\gamma = \mathbf{E}_{\gamma\ \ell\text{-cycle}} L^\gamma$.[5]

Proof. We first compute the action of $|{+_x}\rangle\langle{+_x}|_{D_x}$ in the permutation basis. For any $\pi \in S_N$, we have

$$\begin{aligned}|{+_x}\rangle\langle{+_x}|_{D_x}\,|\pi\rangle_D &= \mathop{\mathbf{E}}_{s\leftarrow[x]} |\pi_{>x}\,(x\ s)\,\pi_{<x}\rangle \\ &= \mathop{\mathbf{E}}_{s\leftarrow[x]} |\pi_{>x}\,(x\ t_x)\,\pi_{<x}\,\pi_{<x}^{-1}\,(x\ t_x)\,\pi_{<x}\,\pi_{<x}^{-1}\,(x\ s)\,\pi_{<x}\rangle \\ &= \mathop{\mathbf{E}}_{s\leftarrow[x]} |\pi\,\big(x\ \pi_{<x}^{-1}(t_x)\big)\,\big(x\ \pi_{<x}^{-1}(s)\big)\rangle \\ &= \mathop{\mathbf{E}}_{s\leftarrow[x]} R_D^{(x\ s)} R_D^{\left(x\ \pi_{<x}^{-1}(t_x(\pi))\right)}\,|\pi\rangle_D\,,\end{aligned}$$

where we denote by $t_x \in [x]$ the number in the decomposition (3.1) of π, i.e., $\pi = \pi_{>x}\,(x\ t_x)\,\pi_{<x}$; in the last line we write $t_x(\pi)$ to make the dependence on π explicit. Thus,

$$|{+_x}\rangle\langle{+_x}|_{D_x} = \sum_{\pi\in S_N} \mathop{\mathbf{E}}_{s\leftarrow[x]} R_D^{(x\ s)} R_D^{\left(x\ \pi_{<x}^{-1}(t_x(\pi))\right)}\,|\pi\rangle_D\,\langle\pi|_D\,.$$

We first average this over the left action, which commutes with the right action, and obtain

$$\begin{aligned}\mathop{\mathbf{E}}_{\tau\leftarrow S_N} (L_D^\tau)^\dagger |{+_x}\rangle\langle{+_x}|_{D_x} L_D^\tau &= \sum_{\pi\in S_N} \mathop{\mathbf{E}}_{s\leftarrow[x]} R_D^{(x\ s)} R_D^{\left(x\ \pi_{<x}^{-1}(t_x(\pi))\right)} \underbrace{\mathop{\mathbf{E}}_{\tau\leftarrow S_N} |\tau^{-1}\pi\rangle_D\,\langle\tau^{-1}\pi|_D}_{=\frac{1}{N!}I_D} \\ &= \mathop{\mathbf{E}}_{\pi\in S_N} \mathop{\mathbf{E}}_{s\leftarrow[x]} R_D^{(x\ s)} R_D^{\left(x\ \pi_{<x}^{-1}(t_x(\pi))\right)} = \mathop{\mathbf{E}}_{s,t\leftarrow[x]} R_D^{(x\ s)} R_D^{(x\ t)} = \mathop{\mathbf{E}}_{s,t\leftarrow[x]} R_D^{(x\ s)(x\ t)}.\end{aligned}$$

If we now average over the right action, the permutation $(x\ s)\,(x\ t)$ is conjugated into a random permutation of the same type (either the identity, a transposition, or a 3-cycle, depending on the cardinality of $\{x, s, t\}$):

$$\begin{aligned}\mathop{\mathbf{E}}_{\sigma,\tau\leftarrow S_N} (L_D^\tau R_D^\sigma)^\dagger |{+_x}\rangle\langle{+_x}|_{D_x} (L_D^\tau R_D^\sigma) &= \mathop{\mathbf{E}}_{s,t\leftarrow[x]} \mathop{\mathbf{E}}_{\sigma\leftarrow S_N} R_D^{\left(\sigma^{-1}(x)\ \sigma^{-1}(s)\right)\left(\sigma^{-1}(x)\ \sigma^{-1}(t)\right)} \\ &= \sum_{\ell=1}^{3} \Pr_{s,t\leftarrow[x]}\big(|\{s,t,x\}| = \ell\big)\,W_D^{(\ell)} \\ &= \frac{1}{x} I_D + \frac{2(x-1)}{x^2} W_D^{(2)} + \frac{(x-1)(x-2)}{x^2} W_D^{(3)}.\end{aligned}$$

[5] To see this, note that $\mathbf{E}_\gamma R^\gamma|\pi\rangle = \mathbf{E}_\gamma|\pi\gamma^{-1}\rangle = \mathbf{E}_\gamma|\pi\gamma^{-1}\pi^{-1}\pi\rangle = \mathbf{E}_\gamma|\gamma\pi\rangle = \mathbf{E}_\gamma L_\gamma|\pi\rangle$, since if γ is a uniformly random ℓ-cycle then so is $\pi\gamma^{-1}\pi^{-1}$, for any permutation $\pi \in S_N$.

and finally, using Eq. (6.20) and $\mathbf{E}_{x\leftarrow[N]}\frac{1}{x^\ell} = H_N^{(\ell)}/N$,

$$\begin{aligned}\Gamma_D &= \mathop{\mathbf{E}}_{x\leftarrow[N]} \frac{1}{x}\left(I_D - \mathop{\mathbf{E}}_{\sigma,\tau\leftarrow S_N} (L_D^\tau R_D^\sigma)^\dagger |{+_x}\rangle\langle{+_x}|_{D_x} (L_D^\tau R_D^\sigma)\right) \\ &= \mathop{\mathbf{E}}_{x\leftarrow[N]} \left(\left(\frac{1}{x} - \frac{1}{x^2}\right) I_D - \frac{2(x-1)}{x^3} W_D^{(2)} - \frac{(x-1)(x-2)}{x^3} W_D^{(3)}\right).\end{aligned}$$ □

Lemma 6.9. *For $Q_{XYD} \in \{O_{XYD}^{\mathsf{SPO}}, O_{XYD}^{\mathsf{SPO},\mathrm{inv}}\}$, we have $\|[\Gamma_D, Q_{XYD}]\| \leq \frac{6(\ln(N)+1)}{N^2}$.*

Proof. We prove the bound in the case that $Q_{XYD} \in \{O_{XYD}^{\mathsf{SPO}}$ – the other case is identical except for using the formula $W^{(\ell)}$ in terms of the left instead of the right action. We first observe that since O_{XYD}^{SPO} is controlled on X,

$$\left\|[\Gamma_D, O_{XYD}^{\mathsf{SPO}}]\right\| = \max_{x\in[N]} \left\|[\Gamma_D, O_{YD}^{\mathsf{SPO},x}]\right\|$$

where $O_{YD}^{\mathsf{SPO},z} := \langle z|_X\, O_{XYD}^{\mathsf{SPO}}\, |z\rangle_X$. Next, note that if $\gamma \in S_N$ is any permutation such that $\gamma(x) = x$, then

$$[R_D^\gamma, O_{YD}^{\mathsf{SPO},x}] = 0.$$

since for any $y \in [N]$ and $\pi \in S_N$ we have $O_{YD}^{\mathsf{SPO},x} R_D^\gamma |y,\pi\rangle_{YD} = |y \oplus \pi(\gamma^{-1}(x)), \pi\gamma^{-1}\rangle_{YD} = |y \oplus \pi(x), \pi\gamma^{-1}\rangle_{YD} = R_D^\gamma O_{YD}^{\mathsf{SPO},x} |y,\pi\rangle_{YD}$. Otherwise, if $\gamma(x) \neq x$ then it still holds that $\|[R_D^\gamma, O_{YD}^{\mathsf{SPO},x}]\| \leq 2$ since the commutator of any two unitaries has operator norm at most two. Accordingly,

$$\begin{aligned}\left\|[W_D^{(2)}, O_{YD}^{\mathsf{SPO},x}]\right\| &\leq 2 \Pr_{\gamma\ \text{2-cycle}}(\gamma(x) \neq x) = \frac{4}{N}, \\ \left\|[W_D^{(3)}, O_{YD}^{\mathsf{SPO},x}]\right\| &\leq 2 \Pr_{\gamma\ \text{3-cycle}}(\gamma(x) \neq x) = \frac{6}{N}.\end{aligned}$$

and hence, using Lemma 6.8, the fact that $H_N \geq H_N^{(2)} \geq H_N^{(3)} \geq 0$, and Eq. (2.2),

$$\begin{aligned}\left\|[\Gamma_D, O_{YD}^{\mathsf{SPO},x}]\right\| &\leq 2\frac{H_N^{(2)} - H_N^{(3)}}{N}\left\|[W_D^{(2)}, O_{YD}^{\mathsf{SPO},x}]\right\| + \frac{H_N - 3H_N^{(2)} + 2H_N^{(3)}}{N}\left\|[W_D^{(3)}, O_{YD}^{\mathsf{SPO},x}]\right\| \\ &\leq 8\frac{H_N^{(2)} - H_N^{(3)}}{N^2} + 6\frac{H_N - 3H_N^{(2)} + 2H_N^{(3)}}{N^2} \\ &= \frac{6H_N - 10H_N^{(2)} + 4H_N^{(3)}}{N^2} \leq \frac{6H_N}{N^2} \leq \frac{6(\ln(N)+1)}{N^2}.\end{aligned}$$

□

Corollary 6.10. *For all j, it holds that*

$$\mathop{\mathbf{E}}_{\substack{x\leftarrow[N],\\ \sigma,\tau\leftarrow S_N}} \frac{\left\|\left(I - |{+_x}\rangle\langle{+_x}|_{D_x}\right)|\phi^{\sigma,\tau,(j)}\rangle\right\|^2}{x} \leq 6\,\frac{j(\ln(N)+1)}{N^2}$$

and hence

$$4qr_{\max}\sum_{j=1}^{2q} \mathop{\mathbf{E}}_{\substack{x\leftarrow[N],\\ \sigma,\tau\leftarrow S_N}} \frac{\left\|\left(I - |{+_x}\rangle\langle{+_x}|_{D_x}\right)|\phi^{\sigma,\tau,(j)}\rangle\right\|^2}{x} \leq 72\frac{q^3 r_{\max}(\ln(N)+1)}{N^2}$$

Proof. The first formula follows from Eqs. (6.19) and (6.21) and Lemma 6.9 by using induction, since $\langle\phi^{(0)}|\,\Gamma_D\,|\phi^{(0)}\rangle = \langle\Phi_{\mathsf{SPO}}|\,\Gamma_D\,|\Phi_{\mathsf{SPO}}\rangle = 0$. The second formula follows at once. □

6.4 Main Theorem

Finally, we can use the preceding analysis, together with the fundamental lemma, to establish our main theorem (which formalizes Theorem 1.1 announced in the introduction):

Theorem 6.11 (Search)**.** *Let $\mathcal{A}$ be a quantum algorithm with quantum query access to a random permutation $\pi \in S_N$ and its inverse (Definition 4.1), which returns an $x \in [N]$, and let $R \subseteq [N] \times [N]$ be any relation. If $\mathcal{A}$ makes fewer than q queries, then the probability that it returns an element x such that $(x, \pi(x)) \in R$ is*

$$\Pr_{\pi \leftarrow S_N,\, x \leftarrow \mathcal{A}^{U^\pi, U^{\pi^{-1}}}} \big[(x, \pi(x)) \in R\big] \leq 914\, \frac{q^3 r_{\max}\big(\ln(N)+2\big)}{N},$$

where we recall $r_{\max} = \max\big\{\max_x |R_x|, \max_y |R_y^{\mathrm{inv}}|\big\}$, with $R_x = \{y : (x,y) \in R\}$ and $R_y^{\mathrm{inv}} = \{x : (x,y) \in R\}$.

Proof. Without loss of generality we can assume that $\mathcal{A}$ is a unitary query algorithm on registers AXY, where X and Y are the two N-dimensional registers that the oracles get applied to such that the classical outcome x can be obtained by measuring the X register. Let $\mathcal{B}$ denote the unitary query algorithm that first runs $x \leftarrow \mathcal{A}$, then makes one more query to load $\pi(x)$ into the Y register. Since $\mathcal{A}$ makes fewer than q queries, the algorithm $\mathcal{B}$ makes at most q queries. For every $\sigma, \tau \in S_N$, let $|\phi^{\sigma,\tau}\rangle_{AXYD}$ be the joint state of algorithm and oracle defined by running $\mathsf{Init}_D^{\mathsf{SPO}}$ and then $\mathcal{B}^{\mathsf{TSPO}_D^{\sigma,\tau}}$. Then, Proposition 6.6 and Corollary 6.10 combine to

$$\begin{aligned}\mathop{\mathbf{E}}_{\substack{x\leftarrow[N],\\ \sigma,\tau\leftarrow S_N}} \left\| E_D^{R^{\sigma,\tau},x} |\phi^{\sigma,\tau}\rangle \right\|^2 &\leq 384\frac{q^2 r_{\max}\big(\ln(N)+2\big)}{N^2} + 72\frac{q^3 r_{\max}\big(\ln(N)+1\big)}{N^2} \\ &\leq 456\frac{q^3 r_{\max}\big(\ln(N)+2\big)}{N^2}.\end{aligned}$$

Using Lemmas 5.3 and 6.1, we can upper bound the quantity $p_{\text{(ii)}}$ defined in the fundamental lemma (Lemma 5.1) as follows:

$$p_{\text{(ii)}} \leq N \mathop{\mathbf{E}}_{\substack{x\leftarrow[N],\\ \sigma,\tau\leftarrow S_N}} \left\| E_D^{R^{\sigma,\tau},x} |\phi^{\sigma,\tau}\rangle \right\|^2 \leq 456\frac{q^3 r_{\max}\big(\ln(N)+2\big)}{N}.$$

Finally, the fundamental lemma states that

$$\sqrt{p_{\text{(i)}}} \leq \sqrt{p_{\text{(ii)}}} + \sqrt{\frac{\ln(N)+1}{N}}$$

and hence

$$p_{\text{(i)}} \leq 2\left(p_{\text{(ii)}} + \frac{\ln(N)+1}{N}\right) \leq 914\, \frac{q^3 r_{\max}\big(\ln(N)+2\big)}{N}$$

concluding our proof. □

7 Application to One-Round Sponge and Unruh's Conjecture

We can now apply our results to obtain bounds for the hardness of search problems for algorithms given quantum query access to a random permutation and its inverse.

We first show a bound on the pre-image search problem for the *sponge construction*, instantiated with a random permutation, restricted to one absorption round and one squeezing round. The sponge function in this special case, for a permutation $\pi \in S_{\{0,1\}^n}$ and with capacity $c < n$, is given by

$$f_\pi \colon \{0,1\}^{n-c} \to \{0,1\}^{n-c}, \quad f_\pi(x) = \pi(x \| 0^c)_{[1,n-c]},$$

where $s_{[1,r]}$ denotes the first r bits of a string s. The following result was stated as Corollary 1.2 in the introduction.

Corollary 7.1 (One-round sponge)**.** *For any $y \in \{0,1\}^{n-c}$, the probability that a quantum algorithm $\mathcal{A}$ with quantum query access to a random permutation $\pi \in S_{\{0,1\}^n}$ and its inverse returns a preimage $x \in \{0,1\}^{n-c}$ under the one-round sponge function f_π, by making fewer than q queries, can be upper bounded as*

$$\Pr_{\pi \leftarrow S_{\{0,1\}^n},\, x \leftarrow \mathcal{A}^{U^\pi, U^{\pi^{-1}}}} \big[f_\pi(x) = y\big] \leq 914\, \frac{q^3(n+2)}{2^{\min(c,n-c)}}.$$

Proof. Let $\mathcal{B}$ denote the algorithm that runs $x \leftarrow \mathcal{A}$ and returns $x' := x \| 0^c$. Clearly, $f_\pi(x) = y$ if and only if $(x', \pi(x')) \in R$, where

$$R = \Big\{(x', y') \in \{0,1\}^n \times \{0,1\}^n \ : \ x'_{[n-c+1,n]} = 0^c,\ y'_{[1,n-c]} = y\Big\}.$$

Note that

$$r_{\max} = \max\Big\{\max_{x'} |R_{x'}|, \max_{y'} |R^{\mathrm{inv}}_{y'}|\Big\} = 2^{\max(c,n-c)}.$$

Applying Theorem 6.11 to $\mathcal{B}$, which makes the same number of queries as $\mathcal{A}$, we find that

$$\Pr_{\substack{\pi \leftarrow S_{\{0,1\}^n},\\ x \leftarrow \mathcal{A}^{U^\pi, U^{\pi^{-1}}}}} \big[f_\pi(x) = y\big] = \Pr_{\substack{\pi \leftarrow S_{\{0,1\}^n},\\ x' \leftarrow \mathcal{B}^{U^\pi, U^{\pi^{-1}}}}} \big[(x', \pi(x')) \in R\big] \leq 914\, \frac{q^3 2^{\max(c,n-c)}(n+2)}{2^n} = 914\, \frac{q^3(n+2)}{2^{\min(c,n-c)}}. \qquad \square$$

Next we consider the *double-sided zero-search conjecture*, which states that no adversary making polynomially many quantum queries to a permutation $\pi \in S_{\{0,1\}^{2n}}$ and its inverse is able to find $x \in \{0,1\}^n$ such that $\pi(x\|0^n)_{[n+1,2n]} = 0^n$ with non-negligible probability [Unr23, Conjecture 1]. The following corollary confirms Unruh's conjecture and establishes more generally an upper bound on the success probability for the problem with an arbitrary number c of zeros. It was stated as Corollary 1.3 in the introduction.

Corollary 7.2. *The probability that a quantum algorithm $\mathcal{A}$ with quantum query access to a random permutation $\pi \in S_{\{0,1\}^{2n}}$ and its inverse returns $x \in \{0,1\}^n$ such that $\pi(x\|0^n)_{[n+1,2n]} = 0^n$, by making fewer than q queries, can be upper bounded as*

$$\Pr_{\pi \leftarrow S_{\{0,1\}^{2n}},\, x \leftarrow \mathcal{A}^{U^\pi, U^{\pi^{-1}}}} \big[\exists y \in \{0,1\}^n : \pi(x\|0^n) = y\|0^n\big] \leq 1828\, \frac{q^3(n+1)}{2^n}.$$

More generally, it holds for any $c \in [2n]$ and any algorithm $\mathcal{A}$ that returns bitstrings $x \in \{0,1\}^{2n-c}$ that

$$\Pr_{\pi \leftarrow S_{\{0,1\}^{2n}},\, x \leftarrow \mathcal{A}^{U^\pi, U^{\pi^{-1}}}} \big[\exists y \in \{0,1\}^{2n-c} : \pi(x\|0^c) = y\|0^c\big] \leq 1828\, \frac{q^3(n+1)}{2^c}.$$

Proof. It suffices to establish the second claim since it implies the first for $c = n$. Similarly to the proof of Corollary 7.1 we can reduce to a relation, namely

$$R = \Big\{(x', y') \in \{0,1\}^{2n} \times \{0,1\}^{2n} \ : \ x'_{[2n-c+1,2n]} = y'_{[2n-c+1,2n]} = 0^c\Big\}.$$

Note that $r_{\max} = 2^{2n-c}$. Thus, Theorem 6.11 yields, with $N = 2^{2n}$,

$$\Pr_{\pi \leftarrow S_{\{0,1\}^{2n}},\, x \leftarrow \mathcal{A}^{U^\pi, U^{\pi^{-1}}}} \big[\exists y \in \{0,1\}^{2n-c} : \pi(x\|0^c) = y\|0^c\big] \leq 914\, \frac{q^3 2^{2n-c}(2n+2)}{2^{2n}} = 1828\, \frac{q^3(n+1)}{2^c}. \qquad \square$$

Acknowledgments

The authors thank Saliha Tokat for feedback on an earlier version of this manuscript.

GM and MW acknowledge support by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2092 CASA - 390781972. GM also acknowledges support by the European Research Council through an ERC Starting Grant (grant agreement No. 101077455, ObfusQation). MW also acknowledges support by the European Research Council through an ERC Starting Grant (grant agreement No. 101040907, SYMOPTIC), by the NWO through grant OCENW.KLEIN.267, and by the BMBF through project Quantum Methods and Benchmarks for Resource Allocation (QuBRA). CM acknowledges support by the Independent Research Fund Denmark via a DFF Sapere Aude grant (IM-3PQC, grant ID 10.46540/2064-00034B). MW acknowledges the Simons Institute for the Theory of Computing at UC Berkeley for its hospitality and support. MW and CM thank the Leibniz Center for Informatics for its hospitality.

References

[ABK+] Gorjan Alagic, Chen Bai, Jonathan Katz, Christian Majenz, and Patrick Struck. Post-quantum security of tweakable Even-Mansour, and applications. Cryptology ePrint Archive, Paper 2022/1097. Accepted at EUROCRYPT 2024. URL: https://eprint.iacr.org/2022/1097.

[ABKM22] Gorjan Alagic, Chen Bai, Jonathan Katz, and Christian Majenz. Post-quantum security of the Even-Mansour cipher. In *Advances in Cryptology—Eurocrypt 2022, Part III*, volume 13277 of *LNCS*, pages 458–487. Springer, 2022.

[ABPS23] Gorjan Alagic, Chen Bai, Alexander Poremba, and Kaiyan Shi. On the two-sided permutation inversion problem. Cryptology ePrint Archive, Paper 2023/985, 2023. URL: https://eprint.iacr.org/2023/985.

[BBBV97] Charles H. Bennett, Ethan Bernstein, Gilles Brassard, and Umesh V. Vazirani. Strengths and weaknesses of quantum computing. *SIAM J. Comput.*, 26(5):1510–1523, 1997.

[BDF+11] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In *Advances in Cryptology–ASIACRYPT 2011*, pages 41–69. Springer, 2011.

[BHH+19] Nina Bindel, Mike Hamburg, Kathrin Hövelmanns, Andreas Hülsing, and Edoardo Persichetti. Tighter proofs of CCA security in the quantum random oracle model. In *Theory of Cryptography. TCC 2019*, volume 11892 of *Lecture Notes in Computer Science*, pages 61–90. Springer, 2019.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. In *Proceedings of the 1st ACM Conference on Computer and Communications Security*, CCS '93, pages 62–73, New York, NY, USA, 1993. ACM.

[CFHL21] Kai-Min Chung, Serge Fehr, Yu-Hsuan Huang, and Tai-Ning Liao. On the compressed-oracle technique, and post-quantum security of proofs of sequential work. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology – EUROCRYPT 2021*, pages 598–629, Cham, 2021. Springer International Publishing.

[CGH04] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. *Journal of the ACM*, 51:557–594, July 2004.

[CMSZ19] Jan Czajkowski, Christian Majenz, Christian Schaffner, and Sebastian Zur. Quantum lazy sampling and game-playing proofs for quantum indifferentiability. Cryptology ePrint Archive, Report 2019/428, 2019. URL: https://eprint.iacr.org/2019/428.

[CP24] Joseph Carolan and Alexander Poremba. Quantum one-wayness of the single-round sponge with invertible permutations. Cryptology ePrint Archive, Paper 2024/414, 2024. https://eprint.iacr.org/2024/414. URL: https://eprint.iacr.org/2024/414.

[Cza21] Jan Czajkowski. Quantum indifferentiability of SHA-3. Cryptology ePrint Archive, Paper 2021/192, 2021. URL: https://eprint.iacr.org/2021/192.

[DFMS21] Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. Online-extractability in the quantum random-oracle model. Cryptology ePrint Archive, Report 2021/280, 2021. URL: https://eprint.iacr.org/2021/280.

[Dwo15] Morris J Dworkin. SHA-3 standard: Permutation-based hash and extendable-output functions. Federal Inf. Process. Stds. (NIST FIPS) 202, 2015. URL: https://doi.org/10.6028/NIST.FIPS.202.

[HHM22] Kathrin Hövelmanns, Andreas Hülsing, and Christian Majenz. Failing gracefully: Decryption failures and the Fujisaki-okamoto transform. In Shweta Agrawal and Dongdai Lin, editors, *Advances in Cryptology – ASIACRYPT 2022*, pages 414–443, Cham, 2022. Springer Nature Switzerland.

[HM23] Yassine Hamoudi and Frédéric Magniez. Quantum time-space tradeoff for finding multiple collision pairs. *ACM Trans. Comput. Theory*, 15, 2023.

[LR88] Michael Luby and Charles Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM J. Comput.*, 17(2):337–386, 1988.

[LZ19a] Qipeng Liu and Mark Zhandry. On finding quantum multi-collisions. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019*, pages 189–218, Cham, 2019. Springer International Publishing.

[LZ19b] Qipeng Liu and Mark Zhandry. Revisiting post-quantum Fiat-Shamir. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019*, pages 326–355, Cham, 2019. Springer International Publishing.

[NC00] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information.* Cambridge University Press, 2000.

[Ros21] Ansis Rosmanis. Tight bounds for inverting permutations via compressed oracle arguments. 2021. `arXiv:2103.08975`.

[Unr21] Dominique Unruh. Compressed permutation oracles (and the collision-resistance of Sponge/SHA3). Cryptology ePrint Archive, Paper 2021/062, 2021. URL: https://eprint.iacr.org/2021/062.

[Unr23] Dominique Unruh. Towards compressed permutation oracles. In *Advances in Cryptology – ASIACRYPT 2023*, volume 14441 of *Lecture Notes in Computer Science*, pages 369–400. Springer, 2023.

[Wat18] John Watrous. *The Theory of Quantum Information.* Cambridge University Press, 2018.

[Wil19] Mark M Wilde. *Quantum Information Theory.* Cambridge University Press, second edition, 2019.

[Zha19] Mark Zhandry. How to record quantum queries, and applications to quantum indifferentiability. In *Advances in Cryptology–CRYPTO 2019*, pages 239–268. Springer, 2019.