Symmetry-assisted adversaries for quantum state generation

Andris Ambainis
University of Latvia
andris.ambainis@lu.lv

Loïck Magnin
Université Libre de Bruxelles
Université Paris-Sud
NEC Laboratories America
lmagnin@lri.fr

Martin Roetteler and Jérémie Roland
NEC Laboratories America
{mroetteler, jroland}@nec-labs.com

***Abstract*—We introduce a new quantum adversary method to prove lower bounds on the query complexity of the quantum state generation problem. This problem encompasses both, the computation of partial or total functions and the preparation of target quantum states. There has been hope for quite some time that quantum state generation might be a route to tackle the GRAPH ISOMORPHISM problem. We show that for the related problem of INDEX ERASURE our method leads to a lower bound of square root of *N* which matches an upper bound obtained via reduction to quantum search on *N* elements. This closes an open problem first raised by Shi [FOCS'02].**

Our approach is based on two ideas: (i) on the one hand we generalize the known additive and multiplicative adversary methods to the case of quantum state generation, (ii) on the other hand we show how the symmetries of the underlying problem can be leveraged for the design of optimal adversary matrices and dramatically simplify the computation of adversary bounds. Taken together, these two ideas give the new result for INDEX ERASURE by using the representation theory of the symmetric group. Also, the method can lead to lower bounds even for small success probability, contrary to the standard adversary method. Furthermore, we answer an open question due to Špalek [CCC'08] by showing that the multiplicative version of the adversary method is stronger than the additive one for any problem. Finally, we prove that the multiplicative bound satisfies a strong direct product theorem, extending a result by Špalek to quantum state generation problems.

***Keywords*-quantum query complexity; adversary method; strong direct product theorem; index-erasure**

INTRODUCTION

The query model provides a way to analyze quantum algorithms including, but not limited to, those of Shor [1] and Grover [2] as well as quantum walks, quantum counting, and hidden subgroup problems. Traditionally, in this model the input is a black-box function which can be accessed via queries and the output is a *classical* value. The measure of complexity of an algorithm is then defined as the number of queries made by the algorithm. Studying the quantum query complexity of functions is quite fruitful since the model is simple enough that one can show tight bounds for several problems and hence provides some intuition about the power of quantum computing.

In this paper, we study a generalization of the query model to include problems in which the input is still a black-box function, however, the output is no longer a classical value but a target *quantum* state. An example for the resulting quantum state generation problem is INDEX ERASURE. Here we are given access to an injective function $f : [N] \to [M]$ and the task is to prepare the quantum state $\frac{1}{\sqrt{N}} \sum_{x=1}^{N} |f(x)\rangle$ using as few queries to f as possible. The name "index erasure" stems from the observation that while it is straightforward to prepare the (at first glance perhaps similar looking) state $\frac{1}{\sqrt{N}} \sum_{x=1}^{N} |x\rangle|f(x)\rangle$, it is quite challenging to forget ("erase") the contents of the first register of this state which carries the input ("index") of the function.

In particular, this approach has been considered in [3] to solve statistical zero knowledge problems, one ultimate goal being to tackle GRAPH ISOMORPHISM [4]. The quantum state generation problem resulting from the well-known reduction of GRAPH ISOMORPHISM to INDEX ERASURE would be to generate the uniform superposition of all the permutations of a graph Γ:

$$|\Gamma\rangle = \frac{1}{\sqrt{n!}} \sum_{\pi \in S_n} |\Gamma^\pi\rangle.$$

By coherently generating this state for two given graphs, one could then use the standard SWAP-test to check whether the two states are equal or orthogonal, and therefore decide whether the graphs are isomorphic or not. Such a method for solving GRAPH ISOMORPHISM would be drastically different from more standard approaches based on the reduction to the hidden subgroup problem, and might therefore provide a way around serious limitations of the coset state approach [5]. There has been hope for quite some time that quantum state generation might be a route to tackle the GRAPH ISOMORPHISM problem, however one of the main results of this paper is that any approach that tries to generate $|\Gamma\rangle$ without exploiting further structure[1] of the graph cannot improve on the simple $O(\sqrt{n!})$ upper bound via search. More generally, we are interested in the query complexity of the quantum state generation problem, in which the amplitudes of the target quantum state can depend

[1] Indeed, here we assume that the only way to access the graph Γ would be by querying an oracle that, given a permutation π, returns the permuted graph Γ^π.

on the given function in an arbitrary way. Subroutines for quantum state generation might provide a useful toolbox to design efficient quantum algorithms for a large class of problems.

Adversaries: Lower bounds on the quantum query complexity have been shown for a wide range of (classical in the above sense) functions. Roughly speaking, currently there are two main ideas for proving lower bounds on quantum query complexity: the polynomial method [6], [7], [8], [9], [10], [11] and the adversary method [12]. The latter method has seen a sequence of variations, generalizations, and improvements over the past decade including [13], [10], [14], [15].

The basic idea behind the adversary method and its variations is to define a progress function that monotonically changes from an initial value (before any query) to a final value (depending on the success probability of the algorithm) with one main property: the value of the progress function changes only when the oracle is queried. Then, a lower bound on the quantum query complexity of the problem can be obtained by bounding the amount of progress done by one query.

Different adversary methods were introduced, but they were later proved to be all equivalent [16]. They rely on optimizing an adversary matrix assigning weights to different pairs of inputs to the problem. While originally these methods only considered positive weights, it was later shown that negative weights also lead to a lower bound, which can actually be stronger in some cases [17]. The relevance of this new adversary method with negative weights, called *additive*, was made even clearer when it was very recently shown to be tight for the quantum query complexity of functions in the bounded-error model [18], [19].

Nevertheless, for some problems other methods (such as the polynomial method or other ad-hoc techniques) might be easier to implement while also leading to strong bounds. The additive adversary method also suffers from one main drawback: it cannot prove lower bounds for very small success probability. To circumvent it, Špalek introduced the *multiplicative* adversary method [20] that generalizes some previous ad-hoc methods [21], [22]. Being able to deal with exponentially small success probability also allowed to prove a strong direct product theorem for any function that admits a multiplicative adversary lower bound [21], [22], [20] (note that a similar result has recently been proved for the polynomial method [23]). Roughly speaking, it means that if we try to compute k independent instances of a function using less than $O(k)$ times the number of queries required to compute one instance, then the overall success probability is exponentially small in k. However, Špalek left unanswered the question of how multiplicative and additive methods relate in the case of high success probability. In particular, it is unknown whether the strong direct product theorem extends to the additive adversary method, and therefore to the quantum query complexity of any function since this method is known to be tight in the bounded error model [19]. The quantum query complexity of functions nevertheless satisfies a weaker property called direct sum theorem, meaning that computing k instances requires at least $\Omega(k)$ times the number of queries necessary to solve one instance, but it is unknown how the success probability decreases if less than $O(k)$ queries are used.

Related work: We are not aware of techniques to directly prove lower bounds for quantum state generation problems, and the only lower bounds are based on reductions to computing functions. One particular example is a lower bound for the already mentioned INDEX ERASURE problem, which consists in generating the uniform superposition over the image of an injective function. The best lower bound comes from a $\Omega(\sqrt[5]{N/\log N})$ lower bound for the SET EQUALITY problem [24], which consists in deciding whether two sets of size N are equal or disjoint or, equivalently, whether two injective functions over a domain of size N have equal or disjoint images. This problem reduces to INDEX ERASURE since by generating the superposition over the image of the two functions, we can decide whether they are equal or not using the SWAP-test. Therefore, this implies the same $\Omega(\sqrt[5]{N/\log N})$ lower bound for INDEX ERASURE. However, this lower bound is probably not tight, neither for SET EQUALITY, whose best upper bound is $O(\sqrt[3]{N})$ due to the algorithm for COLLISION [25], nor for INDEX ERASURE, whose best upper bound is $O(\sqrt{N})$ due to an application of Grover's algorithm fro SEARCH [2]. The question of the complexity of INDEX ERASURE has first been raised by Shi [9] in 2002 and has remained open until the present work.

Our results: The chief technical innovation of this paper is an extension of both, the additive and multiplicative adversary methods, to quantum state generation **(Theorems 5 and 9)**. To do so, we give a geometric interpretation of the adversary methods which is reminiscent of the approach of [21], [20], where this is done for classical problems. As a by-product we give elementary and arguably more intuitive proofs of the additive and multiplicative methods, contrasting with some rather technical proofs *e.g.* in [17], [20].

In order to compare the additive and multiplicative adversary bounds, we introduce yet another flavor of adversary method **(Theorem 7)**, which we will call *hybrid* adversary method. Indeed, this method is a hybridization of the additive and multiplicative methods that uses "multiplicative" arguments in an "additive" setup: it is equivalent to the additive method for large success probability, but is also able to prove non-trivial lower-bounds for small success probability, overcoming the concern [20] that the additive adversary method might fail in this case. We show that for any problem, the hybrid adversary bound lies between the additive and multiplicative adversary bounds **(Theorem 11)**, answering Špalek's open question about the relative power of these methods [20]. By considering the SEARCH problem for exponentially small success probability, we also conclude that the powers of the three methods are *strictly* increasing, since the corresponding lower bounds scale differently as a function of the success probability in that regime **(Theorem 19)**.

We then extend the strong direct product theorem for the multiplicative adversary bound [20] to quantum state generation problems **(Theorem 12)**. Since we have clarified the relation between the additive and multiplicative adversary methods, this also brings us closer to a similar theorem for the additive adversary method. The most important consequence would be for the quantum query complexity of functions, which would therefore also satisfy a strong direct product theorem since the additive adversary bound is tight in this case [19]. However, it remains to prove some technical lemma about the multiplicative bound to be able to conclude.

As it has been previously pointed out many interesting problems have strong symmetries [21], [22], [20]. We show how studying these symmetries helps to address the two main difficulties of the usage the adversary method, namely, how to choose a good adversary matrix Γ and how to compute the spectral norm of $\Gamma_x - \Gamma$ **(Theorem 17)**. Following the *automorphism principle* of [17], we define the automorphism group G of $\mathcal{P}$, and its restrictions G_x, for any input x to the oracle. We show how computing the norm of $\Gamma_x - \Gamma$ can be simplified to compute the norm of much smaller matrices that depend only on the irreps of G and G_x. For problems with strong symmetries, these matrices typically have size at most 3×3 [21], [22], [20]. We have therefore reduced the adversary method from an algebraic problem to the study of the representations of the automorphism group.

Finally, we use our hybrid adversary method to prove a lower bound of $\Omega(\sqrt{N})$ for the quantum query complexity of INDEX ERASURE **(Theorem 20)**, which is tight due to the matching upper bound based on Grover's algorithm, therefore closing the open problem stated by Shi [9]. To the best of our knowledge, this is the first lower bound directly proved for the query complexity of a quantum state generation problem. The lower bound is entirely based on the study of the representations of the symmetric group, a technique that might be fruitful for other problems having similar symmetries, such as the SET EQUALITY problem [24], or in turn some stronger quantum state generation approaches to GRAPH ISOMORPHISM.

I. ADVERSARY METHODS: GENERAL CONCEPTS

A. *Definition of the problem*

In this section, we describe elements which are common to all adversary methods. The goal of these methods is to study the quantum query complexity of some problems in the bounded-error model when we have access to an oracle O_f computing a function $f : \Sigma_I \mapsto \Sigma_O$. In this article, we will consider an oracle acting on two registers, the input register $\mathcal{I}$ and the output register $\mathcal{O}$, as:

$$|x\rangle_{\mathcal{I}} |s\rangle_{\mathcal{O}} \xrightarrow{O_f} |x\rangle_{\mathcal{I}} |s \oplus f(x)\rangle_{\mathcal{O}},$$

where $x \in \Sigma_I$ and $s, f(x) \in \Sigma_O$. Note that it is also possible to consider other types of oracle, for example computing the value of the function into the phase instead of into another register, but these different models all lead to equivalent notions of query complexity (up to a constant).

We denote by F the set of all possible functions f that can be encoded into the oracle. We will consider three types of problems $\mathcal{P}$, a classical one and two quantum ones:

- **Function:** Given an oracle O_f, compute the classical output $\mathcal{P}(f)$. The success probability of an algorithm A solving $\mathcal{P}$ is $\min_{f \in F} \Pr[A(f) = \mathcal{P}(f)]$, where $A(f)$ is the classical output of the algorithm on oracle f.
- **Coherent quantum state generation:** Given an oracle O_f, generate a quantum state $|\mathcal{P}(f)\rangle = |\psi_f\rangle$ in some target register $\mathcal{T}$, and reset all other registers to a default state $|\bar{0}\rangle$. Let $|\psi_f^T\rangle = \sqrt{1-\varepsilon_f}|\psi_f\rangle|\bar{0}\rangle + \sqrt{\varepsilon_f}|\mathrm{err}_f\rangle$ be the final state of an algorithm A on oracle f, where $|\mathrm{err}_f\rangle \perp |\psi_f\rangle|\bar{0}\rangle$ is some error state. Then, the success probability of A is given by $\min_{f \in F}(1 - \varepsilon_f)$.
- **Non-coherent quantum state generation:** Given an oracle O_f, generate a quantum state $|\mathcal{P}(f)\rangle = |\psi_f\rangle$ in some target register $\mathcal{T}$, while some f-dependent junk state may be generated in other registers. The success probability of an algorithm A solving $\mathcal{P}$ is given by $\min_{f \in F} \left\| \Pi_{|\psi_f\rangle} |\psi_f^T\rangle \right\|^2$, where $|\psi_f^T\rangle$ is the final state of the algorithm and $\Pi_{|\psi_f\rangle}$ is the projector on $|\psi_f\rangle$.

Let us note that computing a function is a special case of non-coherent quantum state generation, where all states $|\mathcal{P}(f)\rangle$ are computational basis states. Indeed, no coherence is needed since the state is in this case measured right after its generation. However, when the quantum state generation is used as a subroutine in a quantum algorithm for another problem, coherence is typically needed to allow interferences between different states. This is in particular the case for solving SET EQUALITY via reduction to INDEX ERASURE, and similarly to solve GRAPH ISOMORPHISM via the quantum state generation approach, since coherence is required to implement the SWAP-test.

Without loss of generality we can consider the algorithm as being a circuit $\mathcal{C}$ consisting of a sequence of unitaries $U_0, \ldots, U_T$ and oracle calls O_f acting on the "algorithm" Hilbert space $\mathcal{A}$. $\mathcal{A}$ can be decomposed into three registers: the input register $\mathcal{I}$; the output register $\mathcal{O}$ for the oracle; and an additional workspace register $\mathcal{W}$.

At the end of the circuit, a target register $\mathcal{T}$ holds the output of the algorithm. In the classical case, this register is measured to obtain the classical output $A(f)$. In the quantum case, it holds the output state $A(f)$.

In both cases, for a fixed algorithm, we note $|\psi_f^t\rangle$ the state of the algorithm after the t-th query. The idea behind the adversary methods is to consider that f is in fact an input to the oracle. We therefore introduce a function register $\mathcal{F}$ holding this input, and define a *super-oracle* O acting on registers $\mathcal{I} \otimes \mathcal{O} \otimes \mathcal{F}$ as

$$|x\rangle_{\mathcal{I}}|s\rangle_{\mathcal{O}}|f\rangle_{\mathcal{F}} \xrightarrow{O} |x\rangle_{\mathcal{I}}|s \oplus f(x)\rangle_{\mathcal{O}}|f\rangle_{\mathcal{F}}. \qquad (1)$$

We see that when the function register $\mathcal{F}$ is in state $|f\rangle$, O acts on $\mathcal{I} \otimes \mathcal{O}$ just as O_f. Suppose, just for the sake of analyzing the algorithm, that we prepare register $\mathcal{F}$ in the state $|\delta\rangle = \frac{1}{\sqrt{|F|}} \sum_{f \in F} |f\rangle$, the uniform superposition over all the elements of F, and that we apply the same circuit as before, by replacing each call to O_f by a call to O. Intuitively, each oracle call introduces more entanglement between this new register and the algorithm register. The state of this new circuit after the t-th query is $|\Psi^t\rangle = \frac{1}{\sqrt{|F|}} \sum_{f \in F} |\psi_f^t\rangle_{\mathcal{A}}|f\rangle_{\mathcal{F}}$.

Note that only oracle calls can modify the state of the function register $\mathcal{F}$, since all other gates only affect the algorithm register $\mathcal{A} = \mathcal{I} \otimes \mathcal{O} \otimes \mathcal{W}$. The general idea of all adversary methods is to study the evolution of the algorithm by looking at the reduced state of the input register, $\rho^t = \mathrm{tr}_{\mathcal{A}}|\Psi^t\rangle\langle\Psi^t| = \frac{1}{|F|}\sum_{f,f' \in F}\langle\psi_{f'}^t|\psi_f^t\rangle|f\rangle\langle f'|$. The algorithm starts with the state $\rho^0 = |\delta\rangle\langle\delta|$ and ends in a state ρ^T.

B. Adversary matrices and progress function

The adversary method studies how fast ρ^t can change from ρ^0 to ρ^T. We introduce a progress function in order to do so.

Definition 1 (Adversary matrix). *An adversary matrix Γ is a Hermitian matrix such that $\Gamma|\delta\rangle = |\delta\rangle$. An additive adversary matrix also satisfies $-\mathbb{I} \preceq \Gamma \preceq \mathbb{I}$ (i.e., $\|\Gamma\| = 1$), while a multiplicative adversary matrix satisfies $\Gamma \succeq \mathbb{I}$. In both cases, the progress function is defined as $W^t = \mathrm{tr}\,[\Gamma\rho^t]$.*

We will also use a matrix Γ_x derived from the adversary matrix Γ and defined as follows (for both the additive and the multiplicative case).

Definition 2 (Γ_x, D_x)**.** *For any adversary matrix Γ, let $\Gamma_x = \Gamma \circ D_x$, where $\circ$ denotes the Hadamard (element-wise) product and D_x is the (0-1)-matrix $D_x = \sum_{f,f'} \delta_{f(x),f'(x)}|f'\rangle\langle f|$ where δ denotes the Kronecker's delta.*

We will show that the Hadamard product is closely related to oracle calls: when the input register is in the state $|x\rangle$, the oracle calls acts on the function register as the Hadamard product with D_x. It is easy to check that this Hadamard product is a CP-map.

Fact 3. *The map $\gamma \mapsto \gamma \circ D_x$ is a CP-map and $\gamma \circ D_x = \sum_y \Pi_y^x \gamma \Pi_y^x$ with $\Pi_y^x = \sum_{f:f(x)=y} |f\rangle\langle f|$.*

The basic idea of all adversary methods is to bound how much the value of the progress function can change by one oracle call. To study the action of one oracle call, we isolate the registers $\mathcal{I}$ and $\mathcal{O}$ holding the input and output of the oracle from the rest of the algorithm register. Without loss of generality, we may assume that for any oracle call, the output register $\mathcal{O}$ is in the state $|0\rangle_{\mathcal{O}}$ (*computing* oracle call) or $|f(x)\rangle_{\mathcal{O}}$ (*uncomputing* oracle call). Indeed, an oracle call for any other state $|s\rangle_{\mathcal{O}}$ may be simulated by one computing oracle call, $O(\log|\Sigma_O|)$ XOR gates and one uncomputing oracle call. Therefore, this assumption only increases the query complexity by a factor at most 2.

II. The different adversary methods

A. Lower bounds

Definition 4 ($\rho^{\odot}$, junk matrix)**.** *For a quantum state generation problem $\mathcal{P}$ such that $|\mathcal{P}(f)\rangle = |\psi_f\rangle$, we denote by $\rho^{\odot}$ the* target *state $\rho^{\odot} = \frac{1}{|F|}\sum_{f,f' \in F}\langle\psi_f|\psi_{f'}\rangle|f'\rangle\langle f|$. In the non-coherent case, we call junk matrix any Gram matrix M of size $|F| \times |F|$ such that $M_{ij} = \langle v_i|v_j\rangle$, where $\{v_i : i \in [|F|]\}$ is a set of unit vectors (or, equivalently, any semi-definite matrix M such that $M_{ii} = 1$ for any $i \in [|F|]$). In the coherent case, we call junk matrix the all-1 matrix of size $|F| \times |F|$.*

Theorem 5 (Additive adversary method [17])**.** *Consider a quantum algorithm solving $\mathcal{P}$ with success probability at least $1 - \varepsilon$, and let $\tilde{\Gamma}$ be an additive adversary matrix such that $\mathrm{tr}\left[\tilde{\Gamma}(\rho^{\odot} \circ M)\right] = 0$ for any junk matrix M. Then, $Q_\varepsilon(\mathcal{P}) \geq \frac{1-C(\varepsilon)}{\max_x\left\|\tilde{\Gamma}_x - \tilde{\Gamma}\right\|}$ where $C(\varepsilon) = \varepsilon + 2\sqrt{\varepsilon(1-\varepsilon)}$*

For classical problems, we now prove that our method generalizes [17]. Indeed, our condition on the adversary matrix is different, which allows us to also deal with quantum problems. However, for classical problems, the following lemma shows that the usual condition implies our modified condition. Let $\mathcal{P}(f)$ be the function to be computed.

Lemma 6. $\mathrm{tr}\left[\tilde{\Gamma}(\rho^{\odot} \circ M)\right] = 0$ *for any matrix M if and only if $\tilde{\Gamma}_{ff'} = 0$ for any f, f' such that $\mathcal{P}(f) = \mathcal{P}(f')$.*

The original adversary method can only prove a lower bound when $C(\varepsilon) < 1$, that is, when the success probability $1 - \varepsilon > \frac{4}{5}$. For smaller success probability, we need to prove a stronger bound on the final value of the progress function $\tilde{W}^T$. Inspired by the multiplicative adversary method [20], we prove the following *hybrid* adversary bound.

Theorem 7 (Hybrid adversary method)**.** *Consider a quantum algorithm solving $\mathcal{P}$ with success at least $1 - \varepsilon$. Let $\tilde{\Gamma}$ be any additive adversary matrix, V_{bad} be the direct sum of eigenspaces of $\tilde{\Gamma}$ with eigenvalue strictly larger than*

$\tilde{\lambda} < 1$, *and assume that* $\operatorname{tr}[\Pi_{\mathrm{bad}}(\rho^{\odot} \circ M)] \leq \eta$ *for any junk matrix* M*, where* Π_{bad} *is the projector on* V_{bad}*, and* $0 \leq \eta \leq 1 - \varepsilon$*. We have* $Q_\varepsilon(\mathcal{P}) \geq \frac{\tilde{K}(\tilde{\Gamma},\tilde{\lambda},\varepsilon)}{\max_x \left\|\tilde{\Gamma}_x - \tilde{\Gamma}\right\|}$ *where* $\tilde{K}(\tilde{\Gamma},\tilde{\lambda},\varepsilon) = (1-\tilde{\lambda})(\sqrt{1-\varepsilon} - \sqrt{\eta})^2$.

For classical problems, we can use the following lemma:

Lemma 8. *Let* Π_{bad} *be the projector on* V_{bad}, $\Pi_z = \sum_{\mathcal{P}(f)=z} |f\rangle\langle f|$*, and assume that* $\left\|\Pi_z \Pi_{\mathrm{bad}}\right\|^2 \leq \eta$ *for any* z*. Then,* $\operatorname{tr}[\Pi_{\mathrm{bad}}(\rho^{\odot} \circ M)] \leq \eta$ *for any junk matrix* M*.*

Theorem 9 (Multiplicative adversary method [20])**.** *Consider a quantum algorithm solving* $\mathcal{P}$ *with success at least* $1-\varepsilon$*. Let* Γ *be any multiplicative adversary matrix,* V_{bad} *be the direct sum of eigenspaces of* Γ *with eigenvalue strictly smaller than* $\lambda > 1$*, and assume that* $\operatorname{tr}[\Pi_{\mathrm{bad}}(\rho^{\odot} \circ M)] \leq \eta$ *for any junk matrix* M*, where* Π_{bad} *is the projector on* V_{bad}*, and* $0 \leq \eta \leq 1 - \varepsilon$*. We have*

$$Q_\varepsilon(\mathcal{P}) \geq \frac{\log K(\Gamma,\lambda,\varepsilon)}{\log \max\left\{\left\|\Gamma_x^{1/2}\Gamma^{-1/2}\right\|^2, \left\|\Gamma^{1/2}\Gamma_x^{-1/2}\right\|^2 : \forall x \in \mathcal{I}\right\}},$$

where $K(\Gamma,\lambda,\varepsilon) = 1 + (\lambda - 1)(\sqrt{1-\varepsilon} - \sqrt{\eta})^2$.

Note that since the condition on the adversary matrix is very similar as for the hybrid adversary, we can also use an analogue of Lemma 8 to choose the adversary matrix in the special case of classical problems. This implies that our method is an extension of Špalek's original multiplicative adversary method [20].

B. Comparison of the adversary methods

We show that the three methods are progressively stronger.

Definition 10. *We define the* additive adversary bound *and the* hybrid adversary bound *respectively as*

$$\mathrm{ADV}^{\pm}_\varepsilon(\mathcal{P}) = \max_{\tilde{\Gamma}} \frac{1 - C(\varepsilon)}{\max_x \left\|\tilde{\Gamma} - \tilde{\Gamma}_x\right\|},$$

$$\widetilde{\mathrm{ADV}}_\varepsilon(\mathcal{P}) = \max_{\tilde{\Gamma},\tilde{\lambda}<1} \frac{\tilde{K}(\tilde{\Gamma},\tilde{\lambda},\varepsilon)}{\max_x \left\|\tilde{\Gamma} - \tilde{\Gamma}_x\right\|}$$

where, for $\mathrm{ADV}^{\pm}$*, the maximum is taken over additive adversary matrices* $\tilde{\Gamma}$ *such that* $\operatorname{tr}\left[\tilde{\Gamma}(\rho^{\odot} \circ M)\right] = 0$ *for any junk matrix* M*, while for* $\widetilde{\mathrm{ADV}}$ *it is taken over all additive adversary matrices. Finally, we define the* multiplicative adversary bound *as* $\mathrm{MADV}_\varepsilon(\mathcal{P}) = \sup_{\lambda>1} \mathrm{MADV}^{(\lambda)}_\varepsilon(\mathcal{P})$ *where*

$$\mathrm{MADV}^{(\lambda)}_\varepsilon(\mathcal{P}) = \sup_{\Gamma} \frac{\log K(\Gamma,\lambda,\varepsilon)}{\log \max\left\{\left\|\Gamma_x^{1/2}\Gamma^{-1/2}\right\|^2, \left\|\Gamma^{1/2}\Gamma_x^{-1/2}\right\|^2 : \forall x \in \mathcal{I}\right\}},$$

and the supremum is taken over all multiplicative adversary matrices Γ*.*

Theorem 11. $\mathrm{MADV}_\varepsilon(\mathcal{P}) \geq \widetilde{\mathrm{ADV}}_\varepsilon(\mathcal{P}) \geq \mathrm{ADV}^{\pm}_\varepsilon(\mathcal{P})/60$.

Proof: The second inequality is obtained by a straightforward comparison of the bound on the final value of the progress function. The first inequality relies on $\lim_{\lambda\to 1} \mathrm{MADV}^{(\lambda)}_\varepsilon(\mathcal{P}) \geq \widetilde{\mathrm{ADV}}_\varepsilon(\mathcal{P})$. This is proven by using a multiplicative adversary matrix $\Gamma(\gamma) = \mathbb{I} + \gamma(\mathbb{I} - \tilde{\Gamma})$ with threshold $\lambda(\gamma) = 1 + \gamma(1 - \tilde{\lambda})$, where $\tilde{\Gamma}$ and $\tilde{\lambda}$ are the optimal adversary matrix and threshold for the hybrid method, and taking the limit $\gamma \to 0$. ■

C. Strong direct product theorem

We can show that Špalek's strong direct product theorem also holds for quantum state generation problems.

Theorem 12 (Strong direct product)**.** *For any* $\varepsilon > 0$ *and* $\lambda > 1$ *there exist a constant* $0 < c < 1$ *and an integer* k_0 *such that for any problem* $\mathcal{P}$ *and* $k > k_0$*:*

$$\mathrm{MADV}^{(\lambda)}_{1-c^k}(\mathcal{P}^{(k)}) \geq \frac{k}{10} \cdot \mathrm{MADV}^{(\lambda)}_\varepsilon(\mathcal{P}).$$

Let us note that while we have proved that the multiplicative adversary method is stronger than the additive one, we cannot directly conclude that this strong direct product theorem also applies to the additive bound. This is because we can only prove that the multiplicative adversary method becomes stronger in the limit of λ going to 1, while in the same limit the constant c in the theorem also goes to 1. Therefore, this only implies a direct sum theorem for the additive adversary bound. To conclude that it also satisfies a direct product theorem, one approach would be to prove that whenever the multiplicative adversary method can prove a lower bound in the limit $\lambda \to 1$, there exists some fixed $\lambda > 1$ which leads to the same bound. The most important consequence would be for the quantum query complexity of functions, which would itself satisfy a strong direct product theorem for any function, since the additive adversary method is known to be tight in that case [18], [19].

III. Representation theory

In this section we will study how the symmetries of the problem can help choosing the adversary matrix and in turn obtain the lower bounds. Recall that the oracle computes a function $f \in F$ from Σ_I to Σ_O, where the input alphabet has size $N = |\Sigma_I|$ and the output alphabet has size $M = |\Sigma_O|$. Let us consider permutations $(\pi,\tau) \in S_N \times S_M$ acting on $f \in F$ as $f_{\pi,\tau} = \tau \circ f \circ \pi$, that is, $f_{\pi,\tau} : \Sigma_I \mapsto \Sigma_O : x \mapsto \tau(f(\pi(x)))$.

Definition 13 (Automorphism group of $\mathcal{P}$)**.** *We call a group* $G \subseteq S_N \times S_M$ *an* automorphism group *of a problem* $\mathcal{P}$ *if*

- *For any* $(\pi,\tau) \in G$ *and* $f \in F$*, we have* $f_{\pi,\tau} \in F$*.*
- *For any* $(\pi,\tau) \in G$*, there exists a unitary* $V_{\pi,\tau}$ *such that* $V_{\pi,\tau}|\mathcal{P}(f)\rangle\rangle = |\mathcal{P}(f_{\pi,\tau})\rangle\rangle$ *for all* $f \in F$*.*

Note that from an oracle for f, it is easy to simulate an oracle for $f_{\pi,\tau}$ by prefixing and appending the necessary permutations on the input and output registers. Consider for example a computing oracle call. Then, $O_{f_{\pi,\tau}}$ acts on $|x\rangle|0\rangle$ just as $(\pi^{-1}\otimes\tau)O_f(\pi\otimes\mathbb{I})$.

Therefore, if (π,τ) is an element of an automorphism G of $\mathcal{P}$, we can solve the problem with oracle f in the following indirect way:

1) Solve the problem for $f_{\pi,\tau}$, which will prepare a state close to $|\mathcal{P}(f_{\pi,\tau})\rangle$.
2) Apply $V^\dagger_{\pi,\tau}$ to map this state to a state close to $|\mathcal{P}(f)\rangle$.

Since we want the algorithm to work just as well for any possible f, we can use this property to symmetrize the circuit. The idea is to solve the algorithm for f by solving it for $f_{\pi,\tau}$ for all possible $(\pi,\tau)\in G$ simultaneously in superposition. Just as we considered $|f\rangle$ as an additional input to the circuit, we can also use the same mathematical trick and consider $|\pi,\tau\rangle$ as another input. We then run the algorithm on the superposition $\frac{1}{\sqrt{|G|}}\sum_{(\pi,\tau)\in G}|\pi,\tau\rangle$. Note that we can assume without loss of generality that the best algorithm for $\mathcal{P}$ is symmetrized. Indeed, for any algorithm for $\mathcal{P}$ with success probability p and query complexity T, the symmetrized version will have the same query complexity and a success probability at least p. For the same reason, we can also assume that the optimal adversary matrix satisfies a similar symmetry, in the following sense:

Lemma 14. *For all $(\pi,\tau)\in G$, let $U_{\pi,\tau}$ be the unitary that maps $|f\rangle$ onto $|f_{\pi,\tau}\rangle$. Then, we can assume without loss of generality that the optimal adversary matrix Γ satisfies $U_{\pi,\tau}\Gamma U^\dagger_{\pi,\tau}=\Gamma$ for any $(\pi,\tau)\in G$.*

Note that the mapping $\mathcal{U}:(\pi,\tau)\mapsto U_{\pi,\tau}$ defines a representation of the automorphism group G and that Lemma 14 implies that Γ commutes with $U_{\pi,\tau}$ for any $(\pi,\tau)\in G$. This means that the matrices $U_{\pi,\tau}$ and Γ block-diagonalize simultaneously in a common basis, where each block corresponds to a different irrep of G in $\mathcal{U}$. From now on, we will consider the special case where $\mathcal{U}$ is multiplicity-free. This happens for different interesting problems, such as t-FOLD SEARCH [22], [20] and INDEX ERASURE (see Section IV-B), as a consequence of the following lemma.

Lemma 15. *If, for any $f,g\in F$, there exists $(\pi,\tau)\in G$ such that $g=f_{\pi,\tau}$ and $g_{\pi,\tau}=f$, then $\mathcal{U}$ is multiplicity-free.*

Proof: Let us consider the set of matrices $\mathcal{M}=\{A\in\mathbb{C}^{|F|\times|F|}:\forall(\pi,\tau)\in G,\ U_{\pi,\tau}AU^\dagger_{\pi,\tau}=A\}$. It is easy to see that for any $A,B\in\mathcal{M}$, we have $AB\in\mathcal{M}$, therefore $\mathcal{M}$ defines an algebra. Note that $\mathcal{U}$ is multiplicity-free if and only if $\mathcal{M}$ is commutative, in which case all matrices in $\mathcal{M}$ diagonalize in a common basis [26, p. 65]. For any matrix $A\in\mathcal{M}$, we have $A^t=A$ since there exists $(\pi,\tau)\in G$ such $\langle f|A|g\rangle=\langle f|U_{\pi,\tau}AU^\dagger_{\pi,\tau}|g\rangle=\langle g|A|f\rangle$. This immediately implies that for any $A,B\in\mathcal{M}$, we have $AB=(AB)^t=B^tA^t=BA$, therefore $\mathcal{M}$ is a commutative algebra. (More precisely, it is a Bose-Mesner algebra associated to an association scheme [27]) ■

Recall that oracle calls are closely related to the Hadamard product with D_x. We show that the invariance of Γ under the action of a group G implies the invariance of $\Gamma_x=\Gamma\circ D_x$ under the action of the subgroup G_x of G that leaves x invariant.

Lemma 16. *For any $x\in\Sigma_I$ and $y\in\Sigma_O$, let us define the following subgroups of G*

$$G_{xy}=\{(\pi,\tau)\in G:\pi(x)=x,\tau(y)=y\},$$
$$G_x=\{(\pi,\tau)\in G:\pi(x)=x\}.$$

Then Π^x_y satisfies $U_{\pi,\tau}\Pi^x_yU^\dagger_{\pi,\tau}=\Pi^x_y$ for any $(\pi,\tau)\in G_{xy}$, and Γ_x satisfies $U_{\pi,\tau}\Gamma_xU^\dagger_{\pi,\tau}=\Gamma_x$ for any $(\pi,\tau)\in G_x$.

Proof: Recall that by definition of Π^x_y, we have $U_{\pi,\tau}\Pi^x_yU^\dagger_{\pi,\tau}=\Pi^{\pi^{-1}(x)}_{\tau(y)}$ for any $(\pi,\tau)\in G$. This immediately implies the first part of the lemma for $(\pi,\tau)\in G_{xy}$. Moreover, Fact 3 and Lemma 14 imply that $U_{\pi,\tau}\Gamma_xU^\dagger_{\pi,\tau}=\Gamma_{\pi^{-1}(x)}$ for any $(\pi,\tau)\in G$. This implies the second part of the lemma for $(\pi,\tau)\in G_x$. ■

Since $\mathcal{U}$ is a representation of G, it is also a representation of the subgroup G_x. However, even if $\mathcal{U}$ is multiplicity-free with respect to G, it is typically not with respect to G_x. Indeed, when restricting G to G_x, multiplicities can happen due to two different mechanisms. First, an irrep can become reducible, and one of the new smaller irreps can be a copy of another irrep. Secondly, two irreps that are different for G could be the same when we restrict to the elements of G_x. Let us identify an irrep of G_x by three indices (k,l,m): the first index identifies the irrep k of G from which it originates, the second index identifies the irrep l of G_x, and the last index allows to discriminate betwen different copies of the same irrep of G_x. For example, two irreps having the same index l but different indices k are two copies of the same irrep of G_x originating from different irreps of G. Also, we denote by $V_{k,l,m}$ the subspace spanned by irrep (k,l,m). These subspaces are such that $\bigoplus_{l,m}V_{k,l,m}=V_k$, where V_k is the subspace spanned by the irrep k of G (we assume that $V_{k,l,m}$ is empty if (k,l,m) does not correspond to a valid irrep). In the following, it will also be useful to define $W_l=\bigoplus_{k,m}V_{k,l,m}$ which is sometimes called the isotypical component corresponding to l [28].

When $\mathcal{U}$ is multiplicity-free, Lemma 14 implies that Γ diagonalizes in the irrep basis. Then to choose the adversary matrix, it suffices to assign weights γ_k to each irrep k of G, i.e., $\Gamma=\sum_k\gamma_k\Pi_k$. Moreover, it also implies that computing the associated adversary bounds boils down to bounding for each irrep l of G_x the norm of a small $m_l\times m_l$ matrix, where m_l is the multiplicity of irrep l.

Theorem 17. *Let $\mathcal{U}$ be multiplicity-free for G. Then, we* *have*

$$\left\|\tilde{\Gamma}_x - \tilde{\Gamma}\right\| = \max_l \left\|\tilde{\Delta}_x^l\right\|, \qquad \left\|\Gamma_x^{1/2}\Gamma^{-1/2}\right\|^2 = \max_l \left\|\Delta_x^l\right\|,$$
$$\left\|\Gamma^{1/2}\Gamma_x^{-1/2}\right\|^2 = \max_l \left\|(\Delta_x^l)^{-1}\right\|,$$

where the maximums are over irreps l of G_x. For each irrep l, $\tilde{\Delta}_x^l$ and Δ_x^l are $m_l \times m_l$ matrices, where m_l is the multiplicity of l for G_x, with elements labeled by the different copies of the irrep and such that

$$\begin{aligned}(\tilde{\Delta}_x^l)_{k_1 m_1 k_2 m_2} =& \frac{1}{d_l}\sum_{k,y}\gamma_k \mathrm{tr}\left[\Pi_y^x \Pi_k \Pi_y^x \Pi^l_{k_1 m_1 \leftarrow k_2 m_2}\right] \\ &- \gamma_{k_1}\delta_{k_1 k_2}, \\ (\Delta_x^l)_{k_1 m_1 k_2 m_2} =& \frac{1}{d_l}\sum_{k,y}\frac{\gamma_k}{\sqrt{\gamma_{k_1}\gamma_{k_2}}}\mathrm{tr}\left[\Pi_y^x \Pi_k \Pi_y^x \Pi^l_{k_1 m_1 \leftarrow k_2 m_2}\right]\end{aligned}$$

where d_l is the dimension of irrep l and $\Pi^l_{k_1 m_1 \leftarrow k_2 m_2}$ is the "transporter" *from V_{k_2,l,m_2} to V_{k_1,l,m_1}, i.e., the operator that maps any state in V_{k_2,l,m_2} to the corresponding state in V_{k_1,l,m_1}.*

We see that to obtain the adversary bounds, we need to compute the traces of products of four operators. Since G_{xy} is a subgroup of both G and G_x, each of these operators can be decomposed into a sum of projectors onto irreps of G_{xy} (or transporters from and to these irreps). To compute these traces, we can use the following lemma, which shows that it is sufficient to compute the traces of products of two projectors onto irreps of G_{xy}.

Lemma 18. *Let $\lambda, \mu, \nu_1, \nu_2$ denote irreps of G_{xy}. If any of μ, ν_1 or ν_2 is not isomorphic to λ, then* $\mathrm{tr}\left[\Pi_\lambda \Pi_\mu \Pi_\lambda \Pi_{\nu_1 \leftarrow \nu_2}\right] = 0$. *Otherwise, we have*

$$\mathrm{tr}\left[\Pi_\lambda \Pi_\mu \Pi_\lambda \Pi_{\nu_1 \leftarrow \nu_2}\right] = \frac{1}{d}\mathrm{tr}\left[\Pi_\lambda \Pi_\mu\right] \cdot \mathrm{tr}\left[\Pi_\lambda \Pi_{\nu_1 \leftarrow \nu_2}\right],$$
$$\left|\mathrm{tr}\left[\Pi_\lambda \Pi_{\nu_1 \leftarrow \nu_2}\right]\right| = \sqrt{\mathrm{tr}\left[\Pi_\lambda \Pi_{\nu_1}\right] \cdot \mathrm{tr}\left[\Pi_\lambda \Pi_{\nu_2}\right]},$$

where d is the dimension of the representation λ.

IV. Applications

A. Search

By considering Grover's Search problem [2], which we denote Search$_n$, we can show that the inequalities in Theorem 11 are strict.

Theorem 19. *For any $0 < \varepsilon < 1 - \frac{1}{n}$, we have*

$$\mathrm{ADV}^{\pm}_\varepsilon(\textsc{Search}_n) = \Omega\left((1 - \varepsilon - 2\sqrt{\varepsilon(1-\varepsilon)})\sqrt{n}\right)$$
$$\widetilde{\mathrm{ADV}}_\varepsilon(\textsc{Search}_n) = \Omega\left((\sqrt{1-\varepsilon} - 1/\sqrt{n})^2\sqrt{n}\right)$$
$$\mathrm{MADV}_\varepsilon(\textsc{Search}_n) = \Omega\left((\sqrt{1-\varepsilon} - 1/\sqrt{n})\sqrt{n}\right).$$

In particular, for $\varepsilon > 1/5$, we have $\mathrm{MADV}_\varepsilon(\textsc{Search}_n) > \widetilde{\mathrm{ADV}}_\varepsilon(\textsc{Search}_n) > \mathrm{ADV}^{\pm}_\varepsilon(\textsc{Search}_n)$.

The $\Omega(\sqrt{n})$ lower bound for large success probability is well-known (see e.g [6]), and the case of small success probability has been studied in [21], [20] using the multiplicative adversary method. The fact that a non-trivial bound can also be found in this regime using an additive adversary method (our hybrid method) is new to the present work.

B. Index Erasure

Let us now consider the following coherent quantum state generation problem, called Index Erasure [9]: given an oracle for an injective function $f : [N] \to [M]$, coherently generate the superposition $|\psi_f\rangle = \frac{1}{\sqrt{N}}\sum_{x=1}^N |f(x)\rangle$ over the image of f.

The previously best known lower bound for Index Erasure is $\Omega(\sqrt[5]{N/\log N})$, which follows from a reduction to the Set Equality problem [24]. It is also known that this problem may be solved with $O(\sqrt{N})$ oracle calls. Indeed, given $|f(x)\rangle$, one can find the index $|x\rangle$ with $O(\sqrt{N})$ oracle calls using Grover's algorithm for Search [2]. Therefore, the quantum circuit for this algorithm maps the superposition $|\psi_f\rangle = \frac{1}{\sqrt{N}}\sum_{x=1}^N |f(x)\rangle$ to the the state $\frac{1}{\sqrt{N}}\sum_{x=1}^N |x\rangle|f(x)\rangle$. The algorithm for Index Erasure then follows by inverting this circuit. We now show that this algorithm is optimal by proving a matching lower bound using the hybrid adversary method.

Theorem 20. *For any $0 < \varepsilon < 1 - \frac{N}{M}$, we have* $Q_\varepsilon(\textsc{Index Erasure}) = \Theta(\sqrt{N})$.

Proof: Let $(\pi, \tau) \in S_N \times S_M$ act on the set F of injective functions from $[N]$ to $[M]$ by mapping f to $f_{\pi,\tau} = \tau \circ f \circ \pi$. Since we can obtain the state $|\psi_f\rangle$ from $|\psi_{f_{\pi,\tau}}\rangle$ by applying the permutation τ^{-1} on the target register, the whole group $G = S_N \times S_M$ defines an automorphism group for the problem.

Let us study the representation $\mathcal{U}$ corresponding to the action of G on the set of injective functions F. From Lemma 15, this representation is multiplicity-free: indeed, for any $f, g \in F$, it is easy to construct a group element (π, τ) that maps both f to g and g to f. Therefore, any irrep of G appears in $\mathcal{U}$ at most once. Let us now show that many irreps do not appear at all. Recall that irreps of $G = S_N \times S_M$ can be represented by pairs of Young diagrams (λ_N, λ_M), where λ_N has N boxes, and λ_M has M boxes [29]. We show that only irreps where the diagram λ_N is contained in the diagram λ_M can appear. We show this by induction on M, starting from $M = N$. For the base case, the set of injective functions F is isomorphic to the set of permutations in S_N, and $(\pi, \tau) \in S_N \times S_N$ acts on a permutation σ as $\tau\sigma\pi$. Therefore, the only irreps which occur in $\mathcal{U}$ are those where the two diagrams are the same, that is, $\lambda_N = \lambda_M$. When extending the range of functions in F from M to $M + 1$, we induce irreps of $S_N \times S_M$ to irreps of $S_N \times S_{M+1}$ by adding an extra box on the diagram corresponding to S_M. Since we start from a case where the two diagrams are the same, we can only obtain pairs of diagrams (λ_N, λ_M) where λ_N is contained inside λ_M.

The initial state is $\rho_0 = |\delta\rangle\langle\delta|$, where $|\delta\rangle = \frac{1}{\sqrt{|F|}} \sum_{f \in F} |f\rangle$ is the superposition over all injective functions, which is invariant under any element $(\pi, \tau) \in G$. Therefore, it corresponds to the trivial one-dimensional irrep of $S_N \times S_M$, represented by a pair of diagrams (λ_N, λ_M) where both diagrams contain only one row of N and M boxes, respectively. Let $V_0 = \mathrm{Span}\{|\delta\rangle\}$ be the corresponding one-dimensional subspace. We now show that the target state $\rho^{\odot}$ is a mixed state over $V_0 \oplus V_1$, where $V_1 = \mathrm{Span}\{|\phi_y\rangle : y \in [M]\}$ is the $(M-1)$-dimensional subspace spanned by states $|\phi_y\rangle = \sqrt{1-(N/M)}|\psi_y\rangle - \sqrt{N/M}|\bar{\psi}_y\rangle$, $|\psi_y\rangle$ being the uniform superposition over functions f such that $y \in \mathrm{Im}(f)$, and $|\bar{\psi}_y\rangle$ the uniform superposition over functions f such that $y \notin \mathrm{Im}(f)$. This subspace corresponds to the irrep represented by diagrams (λ_N, λ_M) where λ_N contains only one row of N boxes, and λ_M has $M-1$ boxes on the first row and one box on the second. We have for the target state

$$\begin{aligned}
\rho^{\odot} &= \frac{1}{|F|} \sum_{f, f' \in F} \frac{|\mathrm{Im}(f) \cap \mathrm{Im}(f')|}{N} |f'\rangle\langle f| = \frac{1}{M} \sum_{y=1}^{M} |\psi_y\rangle\langle\psi_y| \\
&= \frac{N}{M} |\delta\rangle\langle\delta| + \left(1 - \frac{N}{M}\right) \frac{1}{M} \sum_{y=1}^{M} |\phi_y\rangle\langle\phi_y| \\
&= \frac{N}{M} \rho_0 + \left(1 - \frac{N}{M}\right) \rho_1,
\end{aligned}$$

where ρ_0 and ρ_1 are the maximally mixed states over V_0 and V_1, respectively.

Since we start from state ρ_0 and we want to reach state $\rho^{\odot}$ which has a large weight over ρ_1, the strategy for the lower bound is to show that it is hard to transfer weight from V_0 to V_1. More precisely, we divide all irreps (and by consequence their corresponding subspaces) into two sets: one set of *bad* irreps containing all irreps represented by diagrams (λ_N, λ_M) where λ_N and λ_M only differ in their first row, and one set of *good* irreps containing all the other irreps. By this definition, the irrep corresponding to V_0 is bad, while the irrep corresponding to V_1 is good. The lower bound is based on the fact that it is hard to transfer weight onto good subspaces (in particular V_1) starting from V_0. From now on, we note the irreps only by their part under the first row; (λ, λ') then denotes an irrep of $S_N \times S_M$. Therefore, bad irreps are precisely those such that $\lambda = \lambda'$. Recall from Lemma 16 that constructing an adversary matrix $\tilde{\Gamma}$ amounts to assigning an eigenvalue to each irrep of G. We choose $\tilde{\Gamma}$ such that it has eigenvalue 0 on good irreps, and eigenvalue $\gamma_{|\lambda|}$ on a bad irrep (λ, λ), which only depends on $|\lambda|$, i.e.,

$$\tilde{\Gamma} = \sum_{\lambda} \gamma_{|\lambda|} \Pi_{(\lambda, \lambda)},$$

where $\Pi_{(\lambda, \lambda')}$ is the projector onto the subspace corresponding to the irrep (λ, λ'). We set

$$\gamma_{|\lambda|} = \begin{cases} 1 - \frac{|\lambda|}{\sqrt{N}} & \text{if } \lambda < \sqrt{N} \\ 0 & \text{otherwise.} \end{cases}$$

Therefore, we have $\gamma_0 = 1$ and $0 \leq \gamma_{|\lambda|} \leq 1$ for any λ, and $\tilde{\Gamma}$ is a valid additive adversary matrix. Let V_{bad} denote the direct-sum of the bad subspaces. Since $\rho^{\odot}$ only has overlap N/M over V_{bad}, we have $\mathrm{tr}(\Pi_{\mathrm{bad}} \rho^{\odot}) \leq N/M$. Therefore, we can set the threshold eigenvalue $\tilde{\lambda} = 0$ and the base success probability $\eta = N/M$.

From Theorem 17, we see that we need to compute the norm of a matrix Δ_x^l for each irrep l of $G_x = S_{N-1} \times S_M$. We show that these matrices are non-zero only for three different types of irreps of G_x. Indeed, for irreps k of G and l of G_x, the quantity $\gamma_k \mathrm{tr}\left[\Pi_y^x \Pi_k \Pi_y^x \Pi_{k_1 m_1 \leftarrow k_2 m_2}^l\right]$ is non-zero only if: ① k is a bad irrep (otherwise $\gamma_k = 0$); ② k and l restrict to a common irrep of $G_{xy} = S_{N-1} \times S_{M-1}$ (otherwise the product of the projectors is zero). The restrictions of an irrep (λ, λ') of G to G_{xy} are obtained by removing one box from each of the diagrams λ and λ'. Similarly, the restrictions of an irrep (λ, λ') of G_x to G_{xy} are obtained by removing one box from λ'. ③ Note that not all irreps of G_{xy} appear in the projector Π_y^x, as it projects on all injective functions such that $f(x) = y$. Therefore, this set is isomorphic to the set of injective functions from $[N-1]$ to $[M-1]$, and we know that the irrep $\mathcal{U}$ acting on this set is multiplicity-free, and that only irreps (λ, λ') where λ is contained in λ' can occur. Altogether, this implies that only three type of irreps of $G_x = S_{N-1} \times S_M$ lead to non-zero matrices:

1) $l = (\lambda, \lambda)$: Same diagram for S_{N-1} and S_M below the first row. This irrep has multiplicity one since there is only one way to induce to a valid irrep of $S_N \times S_M$, by adding a box in the first row of the left diagram, leading to irrep $k = (\lambda, \lambda)$.
2) $l = (\lambda, \lambda^+)$: Diagram for S_M has one additional box below the first row. This irrep has multiplicity two since there are two ways to induce to a valid irrep of $S_N \times S_M$, by adding a box either in the first row, leading to $k = (\lambda, \lambda^+)$, or at the missing place below the first row, leading to $k = (\lambda^+, \lambda^+)$.
3) $l = (\lambda, \lambda^{++})$: Diagram for S_M has two additional boxes below the first row. This irrep has multiplicity three since there are three ways to induce to a valid irrep of $S_N \times S_M$, by adding a box either in the first row, leading to $k = (\lambda, \lambda^+)$, or at to one of the missing places below the first row, leading to $k = (\lambda^+, \lambda^{++})$.

Let us now consider these three cases separately.

Case (λ, λ): Since this irrep has multiplicity one, we just need to compute a scalar. As an irrep of $S_{N-1} \times S_M$, (λ, λ) restricts to only one valid irrep of $S_{N-1} \times S_{M-1}$, by removing a box on the first row of the right diagram, therefore this irrep is also labeled (λ, λ). Inducing from this irrep of $S_{N-1} \times S_{M-1}$ to $S_N \times S_M$, we obtain three valid irreps, two "bad" ones, (λ, λ) and (λ^+, λ^+), and a good one, (λ, λ^+). To differentiate between projectors of irreps of the different groups, we will from now on use superscripts (for example $\Pi^{N,M}_{\lambda,\lambda}$ denotes a projector on the irrep (λ, λ) of $S_N \times S_M$). We therefore have from Theorem 17

$$\begin{aligned}\Delta_x^{\lambda,\lambda} =& \frac{\gamma_{|\lambda|}}{d^{N-1,M}_{\lambda,\lambda}} \sum_y \mathrm{tr}\left[\Pi^x_y \Pi^{N,M}_{\lambda,\lambda} \Pi^x_y \Pi^{N-1,M}_{\lambda,\lambda}\right] \\ &+ \frac{\gamma_{|\lambda|+1}}{d^{N-1,M}_{\lambda,\lambda}} \sum_y \mathrm{tr}\left[\Pi^x_y \Pi^{N,M}_{\lambda^+,\lambda^+} \Pi^x_y \Pi^{N-1,M}_{\lambda,\lambda}\right] - \gamma_{|\lambda|} \\ =& \frac{M\gamma_{|\lambda|}}{d^{N-1,M}_{\lambda,\lambda} d^{N-1,M-1}_{\lambda,\lambda}} \cdot \mathrm{tr}\left[\Pi^{N-1,M-1}_{\lambda,\lambda} \Pi^{N,M}_{\lambda,\lambda}\right] \\ &\cdot \mathrm{tr}\left[\Pi^{N-1,M-1}_{\lambda,\lambda} \Pi^{N-1,M}_{\lambda,\lambda}\right] \\ &+ \frac{M\gamma_{|\lambda|+1}}{d^{N-1,M}_{\lambda,\lambda} d^{N-1,M-1}_{\lambda,\lambda}} \cdot \mathrm{tr}\left[\Pi^{N-1,M-1}_{\lambda,\lambda} \Pi^{N,M}_{\lambda^+,\lambda^+}\right] \\ &\cdot \mathrm{tr}\left[\Pi^{N-1,M-1}_{\lambda,\lambda} \Pi^{N-1,M}_{\lambda,\lambda}\right] - \gamma_{|\lambda|},\end{aligned}$$

where we have used Lemma 18 and the fact that all terms in the sum over y are equal by symmetry.

We also have

$$\mathrm{tr}\left[\Pi^{N-1,M-1}_{\lambda,\lambda} \Pi^{N,M}_{\lambda,\lambda}\right] = \mathrm{tr}\left[\Pi^{N-1,M-1}_{\lambda,\lambda} \Pi^{N-1,M}_{\lambda,\lambda}\right], \quad (2)$$

$$\mathrm{tr}\left[\Pi^{N-1,M-1}_{\lambda,\lambda} \Pi^{N,M}_{\lambda^+,\lambda^+}\right] = \mathrm{tr}\left[\Pi^{N-1,M-1}_{\lambda,\lambda} \Pi^{N,M-1}_{\lambda^+,\lambda}\right], \quad (3)$$

since the only way for (λ, λ) as an irrep of $S_N \times S_M$ to restrict to (λ, λ) as an irrep of $S_{N-1} \times S_{M-1}$ is to first restrict to (λ, λ) as an irrep of $S_{N-1} \times S_M$, and similarly for (λ^+, λ^+). Therefore, we only have two traces to compute. For the first one, we consider the maximally mixed state $\rho^{N-1,M-1}_{\lambda,\lambda}$ over the corresponding irrep. By inducing from S_{M-1} to S_M we find that its overlap over the irrep (λ, λ) of $S_{N-1} \times S_M$ is given by

$$\mathrm{tr}\left[\rho^{N-1,M-1}_{\lambda,\lambda} \Pi^{N-1,M}_{\lambda,\lambda}\right] = \frac{d^{N-1,M}_{\lambda,\lambda}}{M d^{N-1,M-1}_{\lambda,\lambda}} = \frac{d^M_\lambda}{M d^{M-1}_\lambda}.$$

Similarly, we obtain

$$\mathrm{tr}\left[\rho^{N-1,M-1}_{\lambda,\lambda} \Pi^{N,M-1}_{\lambda^+,\lambda}\right] = \frac{d^{N,M-1}_{\lambda^+,\lambda}}{N d^{N-1,M-1}_{\lambda,\lambda}} = \frac{d^N_{\lambda^+}}{N d^{N-1}_\lambda},$$

and finally

$$\begin{aligned}\Delta_x^{\lambda,\lambda} &= \gamma_{|\lambda|} \frac{d^M_\lambda}{M d^{M-1}_\lambda} + \gamma_{|\lambda|+1} \frac{d^N_{\lambda^+}}{N d^{N-1}_\lambda} - \gamma_{|\lambda|} \\ &= \frac{1}{\sqrt{N}} + O(\frac{1}{N}),\end{aligned}$$

where we have used the hook-length formula for dimensions of irreps and the fact that the number of boxes $|\lambda|$ below the first row is at most $\sqrt{N}$, otherwise $\gamma_{|\lambda|} = 0$.

Case (λ, λ^+): This irrep has multiplicity two, so we need to compute a 2×2 matrix. Let $(\lambda, \lambda^+, 1)$ denote the copy of (λ, λ^+) irrep of $S_{N-1} \times S_M$ which is inside the (λ^+, λ^+) irrep of $S_N \times S_M$. Let $(\lambda, \lambda^+, 2)$ denote the copy of (λ, λ^+) irrep of $S_{N-1} \times S_M$ which is inside the (λ, λ^+) irrep of $S_N \times S_M$. Let the first row and the first column of $\Delta_x^{\lambda,\lambda^+}$ be indexed by $(\lambda, \lambda^+, 1)$ and the second row and the second column be indexed by $(\lambda, \lambda^+, 2)$.

An irrep (λ, λ^+) of $S_{N-1} \times S_M$ restricts to two valid irreps of $S_{N-1} \times S_{M-1}$: (λ, λ) and (λ, λ^+). Those two irreps can be induced to the following bad irreps of $S_N \times S_M$: (λ, λ) and any irrep (λ', λ') which has one more square below the first row than λ. (λ' may be equal or different from λ^+.)

For brevity, we denote $\Delta_x^{\lambda,\lambda^+}$ simply by Δ. Since $(\lambda, \lambda^+, 1)$ is contained inside a bad irrep of $S_N \times S_M$, we have

$$\begin{aligned}\Delta_{1,1} =& \frac{\gamma_{|\lambda|}}{d^{N-1,M}_{\lambda,\lambda^+}} \sum_y \mathrm{tr}\left[\Pi^x_y \Pi^{N,M}_{\lambda,\lambda} \Pi^x_y \Pi^{N-1,M}_{\lambda,\lambda^+,1}\right] \\ &+ \frac{\gamma_{|\lambda|+1}}{d^{N-1,M}_{\lambda,\lambda^+}} \sum_{\lambda'} \sum_y \mathrm{tr}\left[\Pi^x_y \Pi^{N,M}_{\lambda',\lambda'} \Pi^x_y \Pi^{N-1,M}_{\lambda,\lambda^+,1}\right] - \gamma_{|\lambda|+1} \\ =& \frac{M\gamma_{|\lambda|}}{d^{N-1,M}_{\lambda,\lambda^+} d^{N-1,M-1}_{\lambda,\lambda}} \mathrm{tr}\left[\Pi^{N-1,M-1}_{\lambda,\lambda} \Pi^{N,M}_{\lambda,\lambda}\right] \\ &\cdot \mathrm{tr}\left[\Pi^{N-1,M-1}_{\lambda,\lambda} \Pi^{N-1,M}_{\lambda,\lambda^+,1}\right] \\ &+ \frac{M\gamma_{|\lambda|+1}}{d^{N-1,M}_{\lambda,\lambda^+} d^{N-1,M-1}_{\lambda,\lambda}} \left(\sum_{\lambda'} \mathrm{tr}\left[\Pi^{N-1,M-1}_{\lambda,\lambda} \Pi^{N,M}_{\lambda',\lambda'}\right]\right) \\ &\cdot \mathrm{tr}\left[\Pi^{N-1,M-1}_{\lambda,\lambda} \Pi^{N-1,M}_{\lambda,\lambda^+,1}\right] \\ &+ \frac{M\gamma_{|\lambda|+1}}{d^{N-1,M}_{\lambda,\lambda^+} d^{N-1,M-1}_{\lambda,\lambda^+}} \mathrm{tr}\left[\Pi^{N-1,M-1}_{\lambda,\lambda^+} \Pi^{N,M}_{\lambda^+,\lambda^+}\right] \\ &\cdot \mathrm{tr}\left[\Pi^{N-1,M-1}_{\lambda,\lambda^+} \Pi^{N-1,M}_{\lambda,\lambda^+,1}\right] - \gamma_{|\lambda|+1}.\end{aligned}$$

We start by evaluating the sum

$$\sum_{\lambda'} \mathrm{tr}\left[\Pi^{N-1,M-1}_{\lambda,\lambda} \Pi^{N,M}_{\lambda',\lambda'}\right].$$

We consider the maximally mixed state $\rho^{N-1,M-1}_{\lambda,\lambda}$ over the corresponding irrep of $S_{N-1} \times S_{M-1}$. By inducing λ from S_{N-1} to S_N, we find that the dimension of the induced representation is $N d^{N-1}_\lambda$ and the induced representation decomposes into irrep λ of S_N, with dimension d^N_λ and irreps λ'. Therefore,

$$\sum_{\lambda'} \mathrm{tr}\left[\Pi^{N,M}_{\lambda',\lambda'} \rho^{N-1,M-1}_{\lambda,\lambda}\right] = 1 - \frac{d^N_\lambda}{N d^{N-1}_\lambda} \leq 1 - \frac{1}{N} \quad (4)$$

where the inequality follows by comparing the hook-length formulas of d^N_λ and d^{N-1}_λ. Similarly, we have

$$\mathrm{tr}\left[\Pi^{N,M}_{\lambda,\lambda} \rho^{N-1,M-1}_{\lambda,\lambda}\right] = O\left(\frac{1}{N}\right). \quad (5)$$

We now evaluate a similar quantity for $\rho^{N-1,M-1}_{\lambda,\lambda^+}$. By inducing λ^+ from S_{M-1} to S_M, we find that the dimension of the induced representation is $Md^{M-1}_{\lambda^+}$ and the induced representation decomposes into irrep λ^+ of S_M, with dimension d^M_λ and irreps λ^{++} which have one more square below the first row than λ^+. Therefore,

$$\mathrm{tr}\left[\Pi^{N,M}_{\lambda^+,\lambda^+}\rho^{N-1,M-1}_{\lambda,\lambda^+}\right] = \frac{d^M_\lambda}{Md^{M-1}_\lambda} = O\left(\frac{1}{M}\right). \qquad (6)$$

By using eqs. (4), (5) and (6), we have

$$\Delta_{1,1} = \frac{M\gamma_{|\lambda|+1}}{d^{N-1,M}_{\lambda,\lambda^+}}\mathrm{tr}\left[\Pi^{N-1,M-1}_{\lambda,\lambda}\Pi^{N-1,M}_{\lambda,\lambda^+,1}\right] + O\left(\frac{1}{N}\right) - \gamma_{|\lambda|+1}. \qquad (7)$$

We have

$$\mathrm{tr}\left[\Pi^{N-1,M-1}_{\lambda,\lambda}\Pi^{N-1,M}_{\lambda,\lambda^+,1}\right] = \mathrm{tr}\left[\Pi^{N-1,M-1}_{\lambda,\lambda}\Pi^{N,M}_{\lambda^+,\lambda^+}\right]$$

because the other irreps of $S_{N-1}\times S_M$ contained in the irrep (λ^+,λ^+) of $S_N\times S_M$ have no overlap with the irrep (λ,λ) of $S_{N-1}\times S_{M-1}$. Let $\rho^{N-1,M-1}_{\lambda,\lambda}$ be the completely mixed state over (λ,λ). Then,

$$\begin{aligned}\mathrm{tr}\left[\Pi^{N-1,M-1}_{\lambda,\lambda}\Pi^{N,M}_{\lambda^+,\lambda^+}\right] &= d^{N-1,M-1}_{\lambda,\lambda}\mathrm{tr}\left[\Pi^{N,M}_{\lambda^+,\lambda^+}\rho^{N-1,M-1}_{\lambda,\lambda}\right]\\ &= d^{N-1,M-1}_{\lambda,\lambda}\frac{d^N_{\lambda^+}}{Nd^{N-1}_\lambda}.\end{aligned}$$

Here, the second equality follows by inducing λ from S_{N-1} to S_N. We have

$$d^{N-1,M-1}_{\lambda,\lambda}\frac{d^N_{\lambda^+}}{Nd^{N-1}_\lambda} = d^{N-1}_\lambda d^{M-1}_\lambda\frac{d^N_{\lambda^+}}{Nd^{N-1}_\lambda} = \frac{d^{M-1}_\lambda d^N_{\lambda^+}}{N}.$$

By matching up the terms in hook-length formulas, we have

$$d^{M-1}_\lambda d^N_{\lambda^+} = \left(1 + O\left(\frac{1}{N}\right)\right)\frac{N}{M}d^{N-1}_\lambda d^M_{\lambda^+}. \qquad (8)$$

Therefore,

$$\mathrm{tr}\left[\Pi^{N-1,M-1}_{\lambda,\lambda}\Pi^{N,M}_{\lambda^+,\lambda^+}\right] = \left(1 + O\left(\frac{1}{N}\right)\right)\frac{d^{N-1,M}_{\lambda,\lambda^+}}{M} \qquad (9)$$

and

$$\Delta_{1,1} = O\left(\frac{1}{N}\right)$$

Similarly to eq. (7), we have

$$\Delta_{2,2} = \frac{M\gamma_{|\lambda|+1}}{d^{N-1,M}_{\lambda,\lambda^+}}\mathrm{tr}\left[\Pi^{N-1,M-1}_{\lambda,\lambda}\Pi^{N-1,M}_{\lambda,\lambda^+,2}\right] + O\left(\frac{1}{N}\right). \qquad (10)$$

We have

$$\begin{aligned}\mathrm{tr}\left[\Pi^{N-1,M-1}_{\lambda,\lambda}\Pi^{N-1,M}_{\lambda,\lambda^+,2}\right] &= \mathrm{tr}\left[\Pi^{N-1,M-1}_{\lambda,\lambda}\Pi^{N,M}_{\lambda,\lambda^+}\right]\\ &= d^{N-1,M-1}_{\lambda,\lambda}\mathrm{tr}\left[\Pi^{N,M}_{\lambda,\lambda^+}\rho^{N-1,M-1}_{\lambda,\lambda}\right],\end{aligned}$$

because the other irreps of $S_{N-1}\times S_M$ contained in the irrep (λ,λ^+) of $S_N\times S_M$ have no overlap with the irrep (λ,λ) of $S_{N-1}\times S_{M-1}$.

By inducing λ from S_{M-1} to S_M, we get

$$\mathrm{tr}\left[\Pi^{N,M}_{\lambda,\lambda^+}\rho^{N-1,M-1}_{\lambda,\lambda}\right] + \mathrm{tr}\left[\Pi^{N,M}_{\lambda^+,\lambda^+}\rho^{N-1,M-1}_{\lambda,\lambda}\right] = \frac{d^M_{\lambda^+}}{Md^{M-1}_\lambda}. \qquad (11)$$

By inducing λ from S_{N-1} to S_N, we get

$$\mathrm{tr}\left[\Pi^{N,M}_{\lambda^+,\lambda^+}\rho^{N-1,M-1}_{\lambda,\lambda}\right] = \frac{d^N_{\lambda^+}}{Nd^{N-1}_\lambda}. \qquad (12)$$

By subtracting eq. (12) from eq. (11), we get

$$\mathrm{tr}\left[\Pi^{N-1,M-1}_{\lambda,\lambda}\Pi^{N,M}_{\lambda,\lambda^+}\right] = \frac{d^M_{\lambda^+}d^{N-1}_\lambda}{M} - \frac{d^N_{\lambda^+}d^{M-1}_\lambda}{N}.$$

Because of eq. (8),

$$\mathrm{tr}\left[\Pi^{N-1,M-1}_{\lambda,\lambda}\Pi^{N,M}_{\lambda,\lambda^+}\right] = O\left(\frac{d^M_{\lambda^+}d^{N-1}_\lambda}{MN}\right). \qquad (13)$$

By substituting this into eq. (10), we get $\Delta_{2,2} = O(\frac{1}{N})$.

Last, we have to bound $\Delta_{1,2}$ and $\Delta_{2,1}$. Similarly to eq. (7), we have

$$\Delta_{i,j} = \frac{M\gamma_{|\lambda|+1}}{d^{N-1,M}_{\lambda,\lambda^+}}\mathrm{tr}\left[\Pi^{N-1,M-1}_{\lambda,\lambda}\Pi^{N-1,M}_{\lambda,\lambda^+,i\leftarrow j}\right] + O\left(\frac{1}{N}\right).$$

By using Lemma 18 and eqs. (9) and (13), we get

$$\Delta_{i,j} = O\left(\frac{1}{\sqrt{N}}\right).$$

We have shown that $\Delta_{i,j} = O(\frac{1}{\sqrt{N}})$ for all i,j. Therefore, $\|\Delta\| = O(\frac{1}{\sqrt{N}})$.

Case (λ,λ^{++}): We treat this case similarly as the two previous one and get: $|\Delta_{1,1}| = O(1/(MN))$, $|\Delta_{1,2}| = O\left(1/(M\sqrt{N})\right)$, $|\Delta_{1,3}| = O\left(1/(M\sqrt{N})\right)$, $|\Delta_{2,2}| = O(1/M)$, $|\Delta_{2,3}| = O(1/M)$ and $|\Delta_{3,3}| = O(1/M)$. Therefore, all elements of Δ are at most $O(1/M)$, so that $\|\Delta\| = O(1/M)$.

Finally, since the matrices corresponding to all irreps have norm at most $O(1/\sqrt{N})$, we have from Theorem 17 $\left\|\tilde{\Gamma}_x - \tilde{\Gamma}\right\| = O(1/\sqrt{N})$, and in turn

$$Q_\varepsilon(\textsc{Index Erasure}) = \Omega\left((\sqrt{1-\varepsilon} - \sqrt{N/M})^2\sqrt{N}\right).$$

■

ACKNOWLEDGMENTS

The authors thank Ben Reichardt, Robert Špalek and Troy Lee for useful comments. L.M., M.R and J.R. acknowledge support by ARO/NSA under grant W911NF-09-1-0569. A.A. and L.M. acknowledge the support of the European Commission IST project "Quantum Computer Science" QCS 25596. A.A. was also supported by ESF project

1DP/1.1.1.2.0/09/APIA/VIAA/044 and FP7 Marie Curie International Reintegration Grant PIRG02-GA-2007-224886. L.M. also acknowledges the financial support of Agence Nationale de la Recherche under the projects ANR-09-JCJC-0067-01 (CRYQ) and ANR-08-EMER-012 (QRAC).

References

[1] P. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM J. Comput.*, vol. 26, no. 5, pp. 1484–1509, 1997.

[2] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proc. ACM STOC 96*. Philadelphia, Pennsylvania, United States: ACM, 1996, pp. 212–219.

[3] D. Aharonov and A. Ta-Shma, "Adiabatic quantum state generation and statistical zero knowledge," in *Proc. ACM STOC 03*. San Diego, CA, USA: ACM, 2003, pp. 20–29.

[4] J. Köbler, U. Schöning, and J. Toran, *The Graph Isomorphism Problem: Its Structural Complexity*, ser. Progress in Theoretical Computer Science. Birkhäuser Boston, 1993.

[5] S. Hallgren, C. Moore, M. Roetteler, A. Russell, and P. Sen, "Limitations of quantum coset states for graph isomorphism," in *Proc. ACM STOC 06*. Seattle, WA, USA: ACM, May 2006, pp. 604–617.

[6] C. H. Bennett, E. Bernstein, G. Brassard, and U. Vazirani, "Strengths and weaknesses of quantum computing," *SIAM J. Comput.*, vol. 26, no. 5, pp. 1510–1523, 1997.

[7] R. Beals, H. Buhrman, R. Cleve, M. Mosca, and R. de Wolf, "Quantum lower bounds by polynomials," in *Proc. IEEE FOCS 98*. IEEE, p. 352.

[8] S. Aaronson, "Quantum lower bound for the collision problem," in *Proc. ACM STOC 02*. Montreal, Quebec, Canada: ACM, pp. 635–642.

[9] Y. Shi, "Quantum lower bounds for the collision and the element distinctness problems," in *Proc. IEEE FOCS 02*. IEEE, pp. 513–519.

[10] A. Ambainis, "Polynomial degree vs. quantum query complexity," in *Proc. IEEE FOCS 03*. Los Alamitos, CA, USA: IEEE, 2003, p. 230.

[11] H. Klauck, R. Špalek, and R. de Wolf, "Quantum and classical strong direct product theorems and optimal Time-Space tradeoffs," *SIAM J. Comput.*, vol. 36, no. 5, pp. 1472–1493, Jan. 2007.

[12] A. Ambainis, "Quantum lower bounds by quantum arguments," in *Proc. ACM STOC 00*. Portland, Oregon, United States: ACM, 2000, pp. 636–643.

[13] P. Høyer, J. Neerbek, and Y. Shi, "Quantum complexities of ordered searching, sorting, and element distinctness," *Algorithmica*, vol. 34, no. 4, pp. 429–448, March 2008.

[14] H. Barnum and M. Saks, "A lower bound on the quantum query complexity of read-once functions," *J. Comput. Syst. Sci.*, vol. 69, no. 2, pp. 244–258, 2004.

[15] S. Laplante and F. Magniez, "Lower bounds for randomized and quantum query complexity using Kolmogorov arguments," *SIAM J. Comput.*, vol. 38, no. 1, pp. 46–62, 2008.

[16] R. Špalek and M. Szegedy, "All quantum adversary methods are equivalent," *Th. Comput.*, vol. 2, no. 1, pp. 1–18, 2006.

[17] P. Høyer, T. Lee, and R. Špalek, "Negative weights make adversaries stronger," in *Proc. ACM STOC 07*. New York, NY, USA: ACM, 2007, pp. 526–535.

[18] B. W. Reichardt, "Span programs and quantum query complexity: The general adversary bound is nearly tight for every Boolean function," in *Proc. IEEE FOCS 09*. Atlanta, Georgia: IEEE, 2009, pp. 544–551.

[19] T. Lee, R. Mittal, B. W. Reichardt, and R. Špalek, "An adversary for algorithms," unpublished. [Online]. Available: http://arxiv.org/abs/1011.3020

[20] R. Špalek, "The multiplicative quantum adversary," in *Proc. IEEE CCC 08*. Washington, DC, USA: IEEE, 2008, pp. 237–248.

[21] A. Ambainis, "A new quantum lower bound method, with an application to a strong direct product theorem for quantum search," *Theory of Computing*, vol. 6, no. 1, pp. 1–25, 2010.

[22] A. Ambainis, R. Špalek, and R. de Wolf, "A new quantum lower bound method with applications to direct product theorems and Time-Space tradeoffs," *Algorithmica*, vol. 55, no. 3, pp. 422–461, 2007.

[23] A. A. Sherstov, "Strong direct product theorems for quantum communication and query complexity," in *Proc. ACM STOC 11*. San Jose, CA, USA: ACM, June 2011, to appear.

[24] G. Midrijānis, "A polynomial quantum query lower bound for the set equality problem," in *Automata, Languages and Programming*, ser. Lecture Notes in Computer Science, J. Diaz, J. Karhumäki, A. Lepistö, and D. Sannella, Eds. Springer Berlin / Heidelberg, 2004, vol. 3142, pp. 29–41.

[25] G. Brassard, P. Høyer, and A. Tapp, "Quantum algorithm for the collision problem," *ACM. SIGACT News (Cryptology column)*, vol. 28, pp. 14–19, 1997.

[26] P. J. Cameron, *Permutation Groups*, ser. London Mathematical Society Student Texts. Cambridge University Press, Feb. 1999, vol. 45.

[27] R. Bailey, *Associations Schemes: Designed Experiments, Algebra and Combinatorics*, ser. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2004, vol. 84.

[28] J.-P. Serre, *Linear Representations of Finite Groups*, ser. Graduate texts in mathematics. New York, NY, USA: Springer-Verlag, 1977, vol. 42.

[29] B. E. Sagan, *The Symmetric Group: Representations, Combinatorial Algorithms, and Symmetric Functions*, 2nd ed., ser. Graduate texts in mathematics. Springer-Verlag, 2001, vol. 203.