Quantum rejection sampling

Maris Ozols
NEC Laboratories America
and
University of Waterloo, IQC
marozols@yahoo.com

Martin Roetteler
NEC Laboratories America
mroetteler@nec-labs.com

Jérémie Roland
NEC Laboratories America
and
QuIC - Ecole Polytechnique de
Bruxelles
Université Libre de Bruxelles
jroland@ulb.ac.be

ABSTRACT

Rejection sampling is a well-known method to sample from a target distribution, given the ability to sample from a given distribution. The method has been first formalized by von Neumann (1951) and has many applications in classical computing. We define a quantum analogue of rejection sampling: given a black box producing a coherent superposition of (possibly unknown) quantum states with some amplitudes, the problem is to prepare a coherent superposition of the same states, albeit with different target amplitudes. The main result of this paper is a tight characterization of the query complexity of this quantum state generation problem. We exhibit an algorithm, which we call quantum rejection sampling, and analyze its cost using semidefinite programming. Our proof of a matching lower bound is based on the automorphism principle which allows to symmetrize any algorithm over the automorphism group of the problem. Our main technical innovation is an extension of the automorphism principle to continuous groups that arise for quantum state generation problems where the oracle encodes unknown quantum states, instead of just classical data. Furthermore, we illustrate how quantum rejection sampling may be used as a primitive in designing quantum algorithms, by providing three different applications. We first show that it was implicitly used in the quantum algorithm for linear systems of equations by Harrow, Hassidim and Lloyd. Secondly, we show that it can be used to speed up the main step in the quantum Metropolis sampling algorithm by Temme *et al.*. Finally, we derive a new quantum algorithm for the hidden shift problem of an arbitrary Boolean function and relate its query complexity to "water-filling" of the Fourier spectrum.

1. INTRODUCTION

We address the problem of preparing a desired target quantum state into the memory of a quantum computer. It is of course unreasonable to try to find an efficient quantum algorithm to achieve this for general quantum states. Indeed, if any state could be prepared efficiently such diffi-

cult tasks as preparing witnesses for QMA-complete problems [30] could be solved efficiently, a task believed to be impossible even if classical side-information about the quantum state is provided [2]. On the other hand, many interesting computational problems can be related to quantum state generation problems that carry some additional *structure* which might be exploited by an efficient algorithm.

Among the most tantalizing examples of problems that are reducible to quantum state generation is the GRAPH ISO-MORPHISM problem [33] which could be solved by preparing the quantum state $|\Gamma\rangle = \frac{1}{\sqrt{n!}} \sum_{\pi \in S_n} |\Gamma^\pi\rangle$, *i. e.*, the uniform superposition of all the permutations of a given graph Γ. By generating such states for two given graphs, one could then use the standard SWAP-test [13] to check whether the two states are equal or orthogonal and therefore decide whether the graphs are isomorphic or not. Furthermore, it is known that all problems in statistical zero knowledge (SZK) can be reduced to instances of quantum state generation [4], along with gap-versions of closest lattice vector problems [39, 3] and subgroup membership problems for arbitrary subgroups of finite groups [50, 51, 22]. Aside from brute-force attempts that try to solve quantum state preparation by applying sequences of controlled rotations (typically of exponential length) to fix the amplitudes of the target state one qubit at a time, not much is known regarding approaches to tackle the quantum state generation problem while exploiting inherent structure.

In this regard, the only examples we are aware of are (i) a direct approach to generate states described by efficiently computable amplitudes [25], (ii) an approach via adiabatic computing [4] in which a sequence of local Hamiltonians has to be found such that the desired target state is the ground state of a final Hamiltonian and the overlap between intermediate ground states is large, and (iii) recent work on quantum analogues of classical annealing processes [10, 45] and of the Metropolis sampling procedure [46, 52].

Conversely, for some problems a *lower* bound on the complexity of solving a corresponding quantum state generation problem would be desirable, for instance to provide further evidence for the security of quantum money schemes, see *e. g.* [1, 21]. Unfortunately, except for a recent result that generalizes the adversary method to a particular case of quantum state generation problems (see [34] and [7]), very little is known about lower bounds for quantum state generation problems in general.

Rejection sampling.

The classical rejection sampling method[1] was introduced by von Neumann [49] to solve the following *resampling* problem: given the ability to sample according to some probability distribution P, one is asked to produce samples from some other distribution S. Conceptually, the method is extremely simple and works as follows: let $\gamma \leq 1$ be the largest scaling factor such that γS lies under P, formally, $\gamma = \min_k(p_k/s_k)$. We accept a sample k from P with probability $\gamma s_k/p_k$, otherwise we reject it and repeat. The expected number T of samples from P to produce one sample from S is then given by $T = 1/\gamma = \max_k(s_k/p_k)$. See also [20] for further details and analysis of the method for various special cases of P and S. In a setting where access to the distribution P is given by a black box, this has been proved to be optimal by Letac [35]. The rejection sampling technique is at the core of many randomized algorithms and has a wide range of applications, ranging from computer science to statistical physics, where it is used for Monte Carlo simulations, the most well-known example being the Metropolis algorithm [37].

In the same way that quantum state preparation can be considered a quantum analogue of classical sampling, it is natural to study a quantum analogue of the classical resampling problem, *i.e.*, the problem of sampling from a distribution S given the ability to sample from another distribution P. We call this problem *quantum resampling* and define it to be the following analogue of the classical resampling problem: given an oracle generating a quantum state $|\pi^\xi\rangle = \sum_{k=1}^n \pi_k|\xi_k\rangle|k\rangle$, where the amplitudes π_k are known but the states $|\xi_k\rangle$ are unknown, the task is to prepare a target state $|\sigma^\xi\rangle = \sum_{k=1}^n \sigma_k|\xi_k\rangle|k\rangle$ with (potentially) different amplitudes σ_k but the same states $|\xi_k\rangle$. Note that while both the initial amplitudes π_k and the final amplitudes σ_k are fixed and known, the fact that the states $|\xi_k\rangle$ are unknown makes the problem non-trivial.

Related work.

The query complexity of quantum state generation problems was studied in [7], where the adversary method was extended to this model and used to prove a tight lower bound on a specific quantum state generation problem called INDEXERASURE. The adversary method was later extended to quantum state *conversion* problems—where the goal is to convert an initial state into a target state— and shown to be nearly tight in the bounded error case for any problem in this class, which includes as special cases state generation and the usual model of function evaluation [34]. In all these cases, the input to the problem is classical, as the oracle encodes some hidden classical data. This is where the quantum resampling problem differs from those models, as in that case the oracle encodes unknown quantum states.

Grover [24] considered a special case of the quantum resampling problem, where the initial state $|\pi\rangle = \frac{1}{\sqrt{2^n}}\sum_x |x\rangle$ is a uniform superposition and one is given access to an oracle that for given input x provides a classical description of σ_x, the amplitude in the target state $|\sigma\rangle = \sum_x \sigma_x|x\rangle$. We significantly extend the scope of Grover's technique by considering any initial superposition and improving the effi-

---

[1]It is also known as the accept/reject method or "hit-and-miss" sampling.

ciency of the algorithm when only an approximate preparation of the target state is required.

Techniques related to quantum resampling have already been used implicitly as a useful primitive for building quantum algorithms. For instance, it was used in a paper by Harrow, Hassidim, and Lloyd [26] for the problem of solving a system of linear equations $Ax = b$, where A is an invertible matrix over the real or complex numbers whose entries are efficiently computable, and b is a vector. The quantum algorithm in [26] solves the problem of preparing the state $|x\rangle = A^{-1}|b\rangle$ by applying the following three basic steps. First, use phase estimation on the state $|b\rangle = \sum_k b_k|\psi_k\rangle$ to prepare the state $\sum_k b_k|\lambda_k\rangle|\psi_k\rangle$, where $|\psi_k\rangle$ and λ_k denote the eigenstates and eigenvalues of A. Next, convert this state to $\sum_k b_k\lambda_k^{-1}|\lambda_k\rangle|\psi_k\rangle$. Finally, undo the phase estimation to obtain the target state $A^{-1}|b\rangle = \sum_k b_k\lambda_k^{-1}|\psi_k\rangle$. The second step of this procedure performs transformation $\sum_k b_k|\lambda_k\rangle|\psi_k\rangle \mapsto \sum_k b_k\lambda_k^{-1}|\lambda_k\rangle|\psi_k\rangle$ which can be seen as an instance of quantum resampling. Note that other works, such as [15, 44], have used a similar technique—i.e., using phase estimation to simulate some function of an operator— to apply a unitary on an unknown quantum state, rather than preparing one particular sate.

The quantum Metropolis sampling algorithm has been proposed by Temme *et al.* [46] to solve the problem of preparing the thermal state of a quantum Hamiltonian. As it is heavily inspired by the classical Metropolis algorithm, the main step uses an accept/reject rule on random moves between eigenstates of the Hamiltonian. The main complication comes from reverting rejected moves, as the no-cloning principle prevents from keeping a copy of the previous eigenstate. We will show that this step actually reduces to a quantum resampling problem, and that quantum rejection sampling leads to an alternative solution which also provides a speed-up over the technique proposed in [46].

Finally, another type of quantum resampling problem has been considered in a paper by Kitaev and Webb [32] in which the task is to prepare a superposition of basis states with Gaussian-distributed weights along a low-dimensional strip inside a high-dimensional space. Authors solve this problem by applying a sequence of lattice transformation and phase estimation steps.

For us, another important case in point are hidden shift problems over an abelian group A. Here it is easy to prepare a quantum state of the form $|\pi^\xi\rangle = \sum_{w \in A} \hat{f}(w)\chi_w(s)|w\rangle$, where χ_w denotes the characters of A and $\hat{f}$ denotes the Fourier transform of f (see e.g., [18, 29, 42]). If we could eliminate the Fourier coefficients $\hat{f}(w)$ from state $|\pi^\xi\rangle$, we would obtain a state $|\sigma^\xi\rangle = |A|^{-1/2}\sum_{w \in A} \chi_w(s)|w\rangle$ from which the hidden shift s can be easily recovered by applying another Fourier transform. Note that in this case the states $|\xi_k\rangle$ are actually just the complex phases $\chi_w(s)$. We will discuss an application of our general framework to the special case of hidden shift problems in Sect. 5.3.

Our results.

We denote the classical resampling problem mentioned above by SAMPLING$_{P \to S}$, where P and S are probability distributions on the set $[n]$. Note that in its simplest form, this problem is not meaningful in the context of query complexity (indeed, if distribution S is known to begin with, there is no need to use the ability to sample from P). How-

ever, there is a natural modification of the problem, that actually models realistic applications, which does not suffer from this limitation. In this version of the problem, there is a function ξ that deterministically associates some unknown data with each sample, and the problem is to sample pairs $(k, \xi(k))$, where k follows the target distribution. Formally, the problem is therefore defined as follows: given oracle access to a black box producing pairs $(k, \xi(k)) \in [n] \times [d]$ such that k is distributed according to P, where $\xi : [n] \to [d]$ is an unknown function, the problem is to produce a sample $(k, \xi(k))$ such that k is distributed according to S. Note that in this model it is not possible to produce the required samples without using the access to the oracle that produces the samples from P, and the algorithm is therefore restricted to act as in Fig. 1.
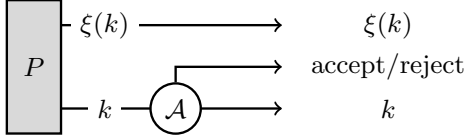

Figure 1: Classical rejection sampling: A black box produces samples k according to a known probability distribution P and accompanied by some unknown classical data $\xi(k)$. The algorithm $\mathcal{A}$ either accepts or rejects these samples, so that accepted samples are distributed according to a target distribution S.

The problem studied in this article is a quantum analogue of $\text{SAMPLING}_{P \to S}$, where probability distributions are replaced by quantum superpositions. More precisely, let $\boldsymbol{\pi}, \boldsymbol{\sigma} \in \mathbb{R}^n$ be such that $\|\boldsymbol{\pi}\|_2 = \|\boldsymbol{\sigma}\|_2 = 1$ and $\pi_k, \sigma_k \geq 0$ for all $k \in [n]$. Let O be a unitary that acts on a default state $|\bar{0}\rangle_{dn} \in \mathbb{C}^d \otimes \mathbb{C}^n$ as $O : |\bar{0}\rangle_{dn} \mapsto |\boldsymbol{\pi}^\xi\rangle := \sum_{k=1}^n \pi_k |\xi_k\rangle |k\rangle$, where $|\xi_k\rangle \in \mathbb{C}^d$ are some fixed unknown normalized quantum states. Given oracle access to unitary black boxes $O, O^\dagger$, the $\text{QSAMPLING}_{\boldsymbol{\pi} \to \boldsymbol{\sigma}}$ problem is to prepare the state $|\boldsymbol{\sigma}^\xi\rangle := \sum_{k=1}^n \sigma_k |\xi_k\rangle |k\rangle$. Note that a special case of this problem is the scenario $d = 1$, when $\xi_k \in \mathbb{C}$ are just unknown phases (complex numbers of absolute value 1).

The main result of this article is a tight characterization of the query complexity of $\text{QSAMPLING}_{\boldsymbol{\pi} \to \boldsymbol{\sigma}}$ for any success probability p (the vector $\boldsymbol{\varepsilon}^p_{\boldsymbol{\pi} \to \boldsymbol{\sigma}}$, as well as the probabilities $p_{\min}, p_{\max}$, will be defined in Sect. 2, intuitively, the vector $\boldsymbol{\varepsilon}^p_{\boldsymbol{\pi} \to \boldsymbol{\sigma}}$ characterizes the amplitudes of the final state prepared by the best algorithm having success probability p):

THEOREM 1. *For $p \in [p_{\min}, p_{\max}]$, the quantum query complexity of $\text{QSAMPLING}_{\boldsymbol{\pi} \to \boldsymbol{\sigma}}$ with success probability p is $Q_{1-p}(\text{QSAMPLING}_{\boldsymbol{\pi} \to \boldsymbol{\sigma}}) = \Theta(1/\|\boldsymbol{\varepsilon}^p_{\boldsymbol{\pi} \to \boldsymbol{\sigma}}\|_2)$. For $p \leq p_{\min}$, the query complexity is 1, and for $p > p_{\max}$, it is infinite.*

Let us note that when $p = p_{\max} = 1$, the query complexity reduces to $\max_k(\sigma_k/\pi_k)$ in analogy with the classical case, except that amplitudes replace probabilities. The lower bound comes from an extension of the automorphism principle (originally introduced in the context of the adversary method [7, 27]) to our framework of quantum state generation problems with quantum oracles. The upper bounds follows from an algorithm based on amplitude amplification that we call quantum rejection sampling. We also show that a modification of this algorithm can also solve a quantum

state conversion problem, which we call strong quantum resampling (SQSAMPLING).

Next, we illustrate the technique by providing different applications. We first show that the main steps in two recent algorithms, namely the quantum algorithm for solving linear systems of equations [26] and the quantum Metropolis sampling algorithm [46], consists in solving quantum state conversion problems which we call $\text{QLINEAREQUATIONS}_\kappa$ and $\text{QMETROPOLISMOVE}_\mathcal{C}$. We then observe that these problems reduce to SQSAMPLING, and can therefore be solved using quantum rejection sampling.

THEOREM 2. *For any $\tilde{\kappa} \in [1, \kappa]$, there is a quantum algorithm that solves $\text{QLINEAREQUATIONS}_\kappa$ with success probability $p = (\boldsymbol{w}^\mathsf{T} \cdot \tilde{\boldsymbol{w}})/(\|\boldsymbol{w}\|_2 \cdot \|\tilde{\boldsymbol{w}}\|_2)$ using an expected number of queries $O(\tilde{\kappa}/\|\tilde{\boldsymbol{w}}\|_2)$, where $w_j := b_j/\lambda_j$, $\tilde{w}_j := b_j/\tilde{\lambda}_j$, and $\tilde{\lambda}_j := \max\{\tilde{\kappa}^{-1}, \lambda_j\}$.*

THEOREM 3. *There is a quantum algorithm that solves $\text{QMETROPOLISMOVE}_\mathcal{C}$ with success probability 1 using an expected number of queries $O(1/\|\boldsymbol{w}^{(i)}\|_2)$.*

Let us note that while the quantum algorithm for linear systems of equations was indeed using this technique implicitly, this was not the case for quantum Metropolis sampling, where quantum rejection sampling provides some speed-up over the original algorithm.

Our final result is an application of quantum rejection sampling to the Boolean hidden shift problem BHSP_f, defined as follows. Let $f : \mathbb{F}_2^n \to \mathbb{F}_2$ be a Boolean function, which is assumed to be completely known. Furthermore, we are given oracle access to *shifted function* $f_s(x) := f(x + s)$ for some unknown bit string $s \in \mathbb{F}_2^n$, with the promise that there exists x such that $f(x + s) \neq f(x)$. The task is to find the bit string s.

We show that we can solve this problem by solving the $\text{QSAMPLING}_{\boldsymbol{\pi} \to \boldsymbol{\sigma}}$ problem for $\boldsymbol{\pi}$ corresponding to the Fourier spectrum of f, and $\boldsymbol{\sigma}$ being a uniformly flat vector. This leads to the following upper bound which expresses the complexity of our quantum algorithm for the Boolean hidden shift problem in terms of a vector $\boldsymbol{\varepsilon}^p_{\boldsymbol{\pi} \to \boldsymbol{\sigma}}$ (defined in Sect. 3) that can be thought of as a "water-filling" of the Fourier spectrum of f:

THEOREM 4. *Let f be a Boolean function and $\hat{f}$ be its Fourier transform. For any p, we have $Q_{1-p}(\text{BHSP}_f) = O(1/\|\boldsymbol{\varepsilon}^p_{\boldsymbol{\pi} \to \boldsymbol{\sigma}}\|_2)$, where components of $\boldsymbol{\pi}$ and $\boldsymbol{\sigma}$ are given by $\pi_w = |\hat{f}_w|$ and $\sigma_w = 1/\sqrt{2^n}$ for $w \in \mathbb{F}_2^n$.*

As special cases of this theorem we obtain the quantum algorithms for hidden shift problem for delta functions, which leads to the Grover search algorithm [23], and for bent functions, which are functions that have perfectly flat absolute values in their Fourier spectrum [42]. In general, the complexity of the algorithm is limited by the smallest Fourier coefficient of the function. By ignoring small Fourier coefficients, one can decrease the complexity of the algorithm, at the cost of a lower success probability. The final success probability can nevertheless be amplified using repetitions and by constructing a procedure to check a candidate shift, which leads to the following theorem.

THEOREM 5. *Let f be a Boolean function and $\hat{f}$ be its Fourier transform. Moreover, let $p, \gamma \in [0, 1]$ be such that*

$\|\hat{\varepsilon}\|_1 = \sqrt{2^n p}$, where $\varepsilon_w = \min\{|\hat{f}_w|, \gamma/\sqrt{2^n}\}$. Then, for any $\delta > 0$, we have $Q_\delta(\mathrm{BHSP}_f) = O\big(\frac{1}{\sqrt{p}}(1/\|\varepsilon\|_2 + 1/\sqrt{\mathrm{I}_f})\big)$.

2. DEFINITION OF THE PROBLEM

In this section, we define different notions related to the quantum resampling problem. It is important to note that this problem goes beyond the usual model of quantum query complexity, where the goal is to compute a function depending on some unknown classical data that can be accessed via an oracle (see [14] for a complete survey). In the usual model, the algorithm is therefore quantum but both its input and output are classical. A first extension of this model is the case were the output is also quantum, that is, the goal is to generate a target quantum state depending on some unknown classical data hidden by the oracle. The quantum adversary method has recently been extended to this model by [7], where is was used to characterize the query complexity if a quantum state generation problem called INDEXERASURE. In both the usual model and this extension, the oracle acts as $O_x : |b\rangle|i\rangle \mapsto |b + x_i\rangle|i\rangle$, where x is the hidden classical data. However, the quantum resampling problem corresponds to another extension of these models, where the input is also quantum, in the sense that the oracle hides unknown quantum states instead of classical data. Let us now define this extended model more precisely.

Definition 1. Let $\mathcal{O} := \{O_x : x \in \mathcal{X}\}$ and $\Psi := \{|\psi_x\rangle : x \in \mathcal{X}\}$, respectively, be sets of quantum oracles (*i.e.*, unitaries) and target quantum states labeled by elements of some set $\mathcal{X}$. Then $\mathcal{P} := (\mathcal{O}, \Psi, \mathcal{X})$ describes the following *quantum state generation problem:* given an oracle O_x for some unknown value of $x \in \mathcal{X}$, prepare a state

$$|\psi\rangle = \sqrt{p}|\psi_x\rangle|\bar{0}\rangle + |\mathrm{error}_x\rangle,$$

where p is the desired success probability, $|\bar{0}\rangle$ is a normalized standard state for some workspace and $|\mathrm{error}_x\rangle$ is an arbitrary error state with norm at most $\sqrt{1-p}$. The quantum query complexity of $\mathcal{P}$ is the minimum number of queries to O_x or $O_x^\dagger$ necessary to solve $\mathcal{P}$ with success probability p and will be denoted by $Q_{1-p}(\mathcal{P})$.

Intuitively, we want the final state $|\psi\rangle$ to have a component of length at least $\sqrt{p}$ in the direction of $|\psi_x\rangle|\bar{0}\rangle$. We can restate the condition $\||\mathrm{error}_x\rangle\|_2 \leq \sqrt{1-p}$ as follows:

$$1 - p \geq \||\psi\rangle - \sqrt{p}|\psi_x\rangle|\bar{0}\rangle\|_2^2 = 1 + p - 2\mathrm{Re}\big[\langle\psi| \cdot \sqrt{p}|\psi_x\rangle|\bar{0}\rangle\big],$$

or equivalently:

$$\mathrm{Re}\big[\langle\psi| \cdot |\psi_x\rangle|\bar{0}\rangle\big] \geq \sqrt{p}. \tag{1}$$

Note that the main difference of the above definition with the usual model of quantum query complexity, and its extension to quantum state generation in [7], is that the oracle is not restricted to act as $O_x : |b\rangle|i\rangle \mapsto |b + x_i\rangle|i\rangle$.

We now formally define QSAMPLING$_{\boldsymbol{\pi} \to \boldsymbol{\sigma}}$ as a special case of quantum state generation problem. Throughout this article, we fix positive integers d, n and we assume that $\boldsymbol{\pi}, \boldsymbol{\sigma} \in \mathbb{R}^n$ are vectors such that $\|\boldsymbol{\pi}\|_2 = \|\boldsymbol{\sigma}\|_2 = 1$ and $\pi_k, \sigma_k \geq 0$ for all $k \in [n]$. We also use the notation $|\boldsymbol{\pi}\rangle := \sum_{k=1}^n \pi_k|k\rangle$ and $|\boldsymbol{\sigma}\rangle := \sum_{k=1}^n \sigma_k|k\rangle$. For simplicity, we assume that $\sigma_k > 0$ for all $k \in [n]$, but the general case can easily be obtained by taking the limit $\sigma_k \to 0$.

Let $\xi = (|\xi_k\rangle \in \mathbb{C}^d : k \in [n])$ be an ordered list of normalized quantum states. Then any unitary that maps a

default state $|\bar{0}\rangle_{dn}$ to $|\boldsymbol{\pi}^\xi\rangle := \sum_{k=1}^n \pi_k|\xi_k\rangle|k\rangle$ is a valid oracle for QSAMPLING$_{\boldsymbol{\pi} \to \boldsymbol{\sigma}}$. Therefore, we will label valid oracles by a pair (ξ, u), where ξ denotes the states hidden by the oracle, and u defines how the oracle acts on states that are orthogonal to $|\bar{0}\rangle_{dn}$. More explicitly, we fix a default oracle $O \in \mathrm{U}(dn)$ as a unitary that acts on $|\bar{0}\rangle_{dn}$ as $O|\bar{0}\rangle_{dn} = |\bar{0}\rangle_d|\boldsymbol{\pi}\rangle$, and arbitrarily on the orthogonal complement. We then use O as a reference point to define the remaining oracles:

$$O_{\xi,u} := V_\xi \, O \, u, \tag{2}$$

where $u \in \mathrm{U}(dn)$ is a unitary such that $u|\bar{0}\rangle_{dn} = |\bar{0}\rangle_{dn}$ and V_ξ is a unitary that acts on $|\bar{0}\rangle_d|k\rangle$ as $V_\xi|\bar{0}\rangle_d|k\rangle = |\xi_k\rangle|k\rangle$ for any $k \in [n]$, and arbitrarily on the orthogonal complement of these states, so that $O_{\xi,u}|\bar{0}\rangle_{dn} = V_\xi O|\bar{0}\rangle_{dn} = V_\xi \sum_{k=1}^n \pi_k|\bar{0}\rangle_d|k\rangle = \sum_{k=1}^n \pi_k|\xi_k\rangle|k\rangle = |\boldsymbol{\pi}^\xi\rangle$ (note that how O and V_ξ are defined on the orthogonal complements is irrelevant as it only affects the exact labeling, but not the set of valid oracles).

Definition 2. The *quantum resampling problem*, denoted by QSAMPLING$_{\boldsymbol{\pi} \to \boldsymbol{\sigma}}$, is an instance of quantum state generation problem $(\mathcal{O}, \Psi, \mathcal{X})$ with

$$\mathcal{X} := \{(\xi, u) : \xi = (|\xi_1\rangle, \dots, |\xi_n\rangle) \in (\mathbb{C}^d)^n, u \in S\},$$
$$S := \{u \in \mathrm{U}(dn) : u|\bar{0}\rangle_{dn} = |\bar{0}\rangle_{dn}\} \cong \mathrm{U}(dn - 1).$$

Oracles in $\mathcal{O}$ that are hiding the states $|\boldsymbol{\pi}^\xi\rangle$ are defined according to Eq. (2) and the corresponding target states are defined by $|\boldsymbol{\sigma}^\xi\rangle := V_\xi|\bar{0}\rangle_d|\boldsymbol{\sigma}\rangle = \sum_{k=1}^n \sigma_k|\xi_k\rangle|k\rangle$.

Let us note that the target states only depend on the index ξ, and not u. Moreover, note that amplitudes π_k and σ_k can be assumed to be real and positive without loss of generality, as any phase can be corrected using a controlled-phase gate, which does not require any oracle call since $\boldsymbol{\pi}$ and $\boldsymbol{\sigma}$ are fixed and known.

In [34], Lee *et al.* have proposed another extension of the query complexity model for quantum state generation of [7] by considering quantum state conversion, where the problem is now to convert a given quantum state into another quantum state, instead of generating a target quantum state from scratch. They have extended the adversary method to this model and showed that it is approximately tight in the bounded-error case for any quantum state conversion problem with a classical oracle. Here, we define a model that combines both extensions (from classical to quantum oracles as well as from state generation to state conversion), hence subsuming all previous models (see Fig. 2).

Definition 3. Let $\mathcal{O} := \{O_x : x \in \mathcal{X}\}$, $\Phi := \{|\phi_x\rangle : x \in \mathcal{X}\}$ and $\Psi := \{|\psi_x\rangle : x \in \mathcal{X}\}$, respectively, be sets of quantum oracles (*i.e.*, unitaries), initial quantum states and target quantum states labeled by elements of some set $\mathcal{X}$. Then $\mathcal{P} := (\mathcal{O}, \Phi, \Psi, \mathcal{X})$ describes the following *quantum state conversion problem:* given an oracle O_x for some unknown value of $x \in \mathcal{X}$ and a copy of the corresponding initial state $|\phi_x\rangle$, prepare a state

$$|\psi\rangle = \sqrt{p}|\psi_x\rangle|\bar{0}\rangle + |\mathrm{error}_x\rangle, \tag{3}$$

where p is the desired success probability, $|\bar{0}\rangle$ is a normalized standard state for some workspace and $|\mathrm{error}_x\rangle$ is an arbitrary error state with norm at most $\sqrt{1-p}$. Again, $Q_{1-p}(\mathcal{P})$ will denote the quantum query complexity of $\mathcal{P}$.

We also define a strong version of the quantum resampling problem, which is a special case of the state conversion problem. Compared to the original resampling problem, it is made harder due to two modifications. First, instead of being given access to an oracle that maps $|\bar{0}\rangle_{dn}$ to $|\boldsymbol{\pi}^\xi\rangle$, we are only provided with one copy of $|\boldsymbol{\pi}^\xi\rangle$ and an oracle that reflects through it, making this a quantum state conversion problem. The second extension assumes that we only know the ratios between the amplitudes π_k and σ_k for each k, instead of the amplitudes themselves. More precisely, instead of vectors $\boldsymbol{\pi}, \boldsymbol{\sigma} \in \mathbb{R}^n$ specifying the initial and target amplitudes, we fix a single vector $\boldsymbol{\tau} \in \mathbb{R}^n$ such that $\tau_k \geq 0$ and $\max_k \tau_k = 1$, specifying the ratios between those amplitudes. Let us now formally define the stronger version of the quantum resampling problem (this definition is motivated by the applications that will be presented in Sect. 5.1 and 5.2).

Definition 4. Let $P := \{\boldsymbol{\pi} \in \mathbb{R}^n : \|\boldsymbol{\pi}\|_2 = 1, \forall k : \pi_k > 0\}$. The *strong quantum resampling problem* SQSAMPLING$_{\boldsymbol{\tau}}$ is a quantum state conversion problem $(\mathcal{O}, \Phi, \Psi, \mathcal{X})$, where $\mathcal{X} := \{(\xi, \boldsymbol{\pi}) : \xi = (|\xi_1\rangle, \ldots, |\xi_n\rangle) \in (\mathbb{C}^d)^n, \boldsymbol{\pi} \in P\}$, oracles in $\mathcal{O}$ are defined by $O_{\xi,\boldsymbol{\pi}} := \mathrm{ref}_{|\boldsymbol{\pi}^\xi\rangle} = I - 2|\boldsymbol{\pi}^\xi\rangle\langle\boldsymbol{\pi}^\xi|$ with the corresponding initial and target states being $|\boldsymbol{\pi}^\xi\rangle$ and $|\boldsymbol{\sigma}^\xi\rangle = \sum_{k=1}^n \sigma_k |\xi_k\rangle|k\rangle$, respectively, where $\boldsymbol{\sigma} := \boldsymbol{\pi} \circ \boldsymbol{\tau}/\|\boldsymbol{\pi} \circ \boldsymbol{\tau}\|_2$ so that $\sigma_k/\pi_k \propto \tau_k$.
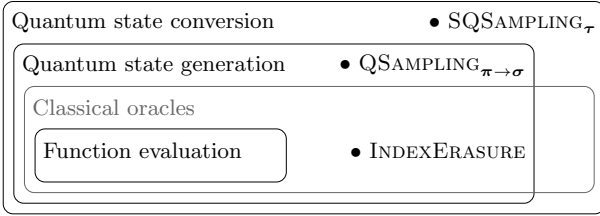

Figure 2: **Comparison of different classes of problems in quantum query complexity. The case of function evaluation has been extensively studied in the literature. The extension to quantum state generation with classical oracles, as well as the problem** INDEXERASURE **which belongs to that class, have been studied in [7]. The adversary method has been extended to the case of quantum state conversion with classical oracles in [34]. The problems** QSAMPLING$_{\boldsymbol{\pi}\to\boldsymbol{\sigma}}$ **and** SQSAMPLING$_{\boldsymbol{\tau}}$ **studied in this article use quantum oracles and therefore belong to yet another extension of the quantum query complexity model.**

The relationship between different classes of query complexity problems introduced above, and strong and regular quantum rejection sampling as special instances of them are summarized in Fig. 2. Our main result is that the quantum query complexities of QSAMPLING$_{\boldsymbol{\pi}\to\boldsymbol{\sigma}}$ and SQSAMPLING$_{\boldsymbol{\tau}}$ for any success probability p depend on a vector $\boldsymbol{\varepsilon}^p_{\boldsymbol{\pi}\to\boldsymbol{\sigma}}$ defined as follows.

Definition 5. For any $\boldsymbol{\pi}, \boldsymbol{\sigma}$, let us define the following

quantities

$$p_{\min} := (\boldsymbol{\sigma}^\mathsf{T} \cdot \boldsymbol{\pi})^2, \qquad \gamma_{\min} := \min_{k:\pi_k>0}(\pi_k/\sigma_k),$$

$$p_{\max} := \sum_{k:\pi_k>0} \sigma_k^2, \qquad \gamma_{\max} := \max_k(\pi_k/\sigma_k).$$

For any $\gamma \in [\gamma_{\min}, \gamma_{\max}]$, let us define a vector $\boldsymbol{\varepsilon}(\gamma)$ and a scalar $p(\gamma)$ by

$$\varepsilon_k(\gamma) := \min\{\pi_k, \gamma\sigma_k\}, \qquad p(\gamma) := \left(\frac{\boldsymbol{\sigma}^\mathsf{T} \cdot \boldsymbol{\varepsilon}(\gamma)}{\|\boldsymbol{\varepsilon}(\gamma)\|_2}\right)^2.$$

For $p \in [p_{\min}, p_{\max}]$, let $\bar{\gamma} \in [\gamma_{\min}, \gamma_{\max}]$ be such that $p(\bar{\gamma}) = p$ and define a vector $\boldsymbol{\varepsilon}^p_{\boldsymbol{\pi}\to\boldsymbol{\sigma}} := \boldsymbol{\varepsilon}(\bar{\gamma})$.

To see that $\boldsymbol{\varepsilon}^p_{\boldsymbol{\pi}\to\boldsymbol{\sigma}}$ is well-defined, note that $\|\boldsymbol{\varepsilon}(\gamma)\|_2$ is monotonically increasing with γ, while $p(\gamma)$ is monotonically decreasing with γ. More precisely, for $\gamma = \gamma_{\min}$, the vector $\boldsymbol{\varepsilon}(\gamma)$ has components $\varepsilon_k(\gamma) = \gamma\sigma_k$ if $\pi_k \neq 0$ or zero otherwise, and $p(\gamma) = p_{\max}$. For $\gamma = \gamma_{\max}$, we have $\boldsymbol{\varepsilon}(\gamma) = \boldsymbol{\pi}$ and $p(\gamma) = p_{\min}$. Between these extreme cases, $p(\gamma)$ interpolates from $p_{\max}$ to $p_{\min}$, which means that for any $p \in [p_{\min}, p_{\max}]$, there exists a value $\bar{\gamma}$ such that $p(\bar{\gamma}) = p$, which uniquely defines $\boldsymbol{\varepsilon}^p_{\boldsymbol{\pi}\to\boldsymbol{\sigma}}$.

Intuitively, $\boldsymbol{\varepsilon}(\gamma)$ may be interpreted as a "water-filling" vector, where γ defines the water level, and π_k defines the height of "tank" number k. As γ increases from 0 to $\gamma_{\min}$, we have $\varepsilon_k(\gamma) = \gamma\sigma_k$, meaning that all tanks are progressively filled proportionally to γ. When $\gamma > \gamma_{\min}$, some tanks are filled ($\gamma\sigma_k > \pi_k$) and cannot hold more water, while others continue to get filled.

3. QUERY COMPLEXITY OF QUANTUM RESAMPLING

Let us first show that $\boldsymbol{\varepsilon}^p_{\boldsymbol{\pi}\to\boldsymbol{\sigma}}$ defines an optimal feasible point of a certain semidefinite program (SDP). Afterwards we will show that the same SDP characterizes the quantum query complexity of QSAMPLING$_{\boldsymbol{\pi}\to\boldsymbol{\sigma}}$.

LEMMA 1. *Let* $p \in [p_{\min}, p_{\max}]$, *and* $\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^p_{\boldsymbol{\pi}\to\boldsymbol{\sigma}}$. *Then, the following SDP*

$$\max_{M \succeq 0} \mathrm{Tr}\, M \quad s.t. \quad \begin{aligned} &\forall k : \pi_k^2 \geq M_{kk}, \\ &\mathrm{Tr}\big[(\boldsymbol{\sigma} \cdot \boldsymbol{\sigma}^\mathsf{T} - pI)M\big] \geq 0. \end{aligned} \quad (4)$$

has optimal value $\|\boldsymbol{\varepsilon}\|_2^2$, *which is achieved by the rank-1 matrix* $M = \boldsymbol{\varepsilon} \cdot \boldsymbol{\varepsilon}^\mathsf{T}$.

PROOF SKETCH. We first show that $M = \boldsymbol{\varepsilon} \cdot \boldsymbol{\varepsilon}^\mathsf{T}$, where $\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^p_{\boldsymbol{\pi}\to\boldsymbol{\sigma}}$, satisfies the constraints in SDP (1) and therefore constitutes a feasible point. Therefore, the optimal value of (1) is at least $\mathrm{Tr}\, M = \|\boldsymbol{\varepsilon}\|_2^2$. Secondly, we dualize the SDP, and provide a dual-feasible point achieving the same objective value. The fact that this objective value is feasible for both the primal and the dual then implies that this is the optimal value. The details of the proof are given in Appendix A. $\square$

Next, let us prove that SDP (4) provides a lower bound for the QSAMPLING$^p_{\boldsymbol{\pi}\to\boldsymbol{\sigma}}$ problem. In Sect. 4, we will also show that this lower bound is tight by providing an explicit algorithm.

Let us emphasize the fact that the lower bound cannot be obtained from standard methods such as the adversary

method [5, 27] (which has recently been proved to be tight for evaluating functions [40, 41, 34]), nor from its extension to quantum state generation problems [7, 34], because in this case the oracle is also quantum, in the sense that it encodes some unknown quantum state rather than some unknown classical data. To prove lower bounds it is useful to exploit possible symmetries of the problem. We extend the notion of automorphism group [7, 27] to our framework of quantum state generation problems:

Definition 6. We say that G is the *automorphism group* of problem $(\mathcal{O}, \Psi, \mathcal{X})$ if:

1. G acts on $\mathcal{X}$ (and thus implicitly also on $\mathcal{O}$ as $g : O_x \mapsto O_{g(x)}$).

2. For any $g \in G$ there is a unitary U_g such that $U_g|\psi_x\rangle = |\psi_{g(x)}\rangle$ for all $x \in \mathcal{X}$.

3. For any given $g \in G$ it is possible to simulate the oracles $O_{g(x)}$ for all $x \in \mathcal{X}$, using only a constant number of queries to the black box O_x.

While for the standard model of quantum query complexity, where the oracle encodes some unknown classical data, the automorphism group is restricted to be a permutation group and is therefore always finite, in this more general framework the automorphism group can be continuous. For example, in the case of $\mathrm{QSAMPLING}^p_{\boldsymbol{\pi} \to \boldsymbol{\sigma}}$ it will involve the unitary group. Taking such symmetries into account might significantly simplify the analysis of algorithms for the corresponding problem and is the key to prove our lower bound.

LEMMA 2. *Any quantum algorithm for* $\mathrm{QSAMPLING}^p_{\boldsymbol{\pi} \to \boldsymbol{\sigma}}$ *with* $p \in [p_{\min}, p_{\max}]$ *requires at least* $\Omega(1/\|\boldsymbol{\varepsilon}^p_{\boldsymbol{\pi} \to \boldsymbol{\sigma}}\|_2)$ *queries to* O *and* $O^\dagger$.

PROOF. The proof proceeds as follows: we first define a subset of oracles that are sufficiently hard to distinguish to characterize the query complexity of the problem. Exploiting the automorphism group of this subset of oracles, we then show that any algorithm may be symmetrized in such a way that the real part of the amplitudes of the final state prepared by the algorithm does not depend on the specific oracle it was given. These amplitudes define a vector $\bar{\boldsymbol{\gamma}}$ that should satisfy some constraints for the algorithm to have success probability p. Moreover, we can use the hybrid argument to show that the components of $\bar{\boldsymbol{\gamma}}$ should also satisfy some constraints for the algorithm to be able to generate the corresponding state in at most T queries. Putting all these constraints together in an optimization program, we then show that such a vector $\bar{\boldsymbol{\gamma}}$ cannot exist unless T is large enough. This optimization program is then shown to be equivalent to the semidefinite program in Lemma 1, which proves the theorem.

Let us now give the details of the proof. For given $\boldsymbol{\pi}, \boldsymbol{\sigma} \in \mathbb{R}^n$, let us choose a subset of oracles $\mathcal{O}'_{\boldsymbol{\pi}, \boldsymbol{\sigma}} \subset \mathcal{O}_{\boldsymbol{\pi}, \boldsymbol{\sigma}}$ that are hard to distinguish. We choose the states hidden inside oracles to be of the form $|\xi_k\rangle = (-1)^{x_k}|\bar{0}\rangle_d$, where phases are given by some unknown string $x \in \mathbb{F}_2^n$. We also choose u so that any oracle in the subset acts trivially on the d-dimensional register holding the unknown states. In that case, this register is effectively one-dimensional, so we will omit it and write $(-1)^{x_k}$ as a relative phase. More precisely, we consider a set of oracles $\mathcal{O}'_{\boldsymbol{\pi}, \boldsymbol{\sigma}} := \{O_{x,u} : x \in \mathbb{F}_2^n, u \in S\}$,

where

$$S := \{u \in \mathrm{U}(n) : u|\bar{0}\rangle_n = |\bar{0}\rangle_n\} \cong \mathrm{U}(n-1). \qquad (5)$$

As in the general case, we define the first oracle $O_{0,I}$ as a unitary that acts on $|\bar{0}\rangle_n$ as $O_{0,I}|\bar{0}\rangle_n = |\boldsymbol{\pi}\rangle$, and arbitrarily on the orthogonal complement, and we use $O_{0,I}$ as a reference point to define the remaining oracles:

$$O_{x,u} := V_x O_{0,I} u, \quad \text{where} \quad V_x := \sum_{k=1}^n (-1)^{x_k}|k\rangle\langle k|.$$

The set of target states is $\Psi'_{\boldsymbol{\pi}, \boldsymbol{\sigma}} := \{|\boldsymbol{\sigma}(x)\rangle : x \in \mathbb{F}_2^n, u \in S\}$ where $|\boldsymbol{\sigma}(x)\rangle := V_x|\boldsymbol{\sigma}\rangle = \sum_{k=1}^n (-1)^{x_k}\sigma_k|k\rangle$. For the quantum state generation problem corresponding to the restricted set of oracles $\mathcal{O}'_{\boldsymbol{\pi}, \boldsymbol{\sigma}}$, the automorphism group is $G = \mathbb{F}_2^n \times \mathrm{U}(n-1)$ and it acts on itself, *i.e.*, $\mathcal{X} = G$. Note that the target states depend only on x, but u is used for parameterizing the oracles. Intuitively, the reason we need this parameter is that the algorithm should not depend on how the black box acts on states other than $|\bar{0}\rangle_n$ (or how its inverse acts on states other than $|\boldsymbol{\pi}^\xi\rangle$). To make this intuition formal, we will later choose the parameter u for different oracles adversarially.

Let us consider an algorithm that uses T queries to the black box $O_{x,u}$ and its inverse, and let us denote the final state of this algorithm by $|\psi_T(x, u)\rangle$. If we expand the first register in the standard basis, we can express this state as

$$|\psi_T(x, u)\rangle = \sum_{k=1}^n (-1)^{x_k}|k\rangle|\gamma_k(x, u)\rangle.$$

Here the workspace vectors $|\gamma_k(x, u)\rangle$ can have arbitrary dimension and are not necessarily unit vectors, but instead satisfy the normalization constraint $\sum_{k=1}^n \||\gamma_k(x, u)\rangle\|_2^2 = 1$. If the algorithm succeeds with probability p, then according to Eq. (1) for any x and u we have

$$\begin{aligned}
\sqrt{p} &\leq \mathrm{Re}\big[\langle\boldsymbol{\sigma}(x)|\langle\bar{0}| \cdot |\psi_T(x, u)\rangle\big] \\
&= \mathrm{Re}\big[\sum_{k=1}^n \sigma_k \cdot \langle\bar{0}|\gamma_k(x, u)\rangle\big] \\
&= \boldsymbol{\sigma}^\mathsf{T} \cdot \boldsymbol{\gamma}(x, u),
\end{aligned}$$

where $\boldsymbol{\gamma}(x, u)$ is a real vector whose components are given by

$$\gamma_k(x, u) := \mathrm{Re}\big[\langle\bar{0}|\gamma_k(x, u)\rangle\big].$$

Note that $\|\boldsymbol{\gamma}(x, u)\|_2 \leq 1$.

Next, let us show that we can symmetrize the algorithm without decreasing its success probability. We do this by replacing each oracle call by $O_{x+y,uv} = V_y O_{x,u} v$ and correcting the final state by applying $V_y^\dagger$ (see Fig. 3). Let μ be the Haar measure on the set S defined in Eq. (5). We define an operation that symmetrizes a set of states:

$$\overline{|\phi(x, u)\rangle} := \frac{1}{\sqrt{2^n}} \sum_{y \in \mathbb{F}_2^n} \int_{v \in S} \Big[(V_y^\dagger \otimes I)|\phi(x+y, uv)\rangle\Big]|y\rangle|v\rangle \, d\mu(v).$$

If we symmetrize the final state $|\psi_T(x, u)\rangle$, we get

$$\overline{|\psi_T(x, u)\rangle} =$$
$$\frac{1}{\sqrt{2^n}} \sum_{y \in \mathbb{F}_2^n} \int_{v \in S} \sum_{k=1}^n (-1)^{x_k}|k\rangle|\gamma_k(x+y, uv)\rangle|y\rangle|v\rangle \, d\mu(v).$$

Note that the target state $|\boldsymbol{\sigma}(x)\rangle|\bar{0}\rangle$ is already symmetric, so symmetrization only introduces an additional workspace register in a default state (uniform superposition):

$$\overline{|\boldsymbol{\sigma}(x)\rangle|\bar{0}\rangle} = |\boldsymbol{\sigma}(x)\rangle|\bar{0}\rangle \frac{1}{\sqrt{2^n}} \sum_{y \in \mathbb{F}_2^n} \int_{v \in S} |y\rangle|v\rangle \, d\mu(v).$$
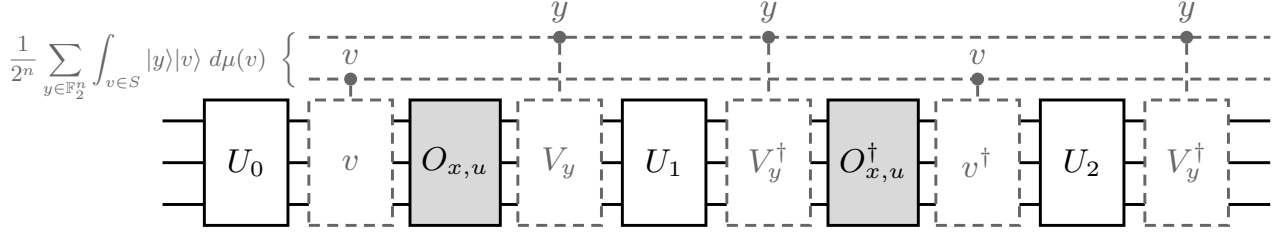
Figure 3: Symmetrized algorithm. We symmetrize the algorithm by introducing unitaries V_y and v, controlled by an extra register prepared in the uniform superposition over all $|y\rangle$ and $|v\rangle$.

The success probability of the symmetrized algorithm is

$$\sqrt{\bar{p}} := \mathrm{Re}\left[\overline{\langle\boldsymbol{\sigma}(x)|\langle\bar{0}|}\cdot\overline{|\psi_T(x,u)\rangle}\right]$$

$$= \sum_{k=1}^n \sigma_k \cdot \frac{1}{2^n}\sum_{y\in\mathbb{F}_2^n}\int_{v\in S}\mathrm{Re}\left[\langle\bar{0}|\gamma_k(x+y,uv)\rangle\,d\mu(v)\right]$$

$$= \boldsymbol{\sigma}^\mathsf{T}\cdot\bar{\boldsymbol{\gamma}},$$

where, by changing variables, we get that $\bar{\boldsymbol{\gamma}}$ is the average of vectors $\boldsymbol{\gamma}(y,v)$ and thus does not depend on x and u:

$$\bar{\boldsymbol{\gamma}} := \frac{1}{2^n}\sum_{y\in\mathbb{F}_2^n}\int_{v\in S}\boldsymbol{\gamma}(y,v)\,d\mu(v).$$

Note that $\|\bar{\boldsymbol{\gamma}}\|_2 \leq 1$ by triangle inequality. Also, note that $\bar{p} \geq p$, since the mean is at least as large as the minimum. Thus without loss of generality we can consider only symmetrized algorithms.

Let $x, x' \in \mathbb{F}_2^n$ and $u, u' \in S$. The difference of final states of the symmetrized algorithm that uses oracles $O_{x,u}$ and $O_{x',u'}$ is given in Eq. (10) on the next page. By the hybrid argument [8, 48], we get the following upper bound:[2]

$$\left\|\overline{|\psi_T(x,u)\rangle} - \overline{|\psi_T(x',u')\rangle}\right\|_2 \leq T\cdot\|O_{x,u}-O_{x',u'}\|_\infty, \quad (6)$$

where $\|\cdot\|_\infty$ denotes the usual operator norm. Bounds (10) and (6) together imply that for any $x, x' \in \mathbb{F}_2^n$ and $u, u' \in S$:

$$T \geq \frac{\sqrt{\sum_{k:x_k\neq x'_k}(2\bar{\gamma}_k)^2}}{\|O_{x,u}-O_{x',u'}\|_\infty}. \quad (7)$$

To obtain a good lower bound, we want to choose oracles $O_{x,u}$ and $O_{x',u'}$ to be as similar as possible. First, let us fix $u := I$, $x := 0$ and $x' := e_l$, where e_l is the l-th standard basis vector. Then, the numerator in Eq. (7) is simply $2\bar{\gamma}_l$. Let us choose u' in order to minimize the denominator. Recall that $O_{0,I}|\bar{0}\rangle_n = |\boldsymbol{\pi}\rangle$ and note that any unitary matrix that fixes $|\boldsymbol{\pi}\rangle$ can be written as $O_{0,I}u'(O_{0,I})^\dagger$ for some choice of u' fixing $|\bar{0}\rangle_n$. Since $O_{x',u'}|\bar{0}\rangle_n = V_{e_l}O_{0,I}u'|\bar{0}\rangle_n = V_{e_l}|\boldsymbol{\pi}\rangle$, we also have $O_{x',u'}(O_{0,I})^\dagger|\boldsymbol{\pi}\rangle = V_{e_l}|\boldsymbol{\pi}\rangle$, and any unitary matrix that sends $|\boldsymbol{\pi}\rangle$ to $V_{e_l}|\boldsymbol{\pi}\rangle$ can be expressed

---

[2]It follows by induction from the following fact. If O and O' are unitary matrices, then for any vectors $|\psi\rangle$ and $|\psi'\rangle$ it holds that

$$\|O|\psi\rangle - O'|\psi'\rangle\|_2 = \|O|\psi\rangle - O'|\psi\rangle + O'|\psi\rangle - O'|\psi'\rangle\|_2$$
$$\leq \|(O-O')|\psi\rangle\|_2 + \|O'(|\psi\rangle-|\psi'\rangle)\|_2$$
$$\leq \|O-O'\|_\infty + \||\psi\rangle-|\psi'\rangle\|_2.$$

as $O_{x',u'}(O_{0,I})^\dagger$ for some choice of u'. Let us choose u' so that $O_{x',u'}(O_{0,I})^\dagger$ acts as a rotation in the two-dimensional subspace $\mathrm{span}\{|\boldsymbol{\pi}\rangle, V_{e_l}|\boldsymbol{\pi}\rangle\}$ and as identity on the orthogonal complement. If θ denotes the angle of this rotation, then $\cos\theta = \langle\boldsymbol{\pi}|V_{e_l}|\boldsymbol{\pi}\rangle = \sum_{k=1}^n \pi_k^2 - 2\pi_l^2 = 1 - 2\pi_l^2$ and $\sin\theta = \sqrt{1-(1-2\pi_l^2)^2} = 2\pi_l\sqrt{1-\pi_l^2}$. Then

$$\|O_{0,I}-O_{x',u'}\|_\infty = \left\|I - O_{x',u'}(O_{0,I})^\dagger\right\|_\infty$$

$$= \left\|I - \begin{pmatrix}\cos\theta & -\sin\theta \\ \sin\theta & \cos\theta\end{pmatrix}\right\|_\infty$$

$$= 2\pi_l\left\|\begin{pmatrix}\pi_l & \sqrt{1-\pi_l^2} \\ -\sqrt{1-\pi_l^2} & \pi_l\end{pmatrix}\right\|_\infty$$

$$= 2\pi_l,$$

where the singular values of the last matrix are equal to 1, since it is a rotation. By plugging this back in Eq. (7), we get that for any $l \in [n]$:

$$T \geq \frac{|\bar{\gamma}_l|}{\pi_l}.$$

Thus any quantum algorithm that solves $\mathrm{QSAMPLING}^p_{\boldsymbol{\pi}\to\boldsymbol{\sigma}}$ with T queries and success probability p gives us some vector $\bar{\boldsymbol{\gamma}}$ such that

$$\|\bar{\boldsymbol{\gamma}}\|_2 \leq 1, \qquad \boldsymbol{\sigma}^\mathsf{T}\cdot\bar{\boldsymbol{\gamma}} \geq \sqrt{p}, \qquad \forall l: |\bar{\gamma}_l| \leq T\pi_l. \quad (8)$$

To obtain a lower bound on T, we have to find the smallest possible t such that there is still a feasible value of $\bar{\boldsymbol{\gamma}}$ that satisfies Eqs. (8) (with T replaced by t). We can state this as an optimization problem:

$$T \geq \min_{\bar{\boldsymbol{\gamma}}} t \quad \text{s.t.} \quad \begin{aligned}&\|\bar{\boldsymbol{\gamma}}\|_2 \leq 1, \\ &\forall l: |\bar{\gamma}_l| \leq t\pi_l, \\ &\boldsymbol{\sigma}^\mathsf{T}\cdot\bar{\boldsymbol{\gamma}} \geq \sqrt{p}.\end{aligned} \quad (9)$$

Finally, let us show that we can start with a feasible solution $\bar{\boldsymbol{\gamma}}$ of problem (9) and modify its components, without increasing the objective value or violating any of the constraints, so that $\forall l: \bar{\gamma}_l \geq 0$ and $\|\bar{\boldsymbol{\gamma}}\|_2 = 1$. Clearly, making all components of $\bar{\boldsymbol{\gamma}}$ non-negative does not affect the objective value and makes the last constraint only easier to satisfy since $\sigma_k \geq 0$ for all k. To turn $\bar{\boldsymbol{\gamma}}$ into a unit vector, first observe that not all of the constraints $\bar{\gamma}_l \leq t\pi_l$ can be saturated (indeed, in that case we would have $\bar{\boldsymbol{\gamma}} = t\boldsymbol{\pi}$ with $t < 1$ since $\|\bar{\boldsymbol{\gamma}}\|_2 < \|\boldsymbol{\pi}\|_2 = 1$, but the last constraint then implies $\boldsymbol{\sigma}^\mathsf{T}\cdot\boldsymbol{\pi} > \sqrt{p}$, which contradicts the assumption $p \geq p_{\min}$). If $\|\bar{\boldsymbol{\gamma}}\|_2 < 1$, let j be such that $\bar{\gamma}_j < t\pi_j$. We increase $\bar{\gamma}_j$ until either $\|\bar{\boldsymbol{\gamma}}\|_2 = 1$ or $\bar{\gamma}_j = t\pi_j$. We then repeat with another j such that $\bar{\gamma}_j < t\pi_j$, until we reach $\|\bar{\boldsymbol{\gamma}}\|_2 = 1$. Note

$$\left\|\overline{|\psi_T(x,u)\rangle} - \overline{|\psi_T(x',u')\rangle}\right\|_2^2 = \left\|\sum_{k=1}^n \frac{1}{\sqrt{2^n}} \sum_{y\in\mathbb{F}_2^n} \int_{v\in S} |k\rangle\big((-1)^{x_k}|\gamma_k(x+y,uv)\rangle - (-1)^{x_k'}|\gamma_k(x'+y,u'v)\rangle\big)|y\rangle|v\rangle\, d\mu(v)\right\|_2^2$$

$$= \sum_{k=1}^n \frac{1}{2^n} \sum_{y\in\mathbb{F}_2^n} \int_{v\in S} \left\|(-1)^{x_k}|\gamma_k(x+y,uv)\rangle - (-1)^{x_k'}|\gamma_k(x'+y,u'v)\rangle\right\|_2^2\, d\mu(v)$$

$$\geq \sum_{k=1}^n \frac{1}{2^n} \sum_{y\in\mathbb{F}_2^n} \int_{v\in S} \big((-1)^{x_k}\gamma_k(x+y,uv) - (-1)^{x_k'}\gamma_k(x'+y,u'v)\big)^2\, d\mu(v)$$

$$\geq \sum_{k=1}^n \big((-1)^{x_k}\bar\gamma_k - (-1)^{x_k'}\bar\gamma_k\big)^2 = \sum_{k:x_k\neq x_k'} \big(2\bar\gamma_k\big)^2. \tag{10}$$

Here the two inequalities were obtained from the following facts:

1. If $|\bar 0\rangle$ is a unit vector then for any $|\gamma\rangle$ we have: $\||\gamma\rangle\|_2^2 \geq \||\bar 0\rangle\langle\bar 0|\gamma\rangle\|_2^2 = |\langle\bar 0|\gamma\rangle|^2 \geq \big(\mathrm{Re}[\langle\bar 0|\gamma\rangle]\big)^2$.

2. For any function $\gamma(y,v)$ by Cauchy–Schwarz inequality we have:

$$\frac{1}{2^n}\sum_{y\in\mathbb{F}_2^n}\int_{v\in S}\gamma(y,v)^2\, d\mu(v) \geq \left(\frac{1}{2^n}\sum_{y\in\mathbb{F}_2^n}\int_{v\in S}\gamma(y,v)\, d\mu(v)\right)^2.$$

---

that while doing so, the other constraints remain satisfied. Therefore, the program reduces to

$$T \geq \min_{\bar\gamma} t \quad \text{s.t.} \quad \begin{aligned} &\|\bar\gamma\|_2 = 1, \\ &\forall l: 0 \leq \bar\gamma_l \leq t\pi_l, \\ &\boldsymbol{\sigma}^\mathsf{T}\cdot\bar\gamma \geq \sqrt{p}. \end{aligned}$$

Note that we need $p \leq p_{\max}$, otherwise this program has no feasible point. Setting $\boldsymbol{\varepsilon} = \bar\gamma/t$, we may rewrite this program as in Eq. (22) in Appendix A:

$$\frac{1}{T^2} \leq \max_{\varepsilon_k\geq 0}\|\boldsymbol{\varepsilon}\|_2^2 \quad \text{s.t.} \quad \begin{aligned} &\forall k: \pi_k \geq \varepsilon_k \geq 0, \\ &\boldsymbol{\sigma}^\mathsf{T}\cdot\boldsymbol{\varepsilon} \geq \sqrt{p}\,\|\boldsymbol{\varepsilon}\|_2, \end{aligned}$$

Finally, setting $M = \boldsymbol{\varepsilon}\cdot\boldsymbol{\varepsilon}^\mathsf{T}$, this program becomes the same as the SDP in Eq. (4), with the additional constraint that M is rank-1. However, we know from Lemma 1 that the SDP in Eq. (4) admits a rank-1 optimal point, therefore adding this constraint does not modify the objective value, which is also $\|\varepsilon_{\pi\to\sigma}^p\|_2^2$ by Lemma 1. $\quad\square$

4. QUANTUM REJECTION SAMPLING ALGORITHM

In this section, we describe quantum rejection sampling algorithms for QSAMPLING$_{\pi\to\sigma}$ and SQSAMPLING$_\tau$ problems. We also explain the intuition behind our method and its relation to the classical rejection sampling. Our algorithms are based on amplitude amplification [12] and can be seen as an extension of the algorithm in [24].

4.1 Intuitive description of the algorithm

The workspace of our algorithm is $\mathbb{C}^d\otimes\mathbb{C}^n\otimes\mathbb{C}^2$, where the last register can be interpreted as a quantum coin that determines whether a sample will be rejected or accepted (this corresponds to basis states $|0\rangle$ and $|1\rangle$, respectively). Our algorithm is parametrized by a vector $\boldsymbol{\varepsilon}\in\mathbb{R}^n$ ($0\leq\varepsilon_k\leq\pi_k$ for all k) that characterizes how much of the amplitude from the initial state will be used for creating the final state (in classical rejection sampling ε_k^2 corresponds to the probability

that a specific value of k is drawn from the initial distribution and accepted).

We start in the initial state $|\bar 0\rangle_d|\bar 0\rangle_n|0\rangle$ and apply the oracle O to prepare $|\boldsymbol{\pi}^\xi\rangle$ in the first two registers:

$$O|\bar 0\rangle_{dn}\otimes|0\rangle = |\boldsymbol{\pi}^\xi\rangle|0\rangle = \sum_{k=1}^n \pi_k|\xi_k\rangle|k\rangle|0\rangle. \tag{11}$$

Next, for each k let $R_{\boldsymbol{\varepsilon}}(k)$ be a single-qubit unitary operation defined[3] as follows (this is a rotation by an angle whose sine is equal to ε_k/π_k):

$$R_{\boldsymbol{\varepsilon}}(k) := \frac{1}{\pi_k}\begin{pmatrix} \sqrt{|\pi_k|^2-\varepsilon_k^2} & -\varepsilon_k \\ \varepsilon_k & \sqrt{|\pi_k|^2-\varepsilon_k^2} \end{pmatrix}. \tag{12}$$

Let $R_{\boldsymbol{\varepsilon}} := \sum_{k=1}^n |k\rangle\langle k|\otimes R_{\boldsymbol{\varepsilon}}(k)$ be a block-diagonal matrix that performs rotations by different angles in mutually orthogonal subspaces. Then $I_d\otimes R_{\boldsymbol{\varepsilon}}$ corresponds to applying $R_{\boldsymbol{\varepsilon}}(k)$ on the last qubit, controlled on the value of the second register being equal to k. This operation transforms state (11) into

$$|\Psi_{\boldsymbol{\varepsilon}}\rangle := (I_d\otimes R_{\boldsymbol{\varepsilon}})\cdot|\boldsymbol{\pi}^\xi\rangle|0\rangle$$
$$= \sum_{k=1}^n |\xi_k\rangle|k\rangle\Big(\sqrt{|\pi_k|^2-\varepsilon_k^2}\,|0\rangle + \varepsilon_k|1\rangle\Big). \tag{13}$$

If we would measure the coin register of state $|\Psi_{\boldsymbol{\varepsilon}}\rangle$ in the basis $\{|0\rangle,|1\rangle\}$, the probability of outcome $|1\rangle$ ("accept") and the corresponding post-measurement state would be

$$q_{\boldsymbol{\varepsilon}} := \left\|\big(I_d\otimes I_n\otimes|1\rangle\langle 1|\big)|\Psi_{\boldsymbol{\varepsilon}}\rangle\right\|_2^2 = \sum_{k=1}^n \varepsilon_k^2 = \|\boldsymbol{\varepsilon}\|_2^2, \tag{14}$$

$$|\Psi_{\Pi,\boldsymbol{\varepsilon}}\rangle := \frac{1}{\|\boldsymbol{\varepsilon}\|_2}\sum_{k=1}^n \varepsilon_k|\xi_k\rangle|k\rangle|1\rangle.$$

---

[3] For those k, for which $\pi_k = 0$, operation $R_{\boldsymbol{\varepsilon}}(k)$ can be defined arbitrarily.

Note that if the vector of amplitudes $\boldsymbol{\varepsilon}$ is chosen to be close to $\boldsymbol{\sigma}$, then the reduced state on the first two registers of $|\Psi_{\Pi,\boldsymbol{\varepsilon}}\rangle$ has a large overlap on the target state $|\boldsymbol{\sigma}^{\xi}\rangle$, more precisely,

$$\sqrt{p_{\boldsymbol{\varepsilon}}} := \big(\langle\boldsymbol{\sigma}^{\xi}|\otimes\langle 1|\big)|\Psi_{\Pi,\boldsymbol{\varepsilon}}\rangle = \boldsymbol{\sigma}^{\mathsf{T}}\cdot\frac{\boldsymbol{\varepsilon}}{\|\boldsymbol{\varepsilon}\|_2}, \qquad (15)$$

Depending on the choice of $\boldsymbol{\varepsilon}$, this can be a reasonably good approximation, so our strategy will be to prepare a state close to $|\Psi_{\Pi,\boldsymbol{\varepsilon}}\rangle$.

In principle, we could obtain $|\Psi_{\Pi,\boldsymbol{\varepsilon}}\rangle$ by repeatedly preparing $|\Psi_{\boldsymbol{\varepsilon}}\rangle$ and measuring its coin register until we get the outcome "accept" (we would succeed with high probability after $O(1/q_{\boldsymbol{\varepsilon}})$ steps). To speed up this process, we can use amplitude amplification [12] to amplify the amplitude of the "accept" state $|1\rangle$ in the coin register of the state in Eq. (13). This allows us to increase the probability of outcome "accept" arbitrarily close to 1 in $O(1/\sqrt{q_{\boldsymbol{\varepsilon}}})$ steps.

4.2 Amplitude amplification subroutine and quantum rejection sampling algorithm
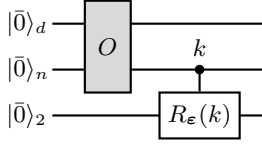

Figure 4: Quantum circuit for implementing $U_{\boldsymbol{\varepsilon}}$.

We will use the following amplitude amplification subroutine extensively in all algorithms presented in this paper:

$$\mathcal{S}_{\mathrm{QRS}}(\mathrm{ref}_{|\boldsymbol{\pi}^{\xi}\rangle|0\rangle},\boldsymbol{\varepsilon},t) := \big(\mathrm{ref}_{|\Psi_{\boldsymbol{\varepsilon}}\rangle}\cdot\mathrm{ref}_{I_d\otimes I_n\otimes|1\rangle\langle 1|}\big)^t,$$

where reflections through the two subspaces are defined as follows:

$$\mathrm{ref}_{I_d\otimes I_n\otimes|1\rangle\langle 1|} := I_d\otimes I_n\otimes\big(I_2-2|1\rangle\langle 1|\big) = I_d\otimes I_n\otimes Z,$$

$$\mathrm{ref}_{|\Psi_{\boldsymbol{\varepsilon}}\rangle} := (I_d\otimes R_{\boldsymbol{\varepsilon}})\,\mathrm{ref}_{|\boldsymbol{\pi}^{\xi}\rangle|0\rangle}\,(I_d\otimes R_{\boldsymbol{\varepsilon}})^{\dagger}.$$

Depending on the application, we will either be given an oracle O for preparing $|\boldsymbol{\pi}^{\xi}\rangle|0\rangle$ or an oracle $\mathrm{ref}_{|\boldsymbol{\pi}^{\xi}\rangle|0\rangle}$ for reflecting through this state. Note that we can always use the preparation oracle to implement the reflection $\mathrm{ref}_{|\boldsymbol{\pi}^{\xi}\rangle|0\rangle}$ as

$$(O\otimes I_2)\,\big(I_d\otimes I_n\otimes I_2-2|\bar{0},\bar{0},0\rangle\langle\bar{0},\bar{0},0|\big)\,(O\otimes I_2)^{\dagger}. \quad (16)$$

Amplitude amplification subroutine $\mathcal{S}_{\mathrm{QRS}}(\mathrm{ref}_{|\boldsymbol{\pi}^{\xi}\rangle|0\rangle},\boldsymbol{\varepsilon},t)$ for quantum rejection sampling
Perform the following steps t times:
1. Perform $\mathrm{ref}_{I_d\otimes I_n\otimes|1\rangle\langle 1|}$ by applying Pauli Z on the coin register.
2. Perform $\mathrm{ref}_{|\Psi_{\boldsymbol{\varepsilon}}\rangle}$ by applying $R_{\boldsymbol{\varepsilon}}^{\dagger}$ on the last two registers, applying $\mathrm{ref}_{|\boldsymbol{\pi}^{\xi}\rangle|0\rangle}$, and then undoing $R_{\boldsymbol{\varepsilon}}$.

The quantum rejection sampling algorithm $\mathcal{A}_{\mathrm{QRS}}(O,\boldsymbol{\pi},\boldsymbol{\varepsilon})$ starts with the initial state $|\bar{0}\rangle_d|\bar{0}\rangle_n|0\rangle$. First, we transform it into $|\Psi_{\boldsymbol{\varepsilon}}\rangle$ defined in Eq. (13), by applying $U_{\boldsymbol{\varepsilon}} := (I_d\otimes R_{\boldsymbol{\varepsilon}})\cdot(O\otimes I_2)$ (see Fig. 4). Then we apply the amplitude amplification subroutine $\mathcal{S}_{\mathrm{QRS}}(\mathrm{ref}_{|\boldsymbol{\pi}^{\xi}\rangle|0\rangle},\boldsymbol{\varepsilon},t)$ with $t=O(1/\|\boldsymbol{\varepsilon}\|_2)$. Finally, we measure the last register: if

the outcome is $|1\rangle$, we output the first two registers, otherwise we output "Fail". To prevent the outcome "Fail" we can slightly adjust the angle of rotation in amplitude amplification so that the target state is reached exactly. More precisely, we prove that one can choose $\boldsymbol{\varepsilon}=r\cdot\boldsymbol{\varepsilon}_{\boldsymbol{\pi}\to\boldsymbol{\sigma}}^{p}$ for some bounded constant r, so that amplitude amplification succeeds with probability 1 (*i.e.*, the outcome of the final measurement is always $|1\rangle$). Moreover, such algorithm is optimal as its cost matches the lower bound in Lemma 2.

Quantum rejection sampling algorithm $\mathcal{A}_{\mathrm{QRS}}(O,\boldsymbol{\pi},\boldsymbol{\varepsilon})$
1. Start in initial state $|\bar{0}\rangle_d|\bar{0}\rangle_n|0\rangle$.
2. Apply $U_{\boldsymbol{\varepsilon}}$.
3. On the current state apply the amplitude amplification subroutine $\mathcal{S}_{\mathrm{QRS}}(\mathrm{ref}_{|\boldsymbol{\pi}^{\xi}\rangle|0\rangle},\boldsymbol{\varepsilon},t)$ where $\mathrm{ref}_{|\boldsymbol{\pi}^{\xi}\rangle|0\rangle}$ is implemented according to Eq. (16) and $t=O(1/\|\boldsymbol{\varepsilon}\|_2)$.
4. Measure the last register. If the outcome is $|1\rangle$, output the first two registers, otherwise output "Fail".

LEMMA 3. *For any $p_{\min}\le p\le p_{\max}$, there is a constant $r\in[\frac{1}{2},1]$, so that the algorithm $\mathcal{A}_{QRS}(O,\boldsymbol{\pi},\boldsymbol{\varepsilon})$ with $\boldsymbol{\varepsilon}=r\cdot\boldsymbol{\varepsilon}_{\boldsymbol{\pi}\to\boldsymbol{\sigma}}^{p}$ solves $\mathrm{QSAMPLING}_{\boldsymbol{\pi}\to\boldsymbol{\sigma}}$ with success probability p using $O\big(1/\|\boldsymbol{\varepsilon}_{\boldsymbol{\pi}\to\boldsymbol{\sigma}}^{p}\|_2\big)$ queries to O and $O^{\dagger}$.*

PROOF. By Def. 5, we have $0\le\varepsilon_k\le\pi_k$ for all k, therefore $\boldsymbol{\varepsilon}_{\boldsymbol{\pi}\to\boldsymbol{\sigma}}^{p}$ is a valid choice of vector $\boldsymbol{\varepsilon}$ for the algorithm. Instead of using $\boldsymbol{\varepsilon}_{\boldsymbol{\pi}\to\boldsymbol{\sigma}}^{p}$ itself, we slightly scale it down by a factor r so that the amplitude amplification never fails. Note that if we were to use $\boldsymbol{\varepsilon}=\boldsymbol{\varepsilon}_{\boldsymbol{\pi}\to\boldsymbol{\sigma}}^{p}$, the probability that the amplitude amplification succeeds after t steps would be $\sin^2\big((2t+1)\theta\big)$, where $\theta:=\arcsin\|\boldsymbol{\varepsilon}_{\boldsymbol{\pi}\to\boldsymbol{\sigma}}^{p}\|_2$ (see *e. g.* [11, 12] for details). Note that this probability would be equal to one at $t=\frac{\pi}{4\theta}-\frac{1}{2}$, which in general might not be an integer. However, following an idea from [12, p.10], we can ensure this by slightly decreasing θ to $\tilde{\theta}:=\frac{\pi}{2(2\tilde{t}+1)}$ where $\tilde{t}:=\lceil\frac{\pi}{4\theta}-\frac{1}{2}\rceil$. This can be done by setting $\boldsymbol{\varepsilon}:=r\cdot\boldsymbol{\varepsilon}_{\boldsymbol{\pi}\to\boldsymbol{\sigma}}^{p}$ with the scaling-down factor $r:=\frac{\sin\tilde{\theta}}{\sin\theta}$. One can check that r satisfies $\frac{1}{2}\le r\le 1$ (this follows from $0\le\theta\le\frac{\pi}{2}$).

Together with Eq. (15), Def. 5 also implies that for this choice, the algorithm solves $\mathrm{QSAMPLING}_{\boldsymbol{\pi}\to\boldsymbol{\sigma}}$ with success probability $\frac{\boldsymbol{\sigma}^{\mathsf{T}}\cdot\boldsymbol{\varepsilon}}{\|\boldsymbol{\varepsilon}\|_2}=\frac{\boldsymbol{\sigma}^{\mathsf{T}}\cdot\boldsymbol{\varepsilon}_{\boldsymbol{\pi}\to\boldsymbol{\sigma}}^{p}}{\|\boldsymbol{\varepsilon}_{\boldsymbol{\pi}\to\boldsymbol{\sigma}}^{p}\|_2}=\sqrt{p}$. It therefore remains to prove that the cost of the algorithm is $O(1/\|\boldsymbol{\varepsilon}\|_2)$, which follows immediately from its description: we need one query to implement the operation $U_{\boldsymbol{\varepsilon}}$ and two queries to implement $\mathrm{ref}_{|\boldsymbol{\pi}^{\xi}\rangle|0\rangle}$, thus in total we need $2t+1=O\big(1/\sqrt{q_{\boldsymbol{\varepsilon}}}\big)=O(1/\|\boldsymbol{\varepsilon}\|_2)$ calls to oracles O and $O^{\dagger}$. $\square$

We now have all the elements to prove Theorem 1.

THEOREM 1. *For $p\in[p_{\min},p_{\max}]$, the quantum query complexity of $\mathrm{QSAMPLING}_{\boldsymbol{\pi}\to\boldsymbol{\sigma}}$ with success probability p is $Q_{1-p}(\mathrm{QSAMPLING}_{\boldsymbol{\pi}\to\boldsymbol{\sigma}})=\Theta(1/\|\boldsymbol{\varepsilon}_{\boldsymbol{\pi}\to\boldsymbol{\sigma}}^{p}\|_2)$. For $p\le p_{\min}$, the query complexity is 1, and for $p>p_{\max}$, it is infinite.*

PROOF. When $p\le p_{\min}$, the state $|\boldsymbol{\pi}^{\xi}\rangle$ is already close enough to $|\boldsymbol{\sigma}^{\xi}\rangle$ to satisfy the constraint on the success probability, therefore one call to O is sufficient, which is clearly optimal. When $p>p_{\max}$, the oracle gives no information about some of the unknown states $|\xi_k\rangle$ (when $\pi_k=0$), but the target state should have some overlap on $|\xi_k\rangle|k\rangle$ to satisfy the constraint on the success probability, therefore the problem is not solvable.

For the general case $p_{\min} \le p \le p_{\max}$, the upper bound comes from the algorithm in Lemma 3, and the matching lower bound is given in Lemma 2. $\quad\square$

4.3 Strong quantum rejection sampling algorithm

Let us now describe how the algorithm can be modified to solve the stronger problem $\mathrm{SQSAMPLING}_{\boldsymbol\tau}$. The first modification follows from the observation that in the previous algorithm, the oracle is only used in two different ways: it is used once to create the state $|\boldsymbol\pi^\xi\rangle$, and then only to reflect through that state. This means that we can still solve the problem if, instead of being given access to an oracle that maps $|\bar 0\rangle_{dn}$ to $|\boldsymbol\pi^\xi\rangle$, we are provided with one copy of $|\boldsymbol\pi^\xi\rangle$ and an oracle that reflects through it.

In order to solve $\mathrm{SQSAMPLING}_{\boldsymbol\tau}$, we should also be able to deal with the case where we only know the ratios between the amplitudes π_k and σ_k for each k, instead of the amplitudes themselves. As we will see, in that case we cannot use the algorithm given above anymore, as we do not know in advance how many steps of amplitude amplification are required. There are different approaches to solve this issue, one of them being to estimate $q_{\boldsymbol\varepsilon}$, and therefore the required number of steps, by performing a phase estimation on the amplitude amplification operator (this is sometimes referred to as amplitude estimation or quantum counting, see [11, 12]). Another option, also proposed by [11, 12], is to repeat the algorithm successively with an increasing number of steps until it succeeds. One advantage of the first option would be that it provides an estimation of the initial acceptance probability $q_{\boldsymbol\varepsilon}$, which might be useful for some applications. Since this is not required for $\mathrm{SQSAMPLING}_{\boldsymbol\tau}$, we will rather describe an algorithm based on the second option, as it is more direct. Note that for both options, we need to adapt the algorithms in [11, 12] as they require to use a fresh copy of the initial state after each failed attempt, whereas for $\mathrm{SQSAMPLING}_{\boldsymbol\tau}$ we only have one copy of that state. This issue can be solved by using the state left over from the previous unsuccessful measurement instead of a fresh copy of the state. More precisely, we can use the following algorithm.

Strong quantum rejection sampling algorithm
$\mathcal{A}_{\mathrm{SQRS}}(|\boldsymbol\pi^\xi\rangle, \boldsymbol\varepsilon, c)$
1. Append an extra qubit prepared in the state $|0\rangle$ to the input state $|\boldsymbol\pi^\xi\rangle$, and apply $I_d \otimes R_{\boldsymbol\varepsilon}$ on the resulting state.
2. Measure the last register of the current state. If the outcome is $|1\rangle$, output the first two registers and stop. Otherwise, set $l = 0$ and continue.
3. Let $T_l := \lceil c^l \rceil$ and pick an integer $t \in [T_l]$ uniformly at random.
4. On the current state apply the amplitude amplification subroutine $\mathcal{S}_{\mathrm{QRS}}(\mathrm{ref}_{|\boldsymbol\pi^\xi\rangle|0\rangle}, \boldsymbol\varepsilon, t)$.
5. Measure the last register of the current state. If the outcome is $|1\rangle$, output the first two registers and stop. Otherwise, increase l by one and go back to Step 3.

LEMMA 4. *For any $\alpha \ge 1$, there is a quantum algorithm that solves* $\mathrm{SQSAMPLING}_{\boldsymbol\tau}$ *with success probability $p(\gamma)$ using an expected number of queries $O(1/\|\boldsymbol\varepsilon(\gamma)\|_2)$, where $\gamma = \alpha \|\boldsymbol\pi \circ \boldsymbol\tau\|_2$. In particular, for $\alpha = 1$ the expected number of queries is $O(1/\|\boldsymbol\pi \circ \boldsymbol\tau\|_2)$ and the success probability is equal to one.*

Here the parameter α allows us to control the trade-off between the success probability and the required number of queries. However, we cannot predict the actual values of both quantities, because they depend on $\boldsymbol\pi$ and $\boldsymbol\tau$, but only $\boldsymbol\tau$ is known to us. The only exception is $\alpha = 1$, when the success probability is equal to one. Also, increasing α above $1/(\min_{k:\tau_k>0} \tau_k)$ will no longer affect the query complexity and success probability of the algorithm.

PROOF. We show that for some choice of $c > 1$ and $\boldsymbol\varepsilon$ the algorithm $\mathcal{A}_{\mathrm{SQRS}}(|\boldsymbol\pi^\xi\rangle, \boldsymbol\varepsilon, c)$ described above solves the problem.

Let us first verify that we can actually perform all steps required in the algorithm. We need one copy of the state $|\boldsymbol\pi^\xi\rangle$, which is indeed provided as an input for the $\mathrm{SQSAMPLING}_{\boldsymbol\tau}$ problem. Note that for applying $R_{\boldsymbol\varepsilon}$ in Step 1 it suffices to know only the ratio ε_k/π_k (see Eq. (12)), which can be deduced from τ_k as follows. Let $\boldsymbol\varepsilon := r \cdot \boldsymbol\varepsilon(\gamma)$ for some $r < 1$, and recall from Def. 5 and 4 that $\varepsilon_k(\gamma) = \min\{\pi_k, \gamma\sigma_k\}$ and $\sigma_k = \pi_k\tau_k/\|\boldsymbol\pi \circ \boldsymbol\tau\|_2$, respectively. Then

$$
\begin{aligned}
\frac{\varepsilon_k}{\pi_k} &= r\min\{1, \gamma\frac{\sigma_k}{\pi_k}\} \\
&= r\min\{1, \gamma\frac{\tau_k}{\|\boldsymbol\pi \circ \boldsymbol\tau\|_2}\} \\
&= r\min\{1, \alpha\tau_k\},
\end{aligned}
\tag{17}
$$

where we substituted $\gamma = \alpha\|\boldsymbol\pi \circ \boldsymbol\tau\|_2$ from the statement of the Lemma. Note that once r is chosen, the final expression in Eq. (17) is completely known. Finally, applying $\mathcal{S}_{\mathrm{QRS}}(\mathrm{ref}_{|\boldsymbol\pi^\xi\rangle|0\rangle}, \boldsymbol\varepsilon, t)$ in Step 4 also requires the ability to apply $R_{\boldsymbol\varepsilon}$, as well as $\mathrm{ref}_{|\boldsymbol\pi^\xi\rangle|0\rangle}$, which can be done by using one oracle query. Therefore, we have all we need to implement the algorithm.

We now show that the algorithm has success probability $p(\gamma)$. Recall from Eq. (13) that Step 1 of the algorithm prepares the state

$$
\begin{aligned}
|\Psi_{\boldsymbol\varepsilon}\rangle &= \sum_{k=1}^n |\xi_k\rangle|k\rangle\left(\sqrt{|\pi_k|^2 - \varepsilon_k^2}\,|0\rangle + \varepsilon_k|1\rangle\right) \\
&= \sin\theta|\Psi_{\Pi,\boldsymbol\varepsilon}\rangle + \cos\theta|\Psi_{\Pi,\boldsymbol\varepsilon}^\perp\rangle,
\end{aligned}
$$

where $\theta := \arcsin\|\boldsymbol\varepsilon\|_2$ and unit vectors

$$
|\Psi_{\Pi,\boldsymbol\varepsilon}\rangle := \frac{1}{\sin\theta}\sum_{k=1}^n \varepsilon_k|\xi_k\rangle|k\rangle|1\rangle,
$$

$$
|\Psi_{\Pi,\boldsymbol\varepsilon}^\perp\rangle := \frac{1}{\cos\theta}\sum_{k=1}^n \sqrt{|\pi_k|^2 - \varepsilon_k^2}|\xi_k\rangle|k\rangle|0\rangle
$$

are orthogonal and span a 2-dimensional subspace. In this subspace $\mathrm{ref}_{I_d \otimes I_n \otimes |1\rangle\langle 1|}$ and $\mathrm{ref}_{|\Psi_{\Pi,\boldsymbol\varepsilon}\rangle}$ act in the same way, so each iteration of the amplitude amplification subroutine consists of a product of two reflections that preserve this subspace, and $\mathcal{S}_{\mathrm{QRS}}(\mathrm{ref}_{|\boldsymbol\pi^\xi\rangle|0\rangle}, \boldsymbol\varepsilon, t)$ corresponds to a rotation by angle $2t\theta$. Measurements in Steps 2 and 5 either project on $|\Psi_{\Pi,\boldsymbol\varepsilon}\rangle$, when the outcome is $|1\rangle$, or on $|\Psi_{\Pi,\boldsymbol\varepsilon}^\perp\rangle$, when the outcome is $|0\rangle$. Therefore, the algorithm always outputs the first two registers of the state $|\Psi_{\Pi,\boldsymbol\varepsilon}\rangle$, and by Eq. (15), the success probability is $p_{\boldsymbol\varepsilon} = p(\gamma)$, as claimed. In particular, for $\alpha = 1$ from Eq. (17) we get $\varepsilon_k = r\pi_k\tau_k$ as $\tau_k \le 1$. Thus, $\boldsymbol\varepsilon = r(\boldsymbol\pi \circ \boldsymbol\tau)$ and since $\boldsymbol\sigma = (\boldsymbol\pi \circ \boldsymbol\tau)/\|\boldsymbol\pi \circ \boldsymbol\tau\|_2$, we get $p_{\boldsymbol\varepsilon} = (\boldsymbol\sigma^{\mathsf T} \cdot \boldsymbol\varepsilon/\|\boldsymbol\varepsilon\|_2)^2 = 1$.

Let us now bound the expected number of oracle queries. We follow the proof of Theorem 3 in [12], but there is an im-

portant difference: a direct analogue of the algorithm in [12, Theorem 3] would use a fresh copy of $|\boldsymbol{\pi}^\xi\rangle$ each time the measurement fails to give a successful outcome, whereas in this algorithm we start from the state left over from the previous measurement, since we only have one copy of $|\boldsymbol{\pi}^\xi\rangle$. Note that $\mathcal{S}_{\text{QRS}}(\text{ref}_{|\boldsymbol{\pi}^\xi\rangle|0\rangle}, \boldsymbol{\varepsilon}, t)$ in Step 4 is always applied on $|\Psi_{\Pi,\boldsymbol{\varepsilon}}^\perp\rangle$, since it is the post-measurement state corresponding to the unsuccessful outcome. Therefore, the state created by Step 4 is $\sin(2t\theta)|\Psi_{\Pi,\boldsymbol{\varepsilon}}\rangle + \cos(2t\theta)|\Psi_{\Pi,\boldsymbol{\varepsilon}}^\perp\rangle$, and the next measurement will succeed with probability $\sin^2(2t\theta)$. Since t is picked uniformly at random between 1 and T_l, the probability that the l-th measurement fails is

$$
\begin{aligned}
p_l &= \frac{1}{T_l}\sum_{t=1}^{T_l}\cos^2(2t\theta) \\
&= \frac{1}{2} + \frac{1}{2T_l}\sum_{t=1}^{T_l}\cos(4t\theta) \\
&\leq \frac{1}{2} + \frac{1}{2T_l\,\|\boldsymbol{\varepsilon}\|_2},
\end{aligned}
\tag{18}
$$

where the upper bound is obtained as follows:

$$
\begin{aligned}
\sum_{t=1}^{T}\cos(4t\theta) &= \operatorname{Re}\left(e^{i4\theta}\sum_{t=0}^{T-1}e^{i4t\theta}\right) \\
&= \operatorname{Re}\left(e^{i4\theta}\cdot\frac{1-e^{i4T\theta}}{1-e^{i4\theta}}\right) \\
&= \operatorname{Re}\left(e^{i2(T+1)\theta}\cdot\frac{e^{-i2T\theta}-e^{i2T\theta}}{e^{-i2\theta}-e^{i2\theta}}\right) \\
&= \cos\big(2(T+1)\theta\big)\frac{\sin(2T\theta)}{\sin(2\theta)} \\
&\leq \frac{1}{\sin(2\theta)} \leq \frac{1}{\sin\theta} = \frac{1}{\|\boldsymbol{\varepsilon}\|_2},
\end{aligned}
$$

where we forced the last inequality by picking $r := \sqrt{3}/2$, so that $\sin\theta = \|\boldsymbol{\varepsilon}\|_2 \leq \sqrt{3}/2$ and thus $0 \leq \theta \leq \pi/3$. Recall from the algorithm that $T_l = \lceil c^l\rceil$ for some $c > 1$, so it is increasing and goes to infinity as l increases. Let $\bar{T} := 1/(2\Delta\,\|\boldsymbol{\varepsilon}\|_2)$ for some $\Delta > 0$ and let $\bar{l}$ be the smallest integer such that $T_l \geq \bar{T}$ for all $l \geq \bar{l}$. Then according to Eq. (18) we get that $p_l \leq 1/2 + 1/(2\bar{T}\,\|\boldsymbol{\varepsilon}\|_2) = 1/2 + \Delta =: \bar{p}$ for all $l \geq \bar{l}$. Note that the l-th execution of the subroutine uses at most $2T_l$ oracle queries, so the expected number of oracle calls is at most $2T_0 + p_0(2T_1 + p_1(2T_2 + \ldots))$. This can be upper bounded by

$$
\begin{aligned}
\sum_{l=0}^{\bar{l}}2T_l + \sum_{d=1}^{\infty}2T_{\bar{l}+d}\bar{p}^d &= \sum_{l=0}^{\bar{l}}2\lceil c^l\rceil + \sum_{d=1}^{\infty}2\lceil c^{\bar{l}+d}\rceil\bar{p}^d \\
&\leq 4\left(\sum_{l=0}^{\bar{l}}c^l + c^{\bar{l}}\sum_{d=1}^{\infty}(c\bar{p})^d\right) \tag{19} \\
&= 4\left(\frac{c^{\bar{l}+1}-1}{c-1} + c^{\bar{l}}\frac{c\bar{p}}{1-c\bar{p}}\right) \\
&\leq 4c^{\bar{l}+1}\left(\frac{1}{c-1} + \frac{\bar{p}}{1-c\bar{p}}\right) \\
&\leq \frac{2c^2}{\Delta\,\|\boldsymbol{\varepsilon}\|_2}\left(\frac{1}{c-1} + \frac{\bar{p}}{1-c\bar{p}}\right), \tag{20}
\end{aligned}
$$

where the first and last inequality is obtained from the following two observations, respectively:

1. $\lceil c^l\rceil = c^l + \delta$ for some $0 \leq \delta < 1$, so $T_l = \lceil c^l\rceil < c^l + 1 < 2c^l$ as $c > 1$.

2. $c^{\bar{l}+1} \leq c^2\lceil c^{\bar{l}-1}\rceil = c^2 T_{\bar{l}-1} < c^2\bar{T} = c^2/(2\Delta\,\|\boldsymbol{\varepsilon}\|_2)$ by the choice of $\bar{l}$.

Finally, we have to make a choice of $c > 1$ and $\Delta > 0$, so that the geometric series in Eq. (19) converges, *i. e.*, $c\bar{p} < 1$ or equivalently $c < 2/(1+2\Delta)$. By choosing $c := 8/7$ and $\Delta := 1/4$ we minimize the upper bound in Eq. (20) and obtain $128/\|\boldsymbol{\varepsilon}\|_2 = O(1/\|\boldsymbol{\varepsilon}(\gamma)\|_2)$. In particular, for $\alpha = 1$ this becomes $O(1/\|\boldsymbol{\pi}\circ\boldsymbol{\tau}\|_2)$. $\square$

5. APPLICATIONS

5.1 Linear systems of equations

As a first example of application, we show that quantum rejection sampling was implicitly used in the quantum algorithm for linear systems of equations proposed by Harrow, Hassidim, and Lloyd [26]. This algorithm solves the following quantum state generation problem: given the classical description of an invertible $d \times d$ matrix A and a unit vector $|b\rangle \in \mathbb{C}^d$, prepare the quantum state $|x\rangle/\|\,|x\rangle\|_2$, where $|x\rangle$ is the solution of the linear system of equations $A|x\rangle = |b\rangle$. As shown in [26], we can assume without loss of generality that A is Hermitian. Similarly to classical matrix inversion algorithms, an important factor of the performance of the algorithm is the condition number κ of A, which is the ratio between the largest and smallest eigenvalue of A. We will assume that all eigenvalues of A are between κ^{-1} and 1, and we denote by $|\psi_j\rangle$ and λ_j the eigenvectors and eigenvalues of A, respectively. We also define[4] the amplitudes $b_j := \langle\psi_j|b\rangle$, so that $|b\rangle = \sum_{j=1}^d b_j|\psi_j\rangle$. Then, the problem is to prepare the state $|x\rangle := A^{-1}|b\rangle = \sum_{j=1}^d b_j\lambda_j^{-1}|\psi_j\rangle$ (up to normalization).

We now show how this problem reduces to the quantum state conversion problem SQSAMPLING$_\tau$. Since A is Hermitian, we can use Hamiltonian simulation techniques [9, 15, 16] to simulate the unitary operator e^{iAt} on any state. Using quantum phase estimation [31, 17] on the operator e^{iAt}, we can implement an operator E_A that acts in the eigenbasis of A as $E_A : |\psi_j\rangle|\bar{0}\rangle \mapsto |\psi_j\rangle|\lambda_j\rangle$, where $|\lambda_j\rangle$ is a quantum state encoding an approximation of the eigenvalue λ_j. Here, we will assume that this can be done exactly, that is, we assume that $|\lambda_j\rangle$ is a computational basis state that exactly encodes λ_j (we refer the reader to [26] for a detailed analysis of the error introduced by this approximation). Under this assumption, the problem reduces to a quantum state conversion problem that we will call QLINEAREQUATIONS$_\kappa$. Its definition requires fixing a set of possible eigenvalues $\Lambda_\kappa \subset [\kappa^{-1}, 1]$ of finite cardinality $n := |\Lambda_\kappa|$. Let us denote the set of $d \times d$ Hermitian matrices by $\operatorname{Herm}(d)$ and the eigenvalues of A by $\operatorname{spec}(A)$.

Definition 7. QLINEAREQUATIONS$_\kappa$, the *quantum linear system of equations problem* is a quantum state conversion problem $(\mathcal{O}, \Phi, \Psi, \mathcal{X})$ with $\mathcal{X} := \{(|b\rangle, A) \in \mathbb{C}^d \times \operatorname{Herm}(d) : \operatorname{spec}(A) \in \Lambda_\kappa^d\}$, oracles in $\mathcal{O}$ being pairs $(O_{|b\rangle, A}, E_A)$ where

---

[4]We choose the global phase of each eigenvector $|\psi_j\rangle$ so that b_j is real and non-negative.

SQSAMPLING$_\tau$	QLINEAREQUATIONS$_\kappa$
$|k\rangle$	$|\lambda\rangle$
π_k	$\pi_\lambda := \begin{cases} b_j \text{ if } \lambda = \lambda_j \in \mathrm{spec}(A) \\ 0 \text{ if } \lambda \notin \mathrm{spec}(A) \end{cases}$
$|\xi_k\rangle$	$|\xi_\lambda\rangle := \begin{cases} |\psi_j\rangle \text{ if } \lambda = \lambda_j \in \mathrm{spec}(A) \\ \text{n/a if } \lambda \notin \mathrm{spec}(A) \end{cases}$
τ_k	$\tau_\lambda := (\kappa\lambda)^{-1}$

Table 1: Reduction from Step 2 in the linear system of equations algorithm to the quantum resampling problem SQSAMPLING$_\tau$**.**

$O_{|b\rangle,A} := \mathrm{ref}_{|b\rangle}$ and E_A acts as $E_A : |\psi_j\rangle|\bar{0}\rangle_n \mapsto |\psi_j\rangle|\lambda_j\rangle$ where $|\psi_j\rangle$ are the eigenvectors of A, and the corresponding initial and target states being $|b\rangle$ and $A^{-1}|b\rangle / \| A^{-1}|b\rangle \|_2$.

Using Lemma 4 we can prove the following result.

THEOREM 2. *For any $\tilde{\kappa} \in [1, \kappa]$, there is a quantum algorithm that solves* QLINEAREQUATIONS$_\kappa$ *with success probability $p = (\boldsymbol{w}^\mathsf{T} \cdot \tilde{\boldsymbol{w}}) / (\|\boldsymbol{w}\|_2 \cdot \|\tilde{\boldsymbol{w}}\|_2)$ using an expected number of queries $O(\tilde{\kappa}/ \|\tilde{\boldsymbol{w}}\|_2)$, where $w_j := b_j/\lambda_j$, $\tilde{w}_j := b_j/\tilde{\lambda}_j$, and $\tilde{\lambda}_j := \max\{\tilde{\kappa}^{-1}, \lambda_j\}$.*

PROOF. Following [26], the algorithm for this problem consists of three steps:

1. Apply the phase estimation operation E_A on $|b\rangle$ to obtain the state $\sum_{j=1}^d b_j |\psi_j\rangle|\lambda_j\rangle$.

2. Convert this state to $\sum_{j=1}^d w_j |\psi_j\rangle|\lambda_j\rangle / \|\boldsymbol{w}\|_2$.

3. Undo the phase estimation operation E_A to obtain the target state $\sum_{j=1}^d w_j |\psi_j\rangle / \|\boldsymbol{w}\|_2$.

We see that Step 2 is an instance of SQSAMPLING$_\tau$, where the basis states $\{|\lambda\rangle : \lambda \in \Lambda_\kappa\}$ of the phase estimation register play the role of the states $|k\rangle$ and the vector $\boldsymbol{\tau}$ of the ratios between the initial and final amplitudes is given by $\tau_\lambda := (\kappa\lambda)^{-1}$ (here the normalization factor κ is to make sure that $\max_\lambda \tau_\lambda = 1$). The rest of the reduction is summarized in Table 1. Therefore, we can use the algorithm from Lemma 4 to perform Step 2.

If we set $\alpha := \kappa/\tilde{\kappa}$ then from Table 1 we get

$$\begin{aligned} \varepsilon_{\lambda_j}(\gamma) &= \pi_{\lambda_j} \min\{1, \alpha\tau_{\lambda_j}\} \\ &= b_j \min\{1, (\tilde{\kappa}\lambda_j)^{-1}\} \\ &= \frac{b_j}{\tilde{\kappa} \max\{\tilde{\kappa}^{-1}, \lambda_j\}} \\ &= \frac{\tilde{w}_j}{\tilde{\kappa}}, \end{aligned}$$

thus $\boldsymbol{\varepsilon}(\gamma) = \tilde{\boldsymbol{w}}/\tilde{\kappa}$ and the expected number of queries is $O(1/\|\boldsymbol{\varepsilon}(\gamma)\|_2) = O(\tilde{\kappa}/\|\tilde{\boldsymbol{w}}\|_2)$. Recall that the amplitudes of the target state are given by $\sigma_j = w_j/\|\boldsymbol{w}\|_2$, so $\boldsymbol{\sigma} = \boldsymbol{w}/\|\boldsymbol{w}\|_2$ and the success probability is

$$p = \left(\frac{\boldsymbol{\sigma}^\mathsf{T} \cdot \boldsymbol{\varepsilon}(\gamma)}{\|\boldsymbol{\varepsilon}(\gamma)\|_2}\right)^2 = \frac{\boldsymbol{w}^\mathsf{T}}{\|\boldsymbol{w}\|_2} \cdot \frac{\tilde{\boldsymbol{w}}}{\|\tilde{\boldsymbol{w}}\|_2}$$

as claimed. □

Note that even though we have a freedom to choose $\tilde{\kappa}$, we cannot predict the query complexity in advance, since it depends on $\tilde{w}_j = \langle\psi_j|b\rangle/\tilde{\lambda}_j$, which in turn is determined by the lengths of projections of $|b\rangle$ in the eigenspaces of A, weighted by the corresponding truncated eigenvalues $\tilde{\lambda}_j$. Similarly, we cannot predict the success probability p. However, by choosing $\tilde{\kappa} = \kappa$ we can at least make sure that $p = 1$ (since $\tilde{\lambda}_j = \lambda_j$ and thus $\tilde{\boldsymbol{w}} = \boldsymbol{w}$). In this case Step 2 is performed exactly (assuming an ideal phase estimation black box) and the expected number of queries is $O(\kappa/\|\boldsymbol{w}\|_2)$. By noting that $\lambda_j \leq 1$ for all j, we see that $\|\boldsymbol{w}\|_2^2 = \sum_{j=1}^d b_j^2 \lambda_j^{-2} \geq 1$ and thus we can put a cruder upper bound of $O(\kappa)$, which coincides with the bound given in [26] for that step of the algorithm. For ill-conditioned matrices, *i. e.*, matrices with a high condition number κ, the approach taken by [26] is to ignore small eigenvalues $\lambda_j \leq \tilde{\kappa}^{-1}$, for some cut-off $\tilde{\kappa}^{-1} \geq \kappa^{-1}$, which reduces the cost of the algorithm to $O(\tilde{\kappa})$, but introduces some extra error. In our case, by choosing $\alpha = \kappa/\tilde{\kappa}$ we obtained bound $O(\tilde{\kappa}/\|\tilde{\boldsymbol{w}}\|_2)$, where $\|\tilde{\boldsymbol{w}}\|_2 \geq 1$. Again, here $\tilde{\boldsymbol{w}}$ depends on additional structure of the problem and cannot be predicted beforehand.

In practical applications, we will of course not be given access to the ideal phase estimation operator E_A, but we can still approximate it by using the phase estimation algorithm [31, 17] on the operator A. It is shown in [26] that if A is s-sparse, this approximation can be implemented with sufficient accuracy at a cost $\tilde{O}(\log(d)s^2\tilde{\kappa}/\epsilon)$, where ϵ is the overall additive error introduced by this approximation throughout the algorithm. Therefore, the total cost of the algorithm is at most $\tilde{O}(\log(d)s^2\tilde{\kappa}^2/\epsilon)$ (see [26] for details).

5.2 Quantum Metropolis sampling

Since rejection sampling lies at the core of the (classical) Metropolis algorithm, it seems natural to use quantum rejection sampling to solve the corresponding problem in the quantum case. The quantum Metropolis sampling algorithm presented in [46] follows the same lines as the classical algorithm by setting up a (classical) random walk between eigenstates of the Hamiltonian, where each move is either accepted or rejected depending on the value of some random coin. The main complication compared to the classical version comes from the case where the move has to be rejected, since we cannot keep a copy of the previous eigenstate due to the no-cloning theorem. The solution proposed by Temme *et al.* [46] is to use an unwinding technique based on successive measurements to revert to the original state. Here, we show that quantum rejection sampling can be used to avoid this step, as it allows to amplify the amplitude of the "accept" state of the coin register, effectively eliminating rejected moves. This yields a more efficient algorithm as it eliminates the cost of reverting rejected moves and provides a quadratic speed-up on the overall cost of obtaining an accepted move.[5]

Before describing in more details how quantum rejection sampling can be used to design a new quantum Metropolis algorithm, let us recall how the standard (classical) Metropolis algorithm works [37]. The goal is to solve the following problem: given a classical Hamiltonian associating energies

---

[5]Martin Schwarz has pointed out to us that this is similar to how [38] provides a speed-up over [36], and that our technique can also be used to speed-up the quantum algorithm in [43] for preparing PEPS.

E_j to a set of possible configurations j, sample from the Gibbs distribution $p(j) = \exp(-\beta E_j)/Z(\beta)$, where β is the inverse temperature and $Z(\beta) = \sum_j \exp(-\beta E_j)$ is the partition function. Since the size of the configuration space is exponential in the number of particles, estimating the Gibbs distribution itself is not an option, therefore the Metropolis algorithm proposes to solve this problem by setting up a random walk on the set of configurations that converges to the Gibbs distribution. More precisely, the random walk works as follows:

1. If i is the current configuration with energy E_i, choose a random move to another configuration j (e. g., for a system of spins, a random move could consist in flipping a random spin), and compute the associated energy E_j.

2. The random move is then accepted or rejected according to the following rule:
   - if $E_j \leq E_i$, then the move is always accepted;
   - if $E_j > E_i$, then the move is only accepted with probability $\exp\big(\beta(E_i - E_j)\big)$.

It can be shown that this random walk converges to the Gibbs distribution.

The quantum Metropolis sampling algorithm by Temme *et al.* [46] follows the same general lines as the classical algorithm. It aims at solving the equivalent problem in the quantum case, where we need to generate the thermal state of a Hamiltonian H, that is, we need to generate a random eigenstate $|\psi_j\rangle$ where j is sampled according to the Gibbs distribution. The fact that the Hamiltonian is quantum, however, adds a few obstacles, since the set of eigenstates $|\psi_j\rangle$ is not known to start with. The main tool to overcome this difficulty is to use quantum phase estimation [31, 17] which, applied on the Hamiltonian H, allows to project any state on an eigenstate $|\psi_j\rangle$, while obtaining an estimate of the corresponding eigenenergy E_j. Similarly to the previous section, we will assume for simplicity that this can be done exactly, that is, we have access to a quantum circuit that acts in the eigenbasis of H as $E_H : |\psi_j\rangle|\bar{0}\rangle \mapsto |\psi_j\rangle|E_j\rangle$, where $|E_j\rangle$ exactly encodes the eigenenergy E_j. We will also assume that the eigenenergies of H are nondegenerate, so that each eigenenergy E_j corresponds to a single eigenstate $|\psi_j\rangle$, instead of a higher dimensional eigenspace. The quantum Metropolis sampling algorithm also requires to choose a set of quantum gates $\mathcal{C}$ that will play the role of the possible random moves between eigenstates. In this case, a given quantum gate $C_l \in \mathcal{C}$ will not simply move an initial eigenstate $|\psi_i\rangle$ to another eigenstate $|\psi_j\rangle$, but rather to a superposition $C_l|\psi_i\rangle = \sum_j c_{ij}^{(l)}|\psi_j\rangle$ where $c_{ij}^{(l)} := \langle\psi_j|C_l|\psi_i\rangle$.

We can now give a high-level description of the quantum Metropolis sampling algorithm by Temme *et al.* [46]. Let $|\psi_i\rangle|E_i\rangle$ be an initial state, that can be prepared by applying the phase estimation operator E_H on an arbitrary state, and measuring the energy register. The algorithm implements each random move by performing the following steps:

1. Apply a random gate $C_l \in \mathcal{C}$ on the first register to prepare the state $\big(C_l|\psi_i\rangle\big)|E_i\rangle = \sum_j c_{ij}^{(l)}|\psi_j\rangle|E_i\rangle$.

2. Apply the phase estimation operator E_H on the $|\psi_j\rangle$ register and an ancilla register initialized in the default state $|\bar{0}\rangle$ to prepare the state $\sum_j c_{ij}^{(l)}|\psi_j\rangle|E_i\rangle|E_j\rangle$.

3. Add another ancilla qubit prepared in the state $|0\rangle$ and apply a controlled-rotation on this register to create the state $\sum_j c_{ij}^{(l)}|\psi_j\rangle|E_i\rangle|E_j\rangle\left[\sqrt{f_{ij}}|1\rangle + \sqrt{1-f_{ij}}|0\rangle\right]$, where $f_{ij} := \min\{1, \exp(\beta(E_i - E_j))\}$.

4. Measure the last qubit. If the outcome is 0, reject the move by reverting the state to $|\psi_i\rangle|E_i\rangle$ (see [46] for details) and go back to Step 1. Otherwise, continue.

5. Discard the $|E_i\rangle$ register and measure the $|E_j\rangle$ register to project the state onto a new eigenstate $|\psi_j\rangle|E_j\rangle$.

It is shown in [46] that by choosing a universal set of quantum gates for the set of moves $\mathcal{C}$, the algorithm simulates random walk on the set of eigenstates of H that satisfies a quantum detailed balanced condition, which ensures that the walk converges to the Gibbs distribution, as in the classical case.

For a given initial state $|\psi_i\rangle|E_i\rangle$, the probability (over all choices of the randomly chosen gate C_l) that the measurement in Step 4 succeeds is $\frac{1}{|\mathcal{C}|}\sum_{j,l} f_{ij}|c_{ij}^{(l)}|^2$. If we define a vector $\boldsymbol{w}^{(i)}$ whose components are $w_{jl}^{(i)} := \sqrt{f_{ij}/|\mathcal{C}|}\,c_{ij}^{(l)}$, then this probability is simply $\|\boldsymbol{w}^{(i)}\|_2^2$. Hence, after one execution of the algorithm (Steps 1-5) the initial state $|\psi_i\rangle|E_i\rangle$ gets mapped to $|\psi_j\rangle|E_j\rangle$ with probability $\sum_l |w_{jl}^{(i)}|^2/\|\boldsymbol{w}^{(i)}\|_2^2$. We could achieve the same random move by converting the initial state $|\psi_i\rangle|E_i\rangle$ to

$$\sum_{j,l} \frac{w_{jl}^{(i)}}{\|\boldsymbol{w}^{(i)}\|_2}|l\rangle|\psi_j\rangle|E_i\rangle|E_j\rangle$$

$$= \frac{1}{\|\boldsymbol{w}^{(i)}\|_2} \sum_j \sqrt{\frac{f_{ij}}{|\mathcal{C}|}}\left[\sum_l c_{ij}^{(l)}|l\rangle\right]|\psi_j\rangle|E_i\rangle|E_j\rangle \quad (21)$$

and discarding the $|l\rangle$ and $|E_i\rangle$ registers and measuring the $|E_j\rangle$ register to project on the state $|\psi_j\rangle|E_j\rangle$ with the correct probability. This implies that one random move reduces to a quantum state conversion problem that we will call $\text{QMETROPOLISMOVE}_{\mathcal{C}}$. This problem assumes that we are able to perform a perfect phase estimation on the Hamiltonian H. Therefore, similarly to the previous section, we fix a set of possible eigenenergies $\mathcal{E}$ of finite cardinality $n := |\mathcal{E}|$.

Definition 8. The *quantum Metropolis move problem*, denoted by $\text{QMETROPOLISMOVE}_{\mathcal{C}}$, is a quantum state conversion problem $(\mathcal{O}, \Phi, \Psi, \mathcal{X})$, where

$$\mathcal{X} := \{(H, i) \in \text{Herm}(d) \times [d] : \text{spec}(H) \in \mathcal{E}^d\}.$$

Oracles in $\mathcal{O}$ act as $E_H : |\psi_j\rangle|\bar{0}\rangle_n \mapsto |\psi_j\rangle|E_j\rangle$, with the corresponding initial states $|\psi_i\rangle$, the eigenvectors of H, and target states $\sum_{j,l} w_{jl}^{(i)}/\|\boldsymbol{w}^{(i)}\|_2|l\rangle|\psi_j\rangle$ where $w_{jl}^{(i)} := \sqrt{f_{ij}/|\mathcal{C}|}\,c_{ij}^{(l)}$, $f_{ij} := \min\{1, \exp(\beta(E_i - E_j))\}$, and $c_{ij}^{(l)} := \langle\psi_j|C_l|\psi_i\rangle$.

A critical part of the algorithm from [46] described above is how to revert a rejected move in Step 4. Temme *et al.* show how this can be done by using an unwinding technique based on successive measurements, but we will not describe this technique in detail, as we now show how this step can be avoided by using quantum rejection sampling. Intuitively, this can be done by using amplitude amplification to ensure that the measurement in Step 4 always projects on the "accept" state $|1\rangle$. This also avoids having to repeatedly

attempt random moves until one is accepted, and the number of steps of amplitude amplification will be quadratically smaller than the number of random moves that have to be attempted until one is accepted. This leads to the following statement:

THEOREM 3. *There is a quantum algorithm that solves* QMETROPOLISMOVE$_\mathcal{C}$ *with success probability 1 using an expected number of queries* $O(1/\|\boldsymbol{w}^{(i)}\|_2)$.

PROOF. The modified algorithm follows the same lines as the original algorithm, except that Steps 3-4 are replaced by a quantum rejection sampling step. We use a quantum coin to choose the random gate in Step 1 in order to make it coherent. The algorithm starts by applying the phase estimation oracle E_H on the initial state to prepare the state $|\psi_i\rangle|E_i\rangle$, and then proceeds with the following steps:

1. Prepare an extra register in the state $\frac{1}{\sqrt{|\mathcal{C}|}}\sum_l |l\rangle$. Conditionally on this register, apply the gate $C_l \in \mathcal{C}$ on the eigenstate register to prepare the state

$$\frac{1}{\sqrt{|\mathcal{C}|}} \sum_j \left[\sum_l c_{ij}^{(l)} |l\rangle \right] |\psi_j\rangle|E_i\rangle.$$

2. Apply the phase estimation operator E_H on the second register and an ancilla register initialized in the default state $|\bar{0}\rangle_n$ to prepare the state

$$\frac{1}{\sqrt{|\mathcal{C}|}} \sum_j \left[\sum_l c_{ij}^{(l)} |l\rangle \right] |\psi_j\rangle|E_i\rangle|E_j\rangle.$$

3. Convert this state to the state given in Eq. (21):

$$\frac{1}{\|\boldsymbol{w}^{(i)}\|_2} \sum_j \sqrt{\frac{f_{ij}}{|\mathcal{C}|}} \left[\sum_l c_{ij}^{(l)} |l\rangle \right] |\psi_j\rangle|E_i\rangle|E_j\rangle.$$

4. Discard $|E_i\rangle$ and uncompute $|E_j\rangle$ by using one call to the phase estimation oracle $E_H^\dagger$.

Note that Step 3 is an instance of SQSAMPLING$_\tau$, where the pair of basis states $|E\rangle|E'\rangle$ of the phase estimation registers plays the role of the states $|k\rangle$, the initial amplitudes $\pi_{E,E'}$ are given by $\frac{1}{\sqrt{|\mathcal{C}|}}\sqrt{\sum_l |c_{ij}^{(l)}|^2}$ for $(E,E') = (E_i, E_j)$ or 0 for values (E,E') that do not correspond to a pair of eigenvalues of H, the states $\left[\sum_l c_{ij}^{(l)}|l\rangle\right]|\psi_j\rangle/\sqrt{\sum_l|c_{ij}^{(l)}|^2}$ play the role of the unknown states $|\xi_k\rangle$, and the ratio between the initial and target amplitudes is given by $\tau_{E,E'} = \sqrt{\min\{1, \exp(\beta(E-E'))\}}$ (the reduction is summarized in Table 2). Therefore, this step may be performed using the algorithm in Lemma 4. Here, we choose $\alpha = 1$ since the full Quantum Metropolis Sampling algorithm requires to apply a large number of successive random moves, therefore each instance of QMETROPOLISMOVE$_\mathcal{C}$ should be solved with high success probability. Choosing $\alpha = 1$ ensures that each random move will have success probability 1 (under our assumption that the phase estimation oracle is perfect), using an expected number of phase estimation oracles $O(1/\|\boldsymbol{w}^{(i)}\|_2)$. $\square$

Note that in this case it is critical that the algorithm only requires one copy of the initial state, hence solving the quantum state conversion problem SQSAMPLING$_\tau$ (in contrast,

the quantum algorithm for linear systems of equations used a unitary to create multiple copies of the initial state, which is allowed only in the weaker quantum state generation problem QSAMPLING$_{\pi\to\sigma}$). Indeed, creating the initial state requires one copy of the previous eigenstate $|\psi_i\rangle$, which cannot be cloned as it is unknown. Here, the algorithm only requires to reflect through the initial state, which can be done by inverting Steps 1-2, applying a phase -1 conditionally on the eigenenergy register being in the state $|E_i\rangle$ (which is possible since E_i is known), and applying Steps 1-2 again.

Repeating the algorithm for QMETROPOLISMOVE$_\mathcal{C}$ a large number of times will simulate the same random walk on the eigenstates of H as the original quantum Metropolis sampling algorithm in [46], except that we have a quadratic speed-up over the number of attempted moves necessary to obtain an accepted move. In order to converge to the Gibbs distribution, we need to take into account this quadratic speed-up in order to decide when to stop the algorithm, effectively assuming that each move takes quadratically longer than it actually does. Another option would be to modify the algorithm for QSAMPLING$_{\pi\to\sigma}$ so that it also estimates $\|\boldsymbol{w}^{(i)}\|_2$ by using amplitude estimation or quantum counting [11, 12]. We leave the full analysis of these technical issues for future work.

5.3 Boolean hidden shift problem

Our final application of the quantum algorithm for the QSAMPLING$_{\pi\to\sigma}$ problem is a new quantum algorithm for the Boolean hidden shift problem BHSP$_f$.

We refer to the recent review [19] for a good overview of basic properties of the Fourier transform of Boolean functions. Fourier analysis on the Boolean cube studies the 2^n-dimensional vector space of all *real*-valued functions defined on the n-dimensional Boolean cube $\mathbb{F}_2^n$. Thus, in the following definition f denotes a function of the form $\mathbb{F}_2^n \to \mathbb{R}$ (*not* a Boolean function). The Boolean case is discussed later.

Definition 9. The *Fourier basis* of $\mathbb{F}_2^n$ consists of functions $\{\chi_w : w \in \mathbb{F}_2^n\}$, where each $\chi_w : \mathbb{F}_2^n \to \{1, -1\}$ is defined as $\chi_w(x) := (-1)^{w\cdot x}$, where $w \cdot x := \sum_{i=1}^n w_i x_i$ is the inner product in $\mathbb{F}_2$. The *Fourier transform* of a function $f : \mathbb{F}_2^n \to \mathbb{R}$ is the function $\hat{f} : \mathbb{F}_2^n \to \mathbb{R}$ defined as $\hat{f}(w) := \frac{1}{2^n}\sum_{x\in\mathbb{F}_2^n}(-1)^{w\cdot x}f(x)$. Note that $\hat{f}(w) := \mathbb{E}_x(\chi_w f) = \frac{1}{2^n}\sum_{x\in\mathbb{F}_2^n}\chi_w(x)f(x)$ which is another way to write the Fourier coefficients. The set $\{\hat{f}(w) : w \in \mathbb{F}_2^n\}$ of all values of $\hat{f}$ is called the *spectrum* of f and each of its elements is called a *Fourier coefficient* of f.

Let us consider a *Boolean function* $f : \mathbb{F}_2^n \to \mathbb{F}_2$. To find its Fourier transform, it is required to associate f with a real-valued function $F : \mathbb{F}_2^n \to \mathbb{R}$ in some way. Instead of the obvious correspondence (treating $0, 1 \in \mathbb{F}_2$ as real numbers) for the purposes of this work it is more natural to let F be the (± 1)-valued function defined by $F(x) := (-1)^{f(x)}$.

Definition 10. By slight abuse of notation, the *Fourier transform of a Boolean function* $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is the function $\hat{f} : \mathbb{F}_2^n \to \mathbb{R}$ defined as

$$\hat{f}(w) := \mathbb{E}_x(\chi_w F) = \frac{1}{2^n} \sum_{x\in\mathbb{F}_2^n} (-1)^{w\cdot x + f(x)}.$$

Based on a reduction to QSAMPLING$_{\pi\to\sigma}$, we can now prove the following upper bound on the query complexity of BHSP$_f$.

SQSampling$_\tau$	QMetropolisMove$_\mathcal{C}$
$\lvert k\rangle$	$\lvert E\rangle\lvert E'\rangle$
π_k	$\pi_{E,E'} := \begin{cases} \frac{1}{\sqrt{\lvert\mathcal{C}\rvert}}\sqrt{\sum_l \lvert c_{ij}^{(l)}\rvert^2} \text{ if } (E,E') = (E_i, E_j) \text{ where } E_i, E_j \in \mathrm{spec}(H) \\ 0 \text{ if } E \notin \mathrm{spec}(H) \text{ or } E' \notin \mathrm{spec}(H) \end{cases}$
$\lvert \xi_k\rangle$	$\lvert \xi_{E,E'}\rangle := \begin{cases} \sum_l c_{ij}^{(l)}\lvert l\rangle\lvert\psi_j\rangle / \sqrt{\sum_l \lvert c_{ij}^{(l)}\rvert^2} \text{ if } (E,E') = (E_i, E_j) \text{ where } E_i, E_j \in \mathrm{spec}(H) \\ \text{n/a if } E \notin \mathrm{spec}(H) \text{ or } E' \notin \mathrm{spec}(H) \end{cases}$
τ_k	$\tau_{E,E'} := \sqrt{\min\{1, \exp(\beta(E - E'))\}}$

Table 2: Reduction from Step 3 in the new quantum Metropolis sampling algorithm to the quantum resampling problem SQSampling$_\tau$.

THEOREM 4. *Let f be a Boolean function and $\hat{f}$ be its Fourier transform. For any p, we have $Q_{1-p}(\mathrm{BHSP}_f) = O(1/\lVert \varepsilon_{\boldsymbol{\pi}\to\boldsymbol{\sigma}}^p \rVert_2)$, where components of $\boldsymbol{\pi}$ and $\boldsymbol{\sigma}$ are given by $\pi_w = \lvert \hat{f}_w\rvert$ and $\sigma_w = 1/\sqrt{2^n}$ for $w \in \mathbb{F}_2^n$.*

PROOF. We will use O_{f_s} to denote the *phase oracle* for function f_s, i.e., a diagonal matrix that acts on the standard basis vectors $x \in \mathbb{F}_2^n$ as follows: $O_{f_s}\lvert x\rangle := (-1)^{f(x+s)}\lvert x\rangle$. Let us consider the quantum oracle O_{f_s} conjugated by the Hadamard transform. The resulting operation

$$V(s) := H^{\otimes n} O_{f_s} H^{\otimes n}$$

is very useful, since, when acting on a register initialized in all-zeros state, it can be used to prepare the following quantum superposition:

$$\lvert \psi_{\hat{f}}(s)\rangle := V(s)\lvert 0\rangle^{\otimes n} = \sum_{w \in \mathbb{F}_2^n}(-1)^{w\cdot s}\hat{f}(w)\lvert w\rangle.$$

If we could eliminate the Fourier coefficients $\hat{f}(w)$ from state $\lvert \psi_{\hat{f}}(s)\rangle$, we would obtain a state

$$\lvert \psi(s)\rangle := \frac{1}{\sqrt{2^n}}\sum_{w \in \mathbb{F}_2^n}(-1)^{w\cdot s}\lvert w\rangle$$

from which the hidden shift s can be easily recovered by applying the Hadamard transform $H^{\otimes n}$. Luckily, the problem of transforming the state $\lvert \psi_{\hat{f}}(s)\rangle$ to $\lvert \psi(s)\rangle$ is a special case of QSampling$_{\boldsymbol{\pi}\to\boldsymbol{\sigma}}$ with

$$\pi_w := \lvert \hat{f}(w)\rvert, \quad \sigma_w := 1/\sqrt{2^n}, \quad \lvert \xi_w\rangle := (-1)^{w\cdot s}\lvert 0\rangle.$$

(More precisely, the initial amplitudes are $\hat{f}(w)$ instead of $\lvert \hat{f}(w)\rvert$. However, the function f and therefore its Fourier transform $\hat{f}$ is completely known, so we can easily correct the phases using a controlled-phase gate.) As a consequence, Theorem 1 immediately gives us a quantum algorithm for solving this problem. $\square$

The complexity of the algorithm is limited by the smallest Fourier coefficient of the function. By ignoring small Fourier coefficients, one can decrease the complexity of the algorithm, at the cost of a lower success probability. However, the success probability of this algorithm can be boosted using repetitions, which requires to construct a procedure to check a candidate shift. We propose such a checking procedure based on a controlled-SWAP test. The number of necessary repetitions may then be decreased quadratically using the amplitude amplification technique of [28] (note

that we cannot use the usual amplitude amplification algorithm since the checking procedure is imperfect). This leads to the following theorem (proved in Appendix B):

THEOREM 5. *Let f be a Boolean function and $\hat{f}$ be its Fourier transform. Moreover, let $p, \gamma \in [0,1]$ be such that $\lVert \hat{\varepsilon}\rVert_1 = \sqrt{2^n p}$, where $\varepsilon_w = \min\{\lvert \hat{f}_w\rvert, \gamma/\sqrt{2^n}\}$. Then, for any $\delta > 0$, we have $Q_\delta(\mathrm{BHSP}_f) = O\big(\frac{1}{\sqrt{p}}(1/\lVert \varepsilon\rVert_2 + 1/\sqrt{\mathrm{I}_f})\big)$.*

Conclusion and open problems

We provide an algorithm for solving the quantum resampling problem. Our algorithm can be viewed as a quantum version of the classical rejection sampling technique. It relies on amplitude amplification [12] to increase the amplitude of some target "accept" state, and its query complexity is given by a semidefinite program. The solution of this SDP and hence the cost of the algorithm depends on the ratio between the amplitudes of the initial and target states, similarly to the case of the classical rejection sampling where the cost is given by the ratio of probabilities. Using the automorphism principle over a unitary group, we derive an SDP for the lower bound that is identical to the one for the upper bound, showing that our algorithm has optimal query complexity.

While the original adversary method cannot be applied as is for this quantum state generation problem because the oracle encodes an unknown quantum state instead of some unknown classical data, it is interesting to note that the query complexity of this problem is also characterized by an SDP. Therefore, an interesting open question is whether the adversary method [5, 27], which has been shown to be tight for evaluating functions [40, 41, 34] and nearly tight for quantum state generation or conversion problems with classical oracles [34], can always be extended and shown to be tight for this more general framework of problems with quantum oracles.

In Sect. 5, we illustrate how quantum rejection sampling may be used as a primitive in algorithm design by providing three different applications. We first show that it was used implicitly in the quantum algorithm for linear systems of equations [26]. By assuming a perfect phase estimation operator on the matrix of the system, we show that this problem reduces to a quantum state conversion problem which we call QLinearEquations$_\kappa$, which itself reduces to SQSampling$_\tau$. An open question is how to combine the quantum rejection sampling approach with the variable time amplitude amplification technique that was proposed by Ambainis [6] to improve on the original algorithm by Harrow *et al.* [26]. In order to do so, we should "open" the phase

estimation black box since Ambainis's main idea is to stop some branches of the phase estimation earlier than others.

As a second application, we show that quantum rejection sampling can be used to speed up the main step in the original quantum Metropolis sampling algorithm [46]. The general idea is to use amplitude amplification to increase the acceptance probability of a move, and therefore quadratically reduce the number of moves that have to be attempted before one is accepted. While this approach also provides some type of quadratic speed-up, it is rather different from the "quantum-quantum" Metropolis algorithm proposed by Yung and Aspuru-Guzik [52]. The main difference is that the approach based on quantum rejection sampling still simulates the same classical random walk on the eigenstates of the Hamiltonian, whereas the quantum-quantum Metropolis algorithm replaces it by a quantum walk. Note that while random walks converge towards their stationary distribution from any initial state, this is not the case for quantum walks as they are reversible by definition. Therefore, while both the original quantum Metropolis sampling algorithm and our variation can start from any initial state and run at a fixed inverse temperature β to converge to the corresponding Gibbs distribution, the quantum-quantum Metropolis sampling algorithm works differently: it starts from a uniform superposition, which corresponds to the Gibbs distribution at $\beta = 0$, and uses a series of measurements to project this state onto superpositions corresponding to Gibbs distributions with increasingly large β, until the desired value is reached.

Finally. as shown in Sect. 5.3, we can apply the quantum rejection sampling technique to solve the hidden shift problem for any Boolean function f. In the limiting cases of flat or highly peaked Fourier spectra we recover the quantum algorithm for bent functions [42] or Grover's algorithm for delta functions [23], respectively. For a general Boolean function the hidden shift problem can be seen as lying somewhere between these two extreme cases. While the algorithm is known to be optimal for the extreme cases of bent and delta functions, its optimality for more general cases remains an open problem. A related question is the optimality of the checking procedure that leads to Theorem 5.

Acknowledgements

6. REFERENCES

[1] S. Aaronson. Quantum copy-protection and quantum money. In *Proceedings of the 24th Annual IEEE Conference on Computational Complexity (CCC'09)*, pages 229–242. IEEE Computer Society, 2009.

[2] S. Aaronson and A. Drucker. A full characterization of quantum advice. In *Proceedings of the 42nd Annual ACM Symposium on Theory of Computing (STOC'10)*, pages 131–140. ACM, 2010.

[3] D. Aharonov and O. Regev. Lattice problems in NP ∩ coNP. *Journal of the ACM*, 52(5):749–765, 2005. (Earlier version in FOCS'04).

[4] D. Aharonov and A. Ta-Shma. Adiabatic quantum state generation and statistical zero knowledge. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing (STOC'03)*, pages 20–29. ACM, 2003.

[5] A. Ambainis. Quantum lower bounds by quantum arguments. In *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing (STOC'00)*, pages 636–643. ACM, 2000.

[6] A. Ambainis. Variable time amplitude amplification and a faster quantum algorithm for solving systems of linear equations. arxiv:1010.4458, 2010.

[7] A. Ambainis, L. Magnin, M. Roetteler, and J. Roland. Symmetry-assisted adversaries for quantum state generation. In *Proceedings of the 26th Annual IEEE Conference on Computational Complexity (CCC'11)*, pages 167–177. IEEE Computer Society, 2011.

[8] C. H. Bennett, E. Bernstein, G. Brassard, and U. Vazirani. Strengths and weaknesses of quantum computing. *SIAM Journal on Computing*, 26(5):1510–1523, 1997.

[9] D. W. Berry, G. Ahokas, R. Cleve, and B. C. Sanders. Efficient quantum algorithms for simulating sparse Hamiltonians. *Communications in Mathematical Physics*, 270(2):9, 2005.

[10] S. Boixo, E. Knill, and R. D. Somma. Eigenpath traversal by phase randomization. *Quantum Information and Computation*, 9(9,10):833–855, 2009.

[11] M. Boyer, G. Brassard, P. Hø yer, and A. Tapp. Tight bounds on quantum searching. *Fortschritte der Physik*, 46(4-5):493–505, 1998.

[12] G. Brassard, P. Høyer, M. Mosca, and A. Tapp. Quantum amplitude amplification and estimation. arxiv:quant-ph/0005055, 2000.

[13] H. Buhrman, R. Cleve, J. Watrous, and R. de Wolf. Quantum fingerprinting. *Physical Review Letters*, 87(16):167902, 2001.

[14] H. Buhrman and R. de Wolf. Complexity measures and decision tree complexity: A survey. *Theoretical Computer Science*, 288(1):21–43, 2002.

[15] A. M. Childs. On the relationship between continuous- and discrete-time quantum walk. *Communications in Mathematical Physics*, 294(2):22, 2008.

[16] A. M. Childs and R. Kothari. Simulating Sparse Hamiltonians with Star Decompositions. In *Theory of Quantum Computation, Communication, and Cryptography (TQC 2010)*, volume 6519 of *Lecture Notes in Computer Science*, page 11, Berlin, Heidelberg, 2011. Springer.

[17] R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca. Quantum Algorithms Revisited. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 454(1969):18, 1997.

[18] W. Dam, S. Hallgren, and L. Ip. Quantum algorithms for some hidden shift problems. In *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'03)*, pages 489–498, 2003.

[19] R. de Wolf. A brief introduction to Fourier analysis on the Boolean cube. *Theory of Computing Library – Graduate Surveys*, 1:1–20, 2008.

[20] L. Devroye. *Non-uniform random variate generation.*

Springer, New York, 1986.

[21] E. Farhi, D. Gosset, A. Hassidim, A. Lutomirski, and P. Shor. Quantum money from knots. In *Proceedings of the 3rd Innovations in Theoretical Computer Science (ITCS'12) conference*, 2012. To appear.

[22] K. Friedl, G. Ivanyos, F. Magniez, M. Santha, and P. Sen. Hidden Translation and Orbit Coset in Quantum Computing. In *Proceedings of the 35fth Annual ACM Symposium on Theory of Computing (STOC'03)*, pages 1–9. ACM, 2002.

[23] L. K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC'96)*, pages 212–219, New York, 1996. ACM.

[24] L. K. Grover. Synthesis of quantum superpositions by quantum computation. *Physical Review Letters*, 85(6):1334–1337, 2000.

[25] L. K. Grover and T. Rudolph. Creating superpositions that correspond to efficiently integrable probability distributions. arxiv:0208112, 2002.

[26] A. W. Harrow, A. Hassidim, and S. Lloyd. Quantum algorithm for linear systems of equations. *Physical Review Letters*, 103(15):150502, 2009.

[27] P. Høyer, T. Lee, and R. Špalek. Negative weights make adversaries stronger. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing (STOC'07)*, pages 526–535. ACM, 2007.

[28] P. Høyer, M. Mosca, and R. de Wolf. Quantum search on bounded-error inputs. In *Proceedings of the 30th International Colloquium on Automata, Languages and Programming (ICALP'03)*, volume 2719 of *Lecture Notes in Computer Science*, pages 291–299. Springer, 2003.

[29] G. Ivanyos. On solving systems of random linear disequations. *Quantum Information and Computation*, 8(6&7):579–594, 2008.

[30] J. Kempe, A. Kitaev, and O. Regev. The complexity of the local Hamiltonian problem. *SIAM Journal on Computing*, 35(5):1070–1097, 2006.

[31] A. Kitaev. Quantum measurements and the Abelian Stabilizer Problem. arxiv:quant-ph/9511026, 1995.

[32] A. Kitaev and W. A. Webb. Wavefunction preparation and resampling using a quantum computer. arxiv:0801.0342, 2008.

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

[34] T. Lee, R. Mittal, B. W. Reichardt, R. Špalek, and M. Szegedy. Quantum query complexity of state conversion. In *Proceedings of the 52nd Annual IEEE Symposium on Foundations of Computer Science (FOCS'11)*, 2011. To appear.

[35] G. Letac. On building random variables of a given distribution. *The Annals of Probability*, 3(2):298–306, 1975.

[36] C. Marriott and J. Watrous. Quantum Arthur–Merlin games. *Computational Complexity*, 14(2):122–152, 2005.

[37] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087, 1953.

[38] D. Nagaj, P. Wocjan, and Y. Zhang. Fast amplification of QMA. *Quantum Information and Computation*, 9(11&12):1053–1068, 2009.

[39] O. Regev. Quantum computation and lattice problems. *SIAM Journal on Computing*, 33(2):738–760, 2004.

[40] B. W. Reichardt. Span Programs and Quantum Query Complexity: The General Adversary Bound Is Nearly Tight for Every Boolean Function. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS'09)*, pages 544–551. IEEE Computer Society Press, 2009.

[41] B. W. Reichardt. Reflections for quantum query algorithms. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'11)*, pages 560–569, 2011.

[42] M. Rötteler. Quantum algorithms for highly non-linear Boolean functions. In *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'10)*, pages 448–457, 2010.

[43] M. Schwarz, K. Temme, and F. Verstraete. Contracting tensor networks and preparing PEPS on a quantum computer. arxiv:1104.1410, 2011.

[44] L. Sheridan, D. Maslov, and M. Mosca. Approximating fractional time quantum evolution. *Journal of Physics A*, 42(18):185302, 2009.

[45] R. D. Somma, S. Boixo, H. Barnum, and E. Knill. Quantum simulations of classical annealing processes. *Physical Review Letters*, 101:130504, 2008.

[46] K. Temme, T. J. Osborne, K. G. Vollbrecht, D. Poulin, and F. Verstraete. Quantum Metropolis sampling. *Nature*, 471(7336):87–90, 2011.

[47] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38(1):49, 1996.

[48] U. Vazirani. On the power of quantum computation. *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, 356(1743):1759–1768, 1998.

[49] J. von Neumann. Various techniques used in connection with random digits. *National Bureau of Standards, Applied Math Series*, 12:36–38, 1951.

[50] J. Watrous. Succinct quantum proofs for properties of finite groups. In *Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science (FOCS'00)*, pages 537–546. IEEE Computer Society, 2000.

[51] J. Watrous. Quantum algorithms for solvable groups. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing (STOC'01)*, pages 60–67. ACM, 2001.

[52] M.-H. Yung and A. Aspuru-Guzik. A quantum-quantum Metropolis algorithm. arxiv:1011.1468, 2010.

APPENDIX

A. WATER-FILLING VECTOR IS OPTIMAL FOR THE SDP

LEMMA 1. *Let $p \in [p_{\min}, p_{\max}]$, and $\varepsilon = \varepsilon_{\pi \to \sigma}^p$. Then,*

the following SDP

$$\max_{M \succeq 0} \operatorname{Tr} M \quad s.t. \quad \begin{aligned} &\forall k : \pi_k^2 \geq M_{kk}, \\ &\operatorname{Tr}\left[(\boldsymbol{\sigma} \cdot \boldsymbol{\sigma}^\mathsf{T} - pI)M\right] \geq 0. \end{aligned} \quad (4)$$

has optimal value $\|\boldsymbol{\varepsilon}\|_2^2$, which is achieved by the rank-1 matrix $M = \boldsymbol{\varepsilon} \cdot \boldsymbol{\varepsilon}^\mathsf{T}$.

PROOF. We now show that the optimal value of the SDP in Eq. (4) can be attained by a rank-1 matrix M. Imposing the additional constraint that M can be written as $M = \boldsymbol{\varepsilon} \cdot \boldsymbol{\varepsilon}^\mathsf{T}$ for some $\boldsymbol{\varepsilon} \in \mathbb{R}^n$, the optimization problem (4) reduces to

$$\max_{\varepsilon_k \geq 0} \|\boldsymbol{\varepsilon}\|_2^2 \quad s.t. \quad \begin{aligned} &\forall k : \pi_k \geq \varepsilon_k \geq 0, \\ &\boldsymbol{\sigma}^\mathsf{T} \cdot \hat{\boldsymbol{\varepsilon}} \geq \sqrt{p}, \end{aligned} \quad (22)$$

where $\hat{\boldsymbol{\varepsilon}} := \boldsymbol{\varepsilon}/\|\boldsymbol{\varepsilon}\|_2$ denotes a unit vector in direction $\boldsymbol{\varepsilon}$.

We show that the optimal value is attained by $\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}_{\boldsymbol{\pi} \to \boldsymbol{\sigma}}^p$. Recall that by Def. 5, we have $\varepsilon_k = \min\{\pi_k, \gamma\sigma_k\}$ and

$$\boldsymbol{\sigma}^\mathsf{T} \cdot \hat{\boldsymbol{\varepsilon}} = \sqrt{p}, \quad (23)$$

so that this vector satisfies the constraints in (22) and is therefore a feasible point. As a consequence $M = \boldsymbol{\varepsilon} \cdot \boldsymbol{\varepsilon}^\mathsf{T}$ is also a feasible point for the SDP (4), which implies that its objective value is at least $\operatorname{Tr} M = \|\boldsymbol{\varepsilon}\|_2^2$.

We now want to find a feasible dual solution that gives the same objective value for the dual of SDP (4), which can be written as [47]

$$\min_{\lambda_k \geq 0, \, \mu \geq 0} \sum_{k=1}^{n} \lambda_k \pi_k^2 \quad s.t. \quad \Lambda - I + \mu(pI - \boldsymbol{\sigma} \cdot \boldsymbol{\sigma}^\mathsf{T}) \succeq 0, \quad (24)$$

where $\Lambda := \operatorname{diag}(\lambda_k \mid k = 1, \ldots, n)$. Indeed, if an objective value is feasible for both the primal and the dual, it implies that this is the optimal value.

We prove that the following solution is feasible for the dual:

$$\lambda_k = \mu\left(\frac{\sigma_k}{\varepsilon_k}\sum_{l=1}^{n}\sigma_l\varepsilon_l - p\right) + 1,$$

$$\mu = \frac{1 - \|\boldsymbol{\varepsilon}\|_2^2}{p - \left(\sum_{l=1}^{n}\sigma_l\varepsilon_l\right)\cdot\left(\sum_{k=1}^{n}\frac{\sigma_k\pi_k^2}{\varepsilon_k}\right)}. \quad (25)$$

This choice yields $\|\boldsymbol{\varepsilon}\|_2^2$ as the dual objective value, so it remains to show that it satisfies the constraints in (24). Let us first prove that $\mu \geq 0$, which is equivalent to

$$\left(\sum_{l=1}^{n}\sigma_l\varepsilon_l\right)\cdot\left(\sum_{k=1}^{n}\frac{\sigma_k\pi_k^2}{\varepsilon_k}\right) \leq p. \quad (26)$$

Let us decompose the vector $\boldsymbol{\pi}$ into two orthogonal parts such that $\boldsymbol{\pi} = \boldsymbol{\pi}_\leq + \boldsymbol{\pi}_>$, where $\boldsymbol{\pi}_\leq$ corresponds to components π_k such that $\pi_k \leq \gamma\sigma_k$, and $\boldsymbol{\pi}_>$ to the remaining components. Decomposing $\boldsymbol{\sigma}$ and $\boldsymbol{\varepsilon}$ similarly, we have $\boldsymbol{\varepsilon} = \boldsymbol{\pi}_\leq + \gamma\boldsymbol{\sigma}_>$. The following are straightforward

$$1 = \|\boldsymbol{\pi}_\leq\|_2^2 + \|\boldsymbol{\pi}_>\|_2^2$$
$$\|\boldsymbol{\varepsilon}\|_2^2 = \|\boldsymbol{\pi}_\leq\|_2^2 + \gamma^2\|\boldsymbol{\sigma}_>\|_2^2$$
$$\boldsymbol{\varepsilon}^\mathsf{T}\cdot\boldsymbol{\sigma} = \boldsymbol{\pi}_\leq^\mathsf{T}\cdot\boldsymbol{\sigma}_\leq + \gamma\|\boldsymbol{\sigma}_>\|_2^2.$$

Using these equalities, we obtain

$$\sum_{k=1}^{n}\frac{\sigma_k\pi_k^2}{\varepsilon_k} = \boldsymbol{\pi}_\leq^\mathsf{T}\cdot\boldsymbol{\sigma}_\leq + \frac{1}{\gamma}\|\boldsymbol{\pi}_>\|_2^2$$

$$= \boldsymbol{\pi}_\leq^\mathsf{T}\cdot\boldsymbol{\sigma}_\leq + \frac{1}{\gamma}\left(1 - \|\boldsymbol{\pi}_\leq\|_2^2\right)$$

$$= \boldsymbol{\varepsilon}^\mathsf{T}\cdot\boldsymbol{\sigma} + \frac{1}{\gamma}\left(1 - \|\boldsymbol{\varepsilon}\|_2^2\right). \quad (27)$$

Therefore, the left hand side of (26) can be written as

$$\left(\sum_{l=1}^{n}\sigma_l\varepsilon_l\right)\cdot\left(\sum_{k=1}^{n}\frac{\sigma_k\pi_k^2}{\varepsilon_k}\right) = \left(\boldsymbol{\varepsilon}^\mathsf{T}\cdot\boldsymbol{\sigma}\right)^2 + \frac{\boldsymbol{\varepsilon}^\mathsf{T}\cdot\boldsymbol{\sigma}}{\gamma}\left(1 - \|\boldsymbol{\varepsilon}\|_2^2\right)$$

$$= p\|\boldsymbol{\varepsilon}\|_2^2 + \frac{\boldsymbol{\varepsilon}^\mathsf{T}\cdot\boldsymbol{\sigma}}{\gamma}\left(1 - \|\boldsymbol{\varepsilon}\|_2^2\right)$$

$$= \left(p - \frac{\boldsymbol{\varepsilon}^\mathsf{T}\cdot\boldsymbol{\sigma}}{\gamma}\right)\|\boldsymbol{\varepsilon}\|_2^2 + \frac{\boldsymbol{\varepsilon}^\mathsf{T}\cdot\boldsymbol{\sigma}}{\gamma},$$

where we have used (23). Since $\varepsilon_k \leq \gamma\sigma_k$, we have $\|\boldsymbol{\varepsilon}\|_2^2 \leq \gamma\boldsymbol{\varepsilon}^\mathsf{T}\cdot\boldsymbol{\sigma}$, which, together with (23) implies that $\frac{\boldsymbol{\varepsilon}^\mathsf{T}\cdot\boldsymbol{\sigma}}{\gamma} \leq p$. Together with $\|\boldsymbol{\varepsilon}\|_2^2 \leq \|\boldsymbol{\pi}\|_2^2 = 1$, this implies

$$\left(p - \frac{\boldsymbol{\varepsilon}^\mathsf{T}\cdot\boldsymbol{\sigma}}{\gamma}\right)\|\boldsymbol{\varepsilon}\|_2^2 + \frac{\boldsymbol{\varepsilon}^\mathsf{T}\cdot\boldsymbol{\sigma}}{\gamma} \leq p,$$

which proves (26) and, in turn, $\mu \geq 0$.

We now show that $\lambda_k \geq 0$ for all $k \in [n]$. From Eqs. (25) we see that this is equivalent to showing

$$\frac{\sigma_k}{\varepsilon_k}\boldsymbol{\varepsilon}^\mathsf{T}\cdot\boldsymbol{\sigma} - p \geq -\frac{p - \boldsymbol{\varepsilon}^\mathsf{T}\cdot\boldsymbol{\sigma}\sum_{k=1}^{n}\frac{\sigma_k\pi_k^2}{\varepsilon_k}}{1 - \|\boldsymbol{\varepsilon}\|_2^2}$$

Note that $1 \geq \|\boldsymbol{\varepsilon}\|_2^2$. By multiplying out everything with $1 - \|\boldsymbol{\varepsilon}\|_2^2$ and expanding, we get

$$\frac{\sigma_k}{\varepsilon_k}\boldsymbol{\varepsilon}^\mathsf{T}\cdot\boldsymbol{\sigma}\left(1 - \|\boldsymbol{\varepsilon}\|_2^2\right) + p\|\boldsymbol{\varepsilon}\|_2^2 \geq \boldsymbol{\varepsilon}^\mathsf{T}\cdot\boldsymbol{\sigma}\sum_{k=1}^{n}\frac{\sigma_k\pi_k^2}{\varepsilon_k}.$$

Note that $p\|\boldsymbol{\varepsilon}\|_2^2 = (\boldsymbol{\varepsilon}^\mathsf{T}\cdot\boldsymbol{\sigma})^2$, so after rearranging terms and dividing by $\boldsymbol{\varepsilon}^\mathsf{T}\cdot\boldsymbol{\sigma}$ we get

$$\frac{\sigma_k}{\varepsilon_k}\left(1 - \|\boldsymbol{\varepsilon}\|_2^2\right) + \boldsymbol{\varepsilon}^\mathsf{T}\cdot\boldsymbol{\sigma} \geq \sum_{k=1}^{n}\frac{\sigma_k\pi_k^2}{\varepsilon_k}.$$

We apply Eq. (27) to the right hand side and get

$$\frac{\sigma_k}{\varepsilon_k}\left(1 - \|\boldsymbol{\varepsilon}\|_2^2\right) + \boldsymbol{\varepsilon}^\mathsf{T}\cdot\boldsymbol{\sigma} \geq \boldsymbol{\varepsilon}^\mathsf{T}\cdot\boldsymbol{\sigma} + \frac{1}{\gamma}\left(1 - \|\boldsymbol{\varepsilon}\|_2^2\right).$$

After simplification this yields $\varepsilon_k \leq \gamma\sigma_k$, which is true by definition of $\boldsymbol{\varepsilon}$. Thus, we have $\lambda_k \geq 0$.

Finally, it remains to show that the following matrix is positive semidefinite:

$$\Lambda - I + \mu(pI - \boldsymbol{\sigma}\cdot\boldsymbol{\sigma}^\mathsf{T}) = \mu\left[\left(\sum_{l=1}^{n}\sigma_l\varepsilon_l\right)\cdot\operatorname{diag}(\sigma_k/\varepsilon_k) - \boldsymbol{\sigma}\cdot\boldsymbol{\sigma}^\mathsf{T}\right].$$

Since $\mu \geq 0$, it is the case if and only if

$$\forall \boldsymbol{v} \in \mathbb{R}^n : \quad \left(\sum_{l=1}^{n}\sigma_l\varepsilon_l\right)\cdot\left(\sum_{k=1}^{n}\frac{v_k^2\sigma_k}{\varepsilon_k}\right) \geq \left(\sum_{k=1}^{n}v_k\sigma_k\right)^2.$$

This follows by Cauchy–Schwarz inequality: $\left(\sum_{l=1}^{n}\sigma_l\varepsilon_l\right)\cdot\left(\sum_{k=1}^{n}v_k^2\sigma_k/\varepsilon_k\right) \geq \left(\sum_{k=1}^{n}\sqrt{\sigma_k\varepsilon_k}\cdot\sqrt{v_k^2\sigma_k/\varepsilon_k}\right)^2$. □

B.   BOOSTING THE SUCCESS PROBABILITY

When we want to find the hidden shift with probability close to one, the algorithm in Theorem 4 might be quite inefficient as, in the special case $p = 1$, its complexity is limited by the smallest Fourier coefficient (in particular, when some Fourier coefficients are zero, it is not possible to obtain $p = 1$). Another approach to find the hidden shift with near certainty would be to use the algorithm with a smaller p and repeat until the right shift is obtained. However, this requires a procedure to check a candidate shift. Classically this can be done, *e. g.*, by querying the oracle on several inputs uniformly at random and checking if the output agrees with the shifted function. Using this strategy one can boost the success probability arbitrarily close to 1 by using $O(1/\mathrm{I}_f)$ queries to the shifted function f_s.

Definition 11. For any Boolean function f and $v \in \mathbb{F}_2^n$, we call $\mathrm{I}_f(v) = \mathrm{Pr}_x[f(x) \neq f(x + v)]$ the *influence* of v over f, and $\mathrm{I}_f = \min_v \mathrm{I}_f(v)$ the *minimum influence* over f.

However, on a quantum computer one can check a candidate shift with only $O(1/\sqrt{\mathrm{I}_f})$ oracle calls using a procedure similar to controlled-SWAP test. If we combine this imperfect checking procedure with the quantum search algorithm with bounded-error inputs from [28], we get the following result:

THEOREM 5. *Let f be a Boolean function and $\hat{f}$ be its Fourier transform. Moreover, let $p, \gamma \in [0, 1]$ be such that $\|\hat{\varepsilon}\|_1 = \sqrt{2^n p}$, where $\varepsilon_w = \min\{|\hat{f}_w|, \gamma/\sqrt{2^n}\}$. Then, for any $\delta > 0$, we have $Q_\delta(\mathrm{BHSP}_f) = O\left(\frac{1}{\sqrt{p}}(1/\|\varepsilon\|_2 + 1/\sqrt{\mathrm{I}_f})\right)$.*
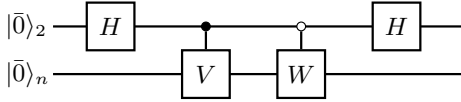

Figure 5: A one-sided test for checking if states $|\psi\rangle$ and $|\phi\rangle$ are equal. If $V|\bar{0}\rangle_n = |\psi\rangle$ and $W|\bar{0}\rangle_n = |\varphi\rangle$ then the above circuit prepares a state $\frac{1}{2}\big[|0\rangle(|\psi\rangle + |\varphi\rangle) + |1\rangle(|\psi\rangle - |\varphi\rangle)\big]$. If the first register is measured in the standard basis, we get outcome 0 ("accept") with probability $p(0) = \frac{1}{4}\||\psi\rangle + |\varphi\rangle\|_2^2 = \frac{1}{2}(1 + \mathrm{Re}\langle\psi|\varphi\rangle)$. Note that we never get outcome 1 if $|\psi\rangle = |\varphi\rangle$.

PROOF. Let us first show how to check a candidate shift v, assuming that s is the actual shift. Using one call to the oracle O_{f_s}, we can easily prepare the state

$$|\phi_f(s)\rangle = \frac{1}{\sqrt{2^n}} \sum_x (-1)^{f(x+s)} |x\rangle$$

Similarly, we can prepare the state $|\phi_f(v)\rangle$ by applying a transformation corresponding to oracle O_{f_v}. The inner product between these two states is

$$\langle\phi_f(s)|\phi_f(v)\rangle = \frac{1}{2^n} \sum_x (-1)^{f(x+s)+f(x+v)}$$
$$= 1 - 2\mathrm{I}_f(s + v).$$

Therefore, the procedure from Fig. 5 will accept the candidate shift v with probability

$$\frac{1 + \langle\phi_f(s)|\phi_f(v)\rangle}{2} = 1 - \mathrm{I}_f(s + v),$$

that is, it will always accept if $s = v$, and reject with probability at least I_f otherwise. Repeating this test $O(1/\mathrm{I}_f)$ times, we can ensure that a wrong candidate shift is rejected with high probability. Moreover, using once again quantum amplitude amplification [12], we can obtain a quadratic improvement, and therefore reject a wrong candidate using only $O(1/\sqrt{\mathrm{I}_f})$ oracle calls.

Using the algorithm from Theorem 4 for success probability p, we can then boost the success probability to any constant $1 - \delta$ using the quantum search algorithm with bounded-error inputs of [28] (note that using the usual quantum amplitude amplification technique [12] would incur an additional factor of $\log(1/p)$ since the checking operation is imperfect). Therefore, the total complexity comes from repeating $O(1/\sqrt{p})$ times the algorithm in Theorem 4, with cost $O(1/\|\varepsilon\|_2)$, and the checking operation, with cost $O(1/\sqrt{\mathrm{I}_f})$. □