Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics

András Gilyén[*] Yuan Su[†] Guang Hao Low[‡] Nathan Wiebe[§]

June 6, 2018

Abstract

Quantum computing is powerful because unitary operators describing the time-evolution of a quantum system have exponential size in terms of the number of qubits present in the system. We develop a new "Singular value transformation" algorithm capable of harnessing this exponential advantage, that can apply polynomial transformations to the singular values of a block of a unitary, generalizing the optimal Hamiltonian simulation results of Low and Chuang [LC17a]. The proposed quantum circuits have a very simple structure, often give rise to optimal algorithms and have appealing constant factors, while typically only use a constant number of ancilla qubits.

We show that singular value transformation leads to novel algorithms. We give an efficient solution to a "non-commutative" measurement problem used for efficient ground-state-preparation of certain local Hamiltonians, and propose a new method for singular value estimation. We also show how to exponentially improve the complexity of implementing fractional queries to unitaries with a gapped spectrum. Finally, as a quantum machine learning application we show how to efficiently implement principal component regression.

"Singular value transformation" is conceptually simple and efficient, and leads to a unified framework of quantum algorithms incorporating a variety of quantum speed-ups. We illustrate this by showing how it generalizes a number of prominent quantum algorithms, and quickly derive the following algorithms: optimal Hamiltonian simulation, implementing the Moore-Penrose pseudoinverse with exponential precision, fixed-point amplitude amplification, robust oblivious amplitude amplification, fast QMA amplification, fast quantum OR lemma, certain quantum walk results and several quantum machine learning algorithms.

In order to exploit the strengths of the presented method it is useful to know its limitations too, therefore we also prove a lower bound on the efficiency of singular value transformation, which often gives optimal bounds.

[*] QuSoft, CWI and University of Amsterdam, the Netherlands. Supported by ERC Consolidator Grant 615307-QPROGRESS. gilyen@cwi.nl

[†] Department of Computer Science, Institute for Advanced Computer Studies, and Joint Center for Quantum Information and Computer Science, University of Maryland, USA. buptsuyuan@gmail.com

[‡] Quantum Architectures and Computing group, Microsoft Research, USA. GuangHao.Low@microsoft.com

[§] Quantum Architectures and Computing group, Microsoft Research, USA. nawiebe@microsoft.com

Contents

1 Introduction

It is often said in quantum computing that there are only a few quantum algorithms that are known to give speed-ups over classical computers. While this is true, a remarkable number of applications stem from from these primitives. The first class of quantum speedups is derived from quantum simulation which was originally proposed by Feynman [Fey82]. Such algorithms yield exponential speedups over the best known classical methods for simulating quantum dynamics as well as probing electronic structure problems in material science and chemistry. The two most influential quantum algorithms developed later in the 90's are Shor's algorithm [Sho97] (based on quantum Fourier transform) and Grover's search [Gro96]. Other examples have emerged over the years, but arguably quantum walks [Sze04] and the quantum linear systems algorithm of Harrow, Hassidim and Lloyd [HHL09] are the most common other primitives that provide speed-ups relative to classical computing. An important question that remains is whether "are these primitives truly independent or can they be seen as examples of a deeper underlying concept?" The aim of this work is to provide an argument that a wide array of techniques from these disparate fields can all be seen as manifestations of a single quantum idea that we call "singular value transformation" generalizing all the above mentioned techniques except for quantum Fourier transform.

Of the aforementioned quantum algorithms, quantum simulation is arguably the most diverse and rapidly developing. Within the last few years a host of techniques have been developed that have led to ever more powerful methods [CMN$^+$17]. The problem in quantum simulation fundamentally is to take an efficient description of a Hamiltonian H, an evolution time t, and an error tolerance ε, and find a quantum operation V such that $\left\|e^{-iHt} - V\right\| \leq \varepsilon$ while the implementation of V should use as few resources as possible. The first methods introduced to solve this problem were Trotter formula decompositions [Llo96, BACS07] and subsequently methods based on linear combinations of unitaries were developed [CW12] to provide better asymptotic scaling of the cost of simulation.

An alternative strategy was also developed concurrent with these methods that used ideas from quantum walks. Asymptotically, this approach is perhaps the favored method for simulating time-independent Hamiltonians because it is capable of achieving near-optimal scaling with all relevant parameters. The main tool developed for this approach is a walk operator that has eigenvalues $e^{-i\arcsin(E_k/\alpha)}$ where E_k is the k^{th} eigenvalue of H and α is a normalizing parameter. While early work adjusted the spectrum recovering the desired eigenvalues e^{-iE_k} by using phase estimation to invert the arcsin, subsequent work achieved better scaling using linear combination of quantum walk steps [BCK15]. Recently another approach, called qubitization [LC16], was introduced to transform the spectrum in a more efficient manner.

Quantum simulation is not the only field that uses such spectral transformations. Quantum linear systems algorithms [HHL09], as well as algorithms for semi-definite programming [BS17, AGGW17], use these ideas extensively. Earlier work on linear systems used a strategy similar to the quantum walk simulation method: use phase estimation to estimate the eigenvalues of a matrix λ_j and then use quantum rejection sampling to rescale the amplitude of each eigenvector $|\lambda_j\rangle$ via the map $|\lambda_j\rangle \mapsto \lambda_j^{-1}|\lambda_j\rangle$. This enacts the inverse of a matrix and generalizations to the pseudo-inverse are straightforward. More recent methods eschew the use of phase estimation in favor of linear-combination of unitary methods [CKS17] which typically approximate the inversion using a Fourier-series or Chebyshev series expansion. Similar ideas can be used to prepare Gibbs states efficiently [CS17, AGGW17].

These improvements typically result in exponentially improved scaling in terms of precision in various important subroutines. However, since these techniques work on quantum states, and usually one needs to learn certain properties of these states to a specified precision ε, a polynomial dependence on $\frac{1}{\varepsilon}$ is unavoidable. Therefore these improvements typically "only" result in polynomial savings in the complexity. Nevertheless, for complex algorithms this can make a huge difference. These techniques played a crucial role in improving the complexity of quantum semi-definite program solvers [AG18] where the scaling with accuracy was improved from the initial $\mathcal{O}(1/\epsilon^{32})$ to $\mathcal{O}(1/\epsilon^4)$.

We provide a new generalization of qubitization that allows us to view all of the above mentioned applications as a manifestation of a single concept we call *singular value transformation*. The central object for this result is *projected unitary encoding*, which is defined as follows. Suppose that $\widetilde{\Pi}$,Π are orthogonal projectors and U is a unitary, then we say that the unitary U and the projectors $\widetilde{\Pi}$,Π form a projected unitary encoding of the operator $A := \widetilde{\Pi} U \Pi$. We define singular value transformation by a polynomial $P \in \mathbb{C}[x]$ in the following way: if P is an odd polynomial and $A = W\Sigma V^\dagger$ is a singular value decomposition (SVD), then $P^{(SV)}(A) := WP(\Sigma)V^\dagger$, where the polynomial P is applied to the diagonal entries of Σ. Our main result is that for any degree-d odd polynomial $P \in \mathbb{R}[x]$, that is bounded by 1 in absolute value on $[-1, 1]$, we can implement a unitary U_Φ with a simple circuit using U and its inverse a total number of d times such that

$$A = \widetilde{\Pi} U \Pi \Longrightarrow P^{(SV)}(A) = \widetilde{\Pi} U_\Phi \Pi.$$

We prove a similar result for even polynomials as well, but with replacing $\widetilde{\Pi}$ by Π in the above equation, and defining $P^{(SV)}(A) := VP(\Sigma)V^\dagger$ for even polynomials. One can view these results as generalizations of the quantum walk techniques introduced by Szegedy [Sze04].

In order to illustrate the power of this technique we briefly explain some corollaries of this result. For example suppose that U is a quantum algorithm that on the initial state $|0\rangle^{\otimes n}$ succeeds with probability at least p, and indicates success by setting the first qubit to $|1\rangle$. Then we can take $\widetilde{\Pi} := |1\rangle\langle 1| \otimes I_{n-1}$ and $\Pi := |0\rangle\langle 0|^{\otimes n}$. Observe that $A = \widetilde{\Pi} U \Pi$ is a rank-1 matrix having a single non-trivial singular value being the square root of the success probability. If P is an odd polynomial bounded by 1 in absolute value such that P is $\frac{\varepsilon}{2}$-close to 1 on the interval $[\sqrt{p}, 1]$, then by applying singular value transformation we get an algorithm U_Φ that succeeds with probability at least $1 - \varepsilon$. Such a polynomial can be constructed with degree $\mathcal{O}\left(\frac{1}{\sqrt{p}} \log\left(\frac{1}{\varepsilon}\right)\right)$ providing a conceptually simple and efficient implementation of fixed-point amplitude amplification.

It also becomes straightforward to implement the Moore-Penrose pseudoinverse directly. Suppose that $A = W\Sigma V^\dagger$ is an SVD, then the pseudoinverse is simply $A^+ = V\Sigma^{-1}W^\dagger$, where we take the inverse of each non-zero diagonal element of Σ. If we have A represented as a projected unitary encoding, then simply finding an appropriately scaled approximation polynomial of $\frac{1}{x}$ and applying singular value transformation to it implements an approximation of the Moore-Penrose pseudoinverse directly. As an application in quantum machine learning, we design a quantum algorithm for *principal component regression*, and argue that singular value transformation could become a central tool in designing quantum machine learning algorithms.

Based on singular value transformation we develop two main general results: *singular vector transformation*, which maps right singular vectors to left singular vectors, and *singular value threshold projectors*, which project out singular vectors with singular value below a certain threshold. These threshold projectors play a major role in quantum algorithms recently proposed by Kerenidis et al. [KP17b, KL18], and our work fills a minor gap that was present in earlier implementation proposals. Our implementation is also simpler and applies in greater generality than the algorithm of Kerenidis and Prakash [KP17b]. As a useful application of singular value threshold projectors we develop *singular value discrimination*, which decides whether a given quantum state has singular value below or above a certain threshold. As another application we show that using singular vector transformation one can efficiently implement a form of "non-commutative measurement" which is used for preparing ground states of local Hamiltonians. Also we propose a new method for *quantum singular value estimation* introduced by [KP17b].

Other algorithms can also be cast in the singular value transformation framework, including optimal Hamiltonian simulation, robust oblivious amplitude amplification, fast QMA amplification, fast quantum OR lemma and certain quantum walk results. Based on these techniques we also show how to exponentially improve the complexity of implementing fractional queries to unitaries with a gapped spectrum. We summarize in Table 1 the various types of quantum speed-ups that are inherently incorporated in our singular value transformation framework.

Speed-up	Source of speed-up	Examples of algorithms
Exponential	Dimensionality of the Hilbert space	Hamiltonian simulation [Llo96]
	Precise polynomial approximations	Improved HHL algorithm [CKS17]
Quadratic	Singular value = square root of probability	Grover search [Gro96]
	Distinguishability of singular values	Amplitude estimation [BHMT02]
	Singular values close to 1 are more useful	Quantum walks [Sze04]

Table 1: This table gives an intuitive summary of the different types of speed-ups that our singular value transformation framework inherently incorporates. The explanations, examples and the cited papers are far from being complete or representative, the table only intends to give some intuition and illustrate the different sources of speed-ups.

In order to harness the power of singular value transformation one needs to construct projected unitary encodings. A special case of projected unitary encoding is called *block-encoding*, when $\widetilde{\Pi} = \Pi = |0\rangle\langle 0|^{\otimes a} \otimes I$. In this case A is literally the top-left block of the unitary U. In the paper we provide a versatile toolbox for efficiently constructing block-encodings, summarizing recent developments in the field. In particular we demonstrate how to construct block-encodings of unitary matrices, density operators, POVM operators, sparse-access matrices and matrices stored in a QROM[1]. Furthermore, we show how to form linear combinations and products of block-encodings.

1.1 Structure of the paper

In Section 3 we derive a new formalization of qubitization that allows us to view all of the aforementioned applications as a manifestation of a single concept we call "singular value transformation". In Subsection 3.1 we develop a slightly improved version of the quantum signal processing result of Low et al. [LYC16]. In Subsection 3.2 we develop our singular value transformation result based on qubitization ideas of Low and Chuang [LC16]. In Subsection 3.3 we prove bounds about the robustness of singular value transformation. We then introduce singular vector transformation and singular value amplification in Subsection 3.4, from which we provide elementary derivations of fixed-point amplitude amplification and robust oblivious amplitude amplification. We then extend these ideas in Subsection 3.5 to solve the problem of singular value threshold projection and singular value discrimination which as we show allow us to detect and find marked elements in a reversible Markov chain. These ideas then allow us to provide an easy derivation of the quantum linear-systems algorithm, and more generally the quantum least-squares fitting algorithm, in Subsection 3.6. In

[1] By QROM we mean quantum read-only memory, which stores classical data that can be accessed in superposition.

Subsection 3.7 we show how to implement a form of "non-commutative measurement" which is used for preparing ground states of local Hamiltonians, and propose a new method for quantum singular value estimation. Finally, in Subsection 3.8, we design a quantum algorithm for principal component regression, and show how various other machine learning problems can be solved within our framework.

Section 4 shows how to efficiently construct block-encodings and contains a discussion of how these techniques can be employed to perform matrix arithmetic on a quantum computer. In particular we show how to perform basic linear algebra operations on Hamiltonians using block-encodings; we discuss matrix addition and multiplication in Subsections 4.3 and 4.4. We follow this up with a discussion of how arbitrary smooth functions of Hermitian matrices can be performed. We then give an elementary proof of the complexity of block-Hamiltonian simulation in Subsection 5.1 and discuss approximating piecewise smooth functions of Hamiltonians in Subsection 5.2 and present the special cases of Gibbs state preparation and fractional queries in Subsection 5.3. We then conclude by proving lower bounds for implementing functions of Hermitian matrices in Section 6, which in turn implies lower bounds on singular value transformation.

2 Preliminaries and notation

It is well known that for every $A \in \mathbb{C}^{m\times n}$ matrix there exists a pair of unitaries $W \in \mathbb{C}^{m\times m}$, $V \in \mathbb{C}^{n\times n}$ and $\Sigma \in \mathbb{R}^{m\times n}$ such that Σ is a diagonal matrix with non-negative non-increasing entries on the diagonal, and $A = W\Sigma V^\dagger$. Such a decomposition is called *singular value decomposition.* Let $k := \min[m, n]$, then we use $\varsigma_i := \Sigma_{ii}$ for $i \in [k]$ to denote the *singular values* of A, which are the diagonal elements of Σ. The columns of V are called right singular vectors, and the columns of W are called the left singular vectors. In this paper we often define the matrix A as the product of two orthogonal projectors $\widetilde{\Pi}, \Pi$ and unitary U such that $A = \widetilde{\Pi} U \Pi$. In such a case we will assume without loss of generality that the first $\mathrm{rank}(\widetilde{\Pi})$ left singular vectors span $\mathrm{img}(\widetilde{\Pi})$ and the first $\mathrm{rank}(\Pi)$ right singular vectors span $\mathrm{img}(\Pi)$.

The singular value decomposition is not unique if there are multiple singular values with the same value. However, the singular value projectors are uniquely determined, see, e.g., Gilyén and Sattath [GS17].

Definition 1 (Singular value projectors)**.** *Let $A = W\Sigma V^\dagger$ be a singular value decomposition. Let Σ_ς be the matrix that we get from Σ by replacing all singular values that have value $= \varsigma$ by 1 and replacing all $\neq \varsigma$ singular values by 0. Then we define the right singular value projector to singular value ς as $V\Sigma_\varsigma V^\dagger$, and define the left singular value projector to singular value ς as $W\Sigma_\varsigma W^\dagger$ projecting orthogonally to the subspace spanned by the corresponding singular vectors. For a set $S \subset \mathbb{R}$ we similarly define the right and left singular value projectors $V\Sigma_S V^\dagger$, $W\Sigma_S W^\dagger$ projecting orthogonally to the subspace spanned by the singular vectors having singular value in S.*

In this paper we will work with polynomial approximations, and therefore we introduce some related notation. For a function $f : I \to \mathbb{C}$ and a subset $I' \subseteq I$ we use the notation $\|f\|_{I'} := \sup_{x\in I'} |f(x)|$ to denote the sup-norm of the function f on the domain I'. We say that a function $f : \mathbb{R} \to \mathbb{C}$ *is even* if for all $x \in \mathbb{R}$ we have $f(-x) = f(x)$, and that it *is odd* if for all $x \in \mathbb{R}$ we have $f(-x) = -f(x)$.

Let $P \in \mathbb{C}[x]$ be a complex polynomial $P(x) = \sum_{j=0}^{k} a_j x^j$, then we denote by $P^*(x) := \sum_{j=0}^{k} a_j^* x^j$ the polynomial with conjugated coefficients, and let $\Re[P](x) := \sum_{j=0}^{k} \Re[a_j] x^j$ denote the real polynomial we get by taking the real part of the coefficients. We say that P *is even* if all coefficients corresponding to odd powers of x are 0, and similarly we say that P *is odd* if all coefficients corresponding to even powers of x are 0. For an integer number $z \in \mathbb{Z}$ we say that P has parity z if z is even and P is even or z is odd and P is odd. We will denote by $T_d \in \mathbb{R}[x]$ the d-th Chebyshev polynomial of the first kind, defined by $T_d(x) := \cos(d \arccos(x))$.

Whenever we present a matrix and put a . in some place we mean a matrix with arbitrary values of the elements in the unspecified block. For example [.] just denotes a matrix with completely arbitrary elements, similarly

$$U = \begin{bmatrix} A & . \\ . & . \end{bmatrix}$$

denotes an arbitrary matrix whose top-left block is A.

For an orthogonal projector Π we will frequently use the Π-controlled NOT gate, denoted by $\mathrm{C}_\Pi\mathrm{NOT}$, which implements a coherent measurement operator by flipping the value of a qubit based on whether the state of a register is in the image of Π or not. For example if $\Pi = |1\rangle\langle 1|$, then we just get back the usual CNOT gate controlled by the second qubit.

Definition 2 ($\mathrm{C}_\Pi\mathrm{NOT}$ gate)**.** *For an orthogonal projector* Π *let us define the* Π*-controlled NOT gate as the unitary operator*

$$\mathrm{C}_\Pi\mathrm{NOT} := X \otimes \Pi + I \otimes (I - \Pi).$$

3 Qubitization and Singular value transformations

The methods in this section are based on the so called "Quantum Signal Processing" techniques introduced by Low, Yoder and Chuang [LYC16]. In Section 3.1 we present a self-contained treatment of these techniques, significantly streamline the formalism, and develop slightly improved versions of the results presented in [LYC16]. As a corollary of the results we also develop Corollary 8-10, which will be the only results that we need in the rest of the paper. We suggest the first-time reader to skip the proofs in Section 3.1, as they are not necessary in order to understand the main ideas of Sections 3.2-3.6.

In Sections 3.2 we show how to leverage the results of Section 3.1 to perform *singular value transformation* of projected unitary matrices, with ideas coming from "qubitization" [LC16]. Singular value transformation is a common generalization of the techniques developed around qubitization, based on which we can quickly derive a host of well-optimized applications in Sections 3.4-3.8.

3.1 Parametrized SU(2) unitaries induced by Pauli rotations

In this section we review the results of Low, Yoder and Chuang [LYC16], who show how to build 2×2 unitary matrices whose entries are trigonometric polynomials by taking products of various rotation and phase gates. They consider essentially the following problem, which they call "Quantum Signal Processing": suppose one can apply a gate sequence

$$e^{i\phi_0\sigma_z} e^{i\theta\sigma_x} e^{i\phi_1\sigma_z} e^{i\theta\sigma_x} e^{i\phi_2\sigma_z} \cdot \ldots \cdot e^{i\theta\sigma_x} e^{i\phi_k\sigma_z}, \tag{1}$$

where θ is unknown (they call $e^{i\theta\sigma_x}$ the signal unitary) but one has control over the angles $\varphi_0, \varphi_1, \ldots, \varphi_k$; which unitary operators can we build this way? They give a characterization of the unitary operators that can be constructed this way, and find that the set of achievable unitary operators is quite rich.

We find it more useful to work with the above matrices using a slightly modified parametrization. For $x\in[-1,1]$ let us define

$$W(x):=\begin{bmatrix} x & i\sqrt{1-x^2}\\ i\sqrt{1-x^2} & x\end{bmatrix}=e^{i\arccos(x)\sigma_x}.$$

It is easy to see that if $\theta\in[0,\pi]$, then by setting $x:=\cos(\theta)$ Eq. (1) can be rewritten as

$$e^{i\phi_0\sigma_z}W(x)e^{i\phi_1\sigma_z}W(x)e^{i\phi_2\sigma_z}\cdot\ldots\cdot W(x)e^{i\phi_k\sigma_z}. \tag{2}$$

Now we present the characterization of Low et al. [LYC16] using the above formalism. Our formulation makes the statement simpler and reduces the number of cases. We also present a succinct simplified proof which can be conveniently described using our formalism.

Theorem 3. *Let $k\in\mathbb{N}$; there exists $\Phi=\{\phi_0,\phi_1,\ldots,\phi_k\}\in\mathbb{R}^{k+1}$ such that for all $x\in[-1,1]$:*

$$e^{i\phi_0\sigma_z}\prod_{j=1}^{k}\left(W(x)e^{i\phi_j\sigma_z}\right)=\begin{bmatrix} P(x) & iQ(x)\sqrt{1-x^2}\\ iQ^*(x)\sqrt{1-x^2} & P^*(x)\end{bmatrix} \tag{3}$$

if and only if $P,Q\in\mathbb{C}[x]$ such[2] *that*

(i) $\deg(P)\leq k$ *and* $\deg(Q)\leq k-1$

(ii) P *has parity-*$(k \mod 2)$ *and* Q *has parity-*$(k-1 \mod 2)$

(iii) $\forall x\in[-1,1]\colon |P(x)|^2+(1-x^2)|Q(x)|^2=1.$

Proof. "$\Longrightarrow$": For the $k=0$ case the unitary on the left hand side of (3) is $e^{i\phi_0\sigma_z}$, so that $P\equiv e^{i\phi_0}$ and $Q\equiv 0$ satisfy the properties (i)-(iii). Now we prove (i)-(ii) by induction. The induction step can be shown as follows: suppose we have proved for $k-1$ that

$$e^{i\phi_0\sigma_z}\prod_{j=1}^{k-1}\left(W(x)e^{i\phi_j\sigma_z}\right)=\begin{bmatrix} \tilde{P}(x) & i\tilde{Q}(x)\sqrt{1-x^2}\\ i\tilde{Q}^*(x)\sqrt{1-x^2} & \tilde{P}^*(x)\end{bmatrix},$$

where $\tilde{P},\tilde{Q}\in\mathbb{C}[x]$ satisfy (i)-(ii). Then

$$\begin{aligned} e^{i\phi_0\sigma_z}\prod_{j=1}^{k}\left(W(x)e^{i\phi_j\sigma_z}\right)&=\begin{bmatrix} \tilde{P}(x) & i\tilde{Q}(x)\sqrt{1-x^2}\\ i\tilde{Q}^*(x)\sqrt{1-x^2} & \tilde{P}^*(x)\end{bmatrix}\begin{bmatrix} e^{i\phi_k}x & ie^{-i\phi_k}\sqrt{1-x^2}\\ ie^{i\phi_k}\sqrt{1-x^2} & e^{-i\phi_k}x\end{bmatrix}\\ &=\begin{bmatrix} \overbrace{e^{i\phi_k}\left(x\tilde{P}(x)+(x^2-1)\tilde{Q}(x)\right)}^{P(x):=} & ie^{-i\phi_k}\left(x\tilde{Q}(x)+\tilde{P}(x)\right)\sqrt{1-x^2}\\ \underbrace{ie^{i\phi_k}\left(x\tilde{Q}^*(x)+\tilde{P}^*(x)\right)}_{Q^*(x):=}\sqrt{1-x^2} & e^{-i\phi_k}\left(x\tilde{P}^*(x)+(x^2-1)\tilde{Q}^*(x)\right)\end{bmatrix}, \end{aligned} \tag{4}$$

[2] Note that the value of $P(x)$ is only determined for $x\in[-1,1]$ and $Q(x)$ for $x\in(-1,1)$; thus more precisely we should talk about the polynomial functions induced by $P(x)|_{[-1,1]}\in\mathbb{C}[x]$ and $Q(x)|_{(-1,1)}\in\mathbb{C}[x]$.

and it is easy to see that P, Q satisfy (i)-(ii). Finally note that the left hand side of (3) is a product of unitaries, therefore the right hand side is unitary too, which implies (iii).

"$\Longleftarrow$": Suppose P, Q satisfy (i)-(iii). First we handle a trivial case: suppose that $\deg(P) = 0$, then due to (iii) we must have that $|P(1)| = 1$ and thus $P \equiv e^{i\phi_0}$ for some $\phi_0 \in \mathbb{R}$. This again using (iii) implies that $Q \equiv 0$. Due to (ii) we must have that k is even, and thus $\Phi = (\phi_0, \frac{\pi}{2}, -\frac{\pi}{2}, \ldots, \frac{\pi}{2}, -\frac{\pi}{2}) \in \mathbb{R}^{k+1}$ is a solution, since

$$e^{i\phi_0\sigma_z} \prod_{j=1}^{k/2} \left(W(x) e^{i\frac{\pi}{2}\sigma_z} W(x) e^{-i\frac{\pi}{2}\sigma_z} \right) = e^{i\phi_0\sigma_z} = \begin{bmatrix} e^{i\phi_0} & 0 \\ 0 & e^{-i\phi_0} \end{bmatrix}.$$

This special case also covers the $k = 0$ case, providing the base of our induction.

Now we show the induction step, assuming that we proved the claim for $k-1$. Note that (iii) can be rewritten as

$$\forall x \in [-1,1]\colon P(x)P^*(x) + (1-x^2)Q(x)Q^*(x) = 1. \tag{5}$$

Since this equation holds for infinitely many points, the polynomial on the right hand side of (5) must be the constant $\equiv 1$ polynomial. Assume without loss of generality that $1 \leq \deg(P) = \ell \leq k$, then we must have that $\deg(Q) = \ell - 1$, and $|p_\ell| = |q_{\ell-1}|$, since the highest order terms cancel each other in (5). Let $\phi_k \in \mathbb{R}$ be such that $e^{2i\phi_k} = \frac{p_\ell}{q_{\ell-1}}$, and let us define

$$\begin{aligned}
\begin{bmatrix} \tilde{P}(x) & i\tilde{Q}(x)\sqrt{1-x^2} \\ i\tilde{Q}^*(x)\sqrt{1-x^2} & \tilde{P}^*(x) \end{bmatrix} :=& \begin{bmatrix} P(x) & iQ(x)\sqrt{1-x^2} \\ iQ^*(x)\sqrt{1-x^2} & P^*(x) \end{bmatrix} e^{-i\phi_k\sigma_z} W^\dagger(x) \\
=& \begin{bmatrix} P(x) & iQ(x)\sqrt{1-x^2} \\ iQ^*(x)\sqrt{1-x^2} & P^*(x) \end{bmatrix} \begin{bmatrix} e^{-i\phi_k}x & -ie^{-i\phi_k}\sqrt{1-x^2} \\ -ie^{i\phi_k}\sqrt{1-x^2} & e^{i\phi_k}x \end{bmatrix} \\
=& \begin{bmatrix} \overbrace{e^{-i\phi_k}xP(x) + e^{i\phi_k}(1-x^2)Q(x)}^{\tilde{P}(x):=} & i\tilde{Q}(x)\sqrt{1-x^2} \\ \underbrace{i\left(e^{-i\phi_k}xQ^*(x) - e^{i\phi_k}P^*(x)\right)}_{\tilde{Q}^*(x):=}\sqrt{1-x^2} & \tilde{P}^*(x) \end{bmatrix}
\end{aligned} \tag{6}$$

where

$$\tilde{P}(x) = e^{-i\phi_k}xP(x) + e^{i\phi_k}(1-x^2)Q(x) = e^{-i\phi_k}\left(xP(x) + \frac{p_\ell}{q_{\ell-1}}(1-x^2)Q(x) \right) \tag{7}$$

and

$$\tilde{Q}(x) = e^{i\phi_k}xQ(x) - e^{-i\phi_k}P(x) = e^{-i\phi_k}\left(\frac{p_\ell}{q_{\ell-1}}xQ(x) - P(x) \right). \tag{8}$$

It is easy to see that the highest order terms in (7)-(8) cancel out, and therefore $\deg(\tilde{P}) \leq \ell - 1 \leq k-1$, $\deg(\tilde{Q}) \leq \ell - 2 \leq k-2$. Using (7)-(8) we can also verify that $\tilde{P}, \tilde{Q}$ satisfy (i)-(ii) regarding $k-1$, moreover property (iii) is preserved due to unitarity. So by the induction hypothesis we get that (6) equals $e^{i\tilde{\phi}_0\sigma_z}\left(\prod_{j=1}^{k-1} W(x)e^{i\tilde{\phi}_j\sigma_z}\right)$ for some $\tilde{\Phi} \in \mathbb{R}^k$, therefore $\Phi := (\tilde{\phi}_0, \tilde{\phi}_1, \tilde{\phi}_2, \ldots, \tilde{\phi}_{k-1}, \phi_k) \in \mathbb{R}^{k+1}$ is a solution. □

Note that the above proof also gives an algorithm that finds Φ using $\mathcal{O}(k^2)$ arithmetic operations. The following two characterizations and their proofs also follow a constructive approach which can be translated to a polynomial time algorithm. However, they have the drawback that they rely on finding roots of high-degree polynomials,[3] which makes it harder in practice to execute the resulting protocols.

Theorem 4. *Let $k \in \mathbb{N}$ be fixed. Let $P \in \mathbb{C}[x]$, there exists some $Q \in \mathbb{C}[x]$ such that P, Q satisfy properties (i)-(iii) of Theorem 3 if and only if P satisfies properties (i)-(ii) of Theorem 3 and*

(iv.a) $\forall x \in [-1,1]\colon |P(x)| \leq 1$

(iv.b) $\forall x \in (-\infty,-1] \cup [1,\infty)\colon |P(x)| \geq 1$

(iv.c) if k is even, then $\forall x \in \mathbb{R}\colon P(ix)P^*(ix) \geq 1$.

Similarly, let $Q \in \mathbb{C}[x]$, there exists some $P \in \mathbb{C}[x]$ such that P, Q satisfy properties (i)-(iii) of Theorem 3 if and only if Q satisfies properties (i)-(ii) of Theorem 3 and

(v.a) $\forall x \in [-1,1]\colon \sqrt{1-x^2}|Q(x)| \leq 1$

(v.b) if k is odd, then $\forall x \in \mathbb{R}\colon (1+x^2)Q(ix)Q^*(ix) \geq 1$.

Proof. "$\Longrightarrow$": Trivially follows from (iii):

$$\forall x \in \mathbb{C}\colon P(x)P^*(x) + (1-x^2)Q(x)Q^*(x) = 1.$$

"$\Longleftarrow$": First consider the case when k is odd, and consider the polynomial $A(x) := 1 - P(x)P^*(x)$. Note that $A \in \mathbb{R}[x]$ and A is even, therefore A is in fact a polynomial in x^2. Let $y = x^2$ and consider the real polynomial $\tilde{A}(y) := A(\sqrt{y})$. Observe that $\forall y \geq 1\colon \tilde{A}(y) \leq 0$ due to (iv.b), $\forall y \in [0,1]\colon \tilde{A}(y) \geq 0$ due to (iv.a) and $\forall y \leq 0\colon \tilde{A}(y) \geq 1$ since

$$\begin{aligned}
\tilde{A}(y) &= A(i\sqrt{-y}) && (y \leq 0)\\
&= 1 - P(i\sqrt{-y})P^*(i\sqrt{-y}) = 1 + P(i\sqrt{-y})P^*(-i\sqrt{-y}) && (P \text{ is odd})\\
&= 1 + P(i\sqrt{-y})(P(i\sqrt{-y}))^* = 1 + |P(i\sqrt{-y})|^2 \geq 1.
\end{aligned}$$

Therefore all real roots have even multiplicity except for 1, moreover all complex roots come in pairs. Thus $\tilde{A}(y) = (1-y)K^2\prod_{s\in S}(y-s)(y-s^*)$ for some $K \in \mathbb{R}$ and $S \subseteq \mathbb{C}$ multiset of roots. Let $W(y) := K\prod_{s\in S}(y-s) \in \mathbb{C}[y]$, then $\tilde{A}(y) = (1-y)W(y)W^*(y)$, and thus $A(x) = (1-x^2)W(x^2)W^*(x^2)$, i.e., $1 = P(x)P^*(x) + (1-x^2)W(x^2)W^*(x^2)$. Setting $Q(x) := W(x^2)$ concludes this case.

The other cases can be proven similarly, by examining the polynomial $1 - P(x)P^*(x)$ or $1-(1-x^2)Q(x)Q^*(x)$ respectively. □

The original proof of the next theorem in [LYC16] used the Weierstrass substitution, which made it difficult to understand, and made it hard to analyze the numerical stability of the induced algorithm. Also the theorem was stated in a slightly less general form requiring $\Re[\tilde{P}](1) = 1$. We roughly follow the approach of [LYC16], but improve all of the mentioned aspects of the theorem and its proof, while making the statement and the proof conceptually simpler.

[3] For a good bound on the complexity of approximate root finding see, e.g., the work of Neff and Reif [NR96].

Theorem 5. *Let $k \in \mathbb{N}$ be fixed. Let $\tilde{P}, \tilde{Q} \in \mathbb{R}[x]$, there exists some $P, Q \in \mathbb{C}[x]$ satisfying properties (i)-(iii) of Theorem 3 such that $\tilde{P} = \Re[P]$, $\tilde{Q} = \Re[Q]$, if and only if $\tilde{P}, \tilde{Q}$ satisfy properties (i)-(ii) of Theorem 3 and*

(vi) $\forall x \in [-1,1]\colon \tilde{P}(x)^2 + (1-x^2)\tilde{Q}^2 \leq 1.$

(Note that the same holds if we replace $\Re[P]$ by $\Im[P]$ and/or $\Re[Q]$ by $\Im[Q]$ in the statement. Moreover we may set $\tilde{Q} \equiv 0$ or $\tilde{P} \equiv 0$ if we are only interested in $\tilde{P}$ or $\tilde{Q}$.)

Proof. "$\Longrightarrow$": Trivial.
"$\Longleftarrow$": Apply Lemma 6 to the polynomial $1 - \tilde{P}(x)^2 - (1-x^2)\tilde{Q}(x)^2$, and set $P := \tilde{P} + iB$, $Q := \tilde{Q} + iC$. □

Lemma 6. *Suppose that $A \in \mathbb{R}[x]$ is an even polynomial such that $\deg(A) \leq 2k$ and for all $x \in [-1,1]$ we have $A(x) \geq 0$. Then there exist polynomials $B, C \in \mathbb{R}[x]$ such that $A(x) = B^2(x) + (1-x^2)C^2(x)$, moreover $\deg(B) \leq k$, $\deg(C) \leq k-1$, B has parity-$(k \mod 2)$ and C has parity-$(k-1 \mod 2)$.*

Proof. If $A = 0$ the statement is trivial, so we assume in the rest that $A \neq 0$. Let S be the multiset of roots, containing the roots of A with their algebraic multiplicity. Note that if $s \in S$ then also $-s \in S$ and $s^* \in S$ since A is an even real polynomial. (This statement holds considering multiplicities.) Let us introduce the following subsets of S (these are again multisets):

$$\begin{aligned}
S_0 &:= \{s \in S\colon s = 0\}\\
S_{(0,1)} &:= \{s \in S\colon s \in (0,1)\}\\
S_{[1,\infty)} &:= \{s \in S\colon s \in [1,\infty)\}\\
S_I &:= \{s \in S\colon \mathrm{Re}(s) = 0 \,\&\, \mathrm{Im}(s) > 0\}\\
S_C &:= \{s \in S\colon \mathrm{Re}(s) > 0 \,\&\, \mathrm{Im}(s) > 0\}.
\end{aligned}$$

Using the roots in S and some scaling factor $K \in \mathbb{R}_+$ we can write

$$A(x) = K^2 x^{|S_0|} \prod_{s \in S_{(0,1)}} (x^2 - s^2) \prod_{s \in S_{[1,\infty)}} (s^2 - x^2) \prod_{s \in S_I} (x^2 + |s|^2) \prod_{(a+bi) \in S_C} \left(x^4 + 2x^2(b^2 - a^2) + (a^2+b^2)^2\right). \tag{9}$$

Consider the following rearrangement of the above terms corresponding to the roots in $S_{[1,\infty)}, S_I, S_C$:

$$s^2 - x^2 = (s^2-1)x^2 + s^2(1-x^2) = \underbrace{\left(\sqrt{(s^2-1)}x + is\sqrt{1-x^2}\right)}_{R_{(s)}(x):=} R^*_{(s)}(x) \tag{10}$$

$$x^2 + |s|^2 = (|s|^2+1)x^2 + |s|^2(1-x^2) = \underbrace{\left(\sqrt{(|s|^2+1)}x + i|s|\sqrt{1-x^2}\right)}_{P_{(s)}(x):=} P^*_{(s)}(x) \tag{11}$$

$$x^4 + 2x^2(b^2-a^2) + (a^2+b^2)^2 = \underbrace{\left(\left(cx^2 - (a^2+b^2)\right) + i\sqrt{c^2-1}x\sqrt{1-x^2}\right)}_{Q_{(a,b)}(x):=} Q^*_{(a,b)}(x), \tag{12}$$

$$\text{where}^{4}\ c = a^2 + b^2 + \sqrt{2(a^2+1)b^2 + (a^2-1)^2 + b^4}.$$

Let us define

$$W(x) := Kx^{|S_0|/2} \prod_{s\in S_{(0,1)}} \sqrt{(x^2-s^2)} \prod_{s\in S_{[1,\infty)}} R_s(x) \prod_{s\in S_I} P_s(x) \prod_{(a+bi)\in S_C} Q_{(a,b)}(x).$$

Note that the factor $x^{|S_0|/2}\prod_{s\in S_{(0,1)}} \sqrt{(x^2-s^2)}$ is a polynomial, since every root in S_0 and $S_{(0,1)}$ has even multiplicity as $A(x) \geq 0$ for all $x \in (-1,1)$. Also note that $W(x)$ is a product of expressions of the form $B'(x) + i\sqrt{1-x^2}C'(x)$ where $B', C' \in \mathbb{R}[x]$ are polynomials having opposite parities (n.b. the zero polynomial is both even and odd, thus it has opposite parity to any even/odd polynomial). Since the product of expressions of such form can again be written in such a form, we have that $W(x) = B(x) + i\sqrt{1-x^2}C(x)$ for some $B, C \in \mathbb{R}[x]$ having opposite parities. Also note that $\deg(B) \leq |S|/2$ and $\deg(C) \leq |S|/2 - 1$.

Finally observe that by (9)-(12) we have that $A(x) = W(x) \cdot W^*(x)$, thus $A(x) = B(x)^2 + (1-x^2)C(x)^2$. Since $\deg(B) \leq |S|/2 \leq k$ and $\deg(C) \leq |S|/2 - 1 \leq k-1$, in case $\deg(A) = 2k$, we must have that B has parity-($k \mod 2$) and C has parity-($k-1 \mod 2$). If $\deg(A) \leq 2k-2$ and B has parity-($k-1 \mod 2$), then consider $\tilde{W}(x) := W(x) \cdot \left(x + i\sqrt{1-x^2}\right)$. Since $\left(x + i\sqrt{1-x^2}\right)\left(x + i\sqrt{1-x^2}\right)^* = 1$ we still have that $\tilde{W}(x)\tilde{W}^*(x) = A(x)$. Now let us denote $\tilde{W}(x) = \tilde{B}(x) + i\sqrt{1-x^2}\tilde{C}(x)$, then we get that $A(x) = \tilde{B}(x)^2 + (1-x^2)\tilde{C}(x)^2$, moreover $\deg(\tilde{B}) \leq k$, $\deg(\tilde{C}) \leq k-1$, $\tilde{B}$ has parity-($k \mod 2$) and $\tilde{C}$ has parity-($k-1 \mod 2$). □

Note that the proofs of Theorems 3-5 are constructive, therefore they also give algorithms to find P, Q and Φ. The most difficult step in the proofs is to find the roots of a given degree-d univariate complex polynomial. This problem is fortunately well studied, and can be solved up to ε precision on a classical computer in time $\mathcal{O}(\mathrm{poly}(d, \log(1/\varepsilon)))$.

Now we prove a corollary of the above result where we replace the $W(x)$ rotation operators with the following $R(x)$ reflection gates, which fit the block-encoding formalism nicer.

Definition 7 (Parametrized family of single qubit reflections). *We define a parametrized family of single qubit reflection operators for all $x \in [-1,1]$ such that*

$$R(x) := \begin{bmatrix} x & \sqrt{1-x^2} \\ \sqrt{1-x^2} & -x \end{bmatrix}. \tag{13}$$

Corollary 8 (Quantum signal processing using reflections). *Let $P \in \mathbb{C}[x]$ be a degree-d polynomial, such that*

- *P has parity-($d \mod 2$),*
- *$\forall x \in [-1,1]\colon |P(x)| \leq 1$,*
- *$\forall x \in (-\infty,-1] \cup [1,\infty)\colon |P(x)| \geq 1$,*
- *if d is even, then $\forall x \in \mathbb{R}\colon P(ix)P^*(ix) \geq 1$.*

[4] Observe that $c \geq 1$ for all $a, b \geq 0$ and thus $\sqrt{c^2-1} \in \mathbb{R}$.

Then there exists $\Phi \in \mathbb{R}^d$ such that[5]

$$\prod_{j=1}^{d}\Big(e^{i\phi_j\sigma_z}R(x)\Big) = \begin{bmatrix} P(x) & \cdot \\ \cdot & \cdot \end{bmatrix}. \tag{14}$$

Moreover for $x \in \{-1,1\}$ we have that $P(x) = x^d \prod_{j=1}^{d} e^{i\phi_j}$, and for d even $P(0) = e^{-i\sum_{j=1}^{d}(-1)^j\phi_j}$.

Proof. By Theorem 4 we have that there exists $\Phi' \in \mathbb{R}^{d+1}$ for some $d \geq 1$ such that

$$e^{i\phi'_0\sigma_z}\left(\prod_{j=1}^{d} W(x)e^{i\phi'_j\sigma_z}\right) = \begin{bmatrix} P(x) & \cdot \\ \cdot & \cdot \end{bmatrix}. \tag{15}$$

Observe that

$$W(x) = ie^{-i\frac{\pi}{4}\sigma_z}R(x)e^{i\frac{\pi}{4}\sigma_z}, \tag{16}$$

thus the left-hand-side of (14) equals

$$e^{i\phi'_0\sigma_z}\left(\prod_{j=2}^{d} e^{i\phi_j\sigma_z}ie^{-i\frac{\pi}{4}\sigma_z}R(x)e^{i\frac{\pi}{4}\sigma_z}\right) = i^d e^{i(\phi'_0-\frac{\pi}{4})\sigma_z}R(x)\left(\prod_{j=2}^{d} e^{i(\phi'_{j-1}-\frac{\pi}{2})\sigma_z}R(x)\right)e^{i(\phi'_d-\frac{\pi}{4})}.$$

Therefore

$$e^{i(\phi'_0+\phi'_d+(d-1)\frac{\pi}{2})}R(x)\left(\prod_{j=2}^{d} e^{i(\phi'_{j-1}-\frac{\pi}{2})\sigma_z}R(x)\right) = \begin{bmatrix} P(x) & \cdot \\ \cdot & \cdot \end{bmatrix}.$$

So choosing $\phi_1 := \phi'_0 + \phi'_d + (d-1)\frac{\pi}{2}$ and for all $j \in \{2,3,\ldots,d\}$ setting $\phi_j := \phi'_{j-1} - \frac{\pi}{2}$, results in a $\Phi \in \mathbb{R}^d$ that clearly satisfies (14). The additional result for $x \in \{-1,1\}$ follows from the fact that for $x \in \{-1,1\}$ every matrix in (14) becomes diagonal.

The claim about $P(0)$ follows from the observation that

$$e^{i\phi_1\sigma_z}R(0)e^{i\phi_2\sigma_z}R(0) = e^{i(\phi_1-\phi_2)\sigma_z}.$$

□

The above requirements on P are not very intuitive, but fortunately we have a good understanding of the polynomials that can emerge by taking the real part of the above complex polynomials. Before stating the corresponding corollary, we note that Chebyshev polynomials satisfy the above requirements. One can prove it directly, but instead of doing so we just explicitly describe[6] the corresponding Φ.

Lemma 9 (Constructing Chebyshev polynomials via quantum signal processing)**.** *Let $T_d \in \mathbb{R}[x]$ be the d-th Chebyshev polynomial of the first kind. Let $\Phi \in \mathbb{R}^d$ be such that $\phi_1 = (1-d)\frac{\pi}{2}$, and for all $i \in [d] \setminus \{1\}$ let $\phi_i := \frac{\pi}{2}$. Using this Φ in equation (14) we get that $P = T_d$.*

Proof. One can prove this, e.g., by induction using the substitution $x := \cos(\theta)$. □

[5] Note that the $e^{i\phi_1\sigma_z}$ gate can in fact be replaced by a simple phase gate $e^{i\phi_1}$.

[6] By Theorem 4 it actually proves that the conditions of Corollary 8 hold for Chebyshev polynomials.

Corollary 10. (Real quantum signal processing) *Let $P_{\Re}(x) \in \mathbb{R}[x]$ be a degree-d polynomial for some $d \geq 1$, such that*

- *$P_{\Re}$ has parity-($d \mod 2$), and*
- *for all $x \in [-1,1]$: $|P_{\Re}(x)| \leq 1$.*

Then there exists $P \in \mathbb{C}[x]$ that satisfies the requirements of Corollary 8.

Moreover, given $P_{\Re}(x)$ and $\delta \geq 0$ we can find a P and a corresponding Φ, such that $|\Re[P]-P_{\Re}| \leq \delta$ for all $x \in [-1,1]$, using a classical computer in time $\mathcal{O}(\mathrm{poly}(d, \log(1/\delta)))$.

Proof. The existence of such P follows directly from Theorem 3-5.

The complexity statement follows from the fact that we can find P and Φ' using the procedures of Theorems 3-5 on a classical computer in time $\mathcal{O}(\mathrm{poly}(d, \log(1/\varepsilon)))$ as noted above. Computing Φ from Φ' as in the proof of Corollary 8 only yields a small overhead. □

3.2 Singular value transformation by qubitization

Qubitization is a technique introduced by Low and Chuang [LC16] in order to apply polynomial transformations to the spectrum of a Hermitian (or normal) operator, which is represented as the top-left block of a unitary matrix. They also showed how to use their techniques in order to develop advanced amplitude amplification techniques. In this section we generalize their results, and develop the technique of singular value transformation, which applies to any operator as opposed to only normal operators.

It turns out that by applying a unitary U back and forth interleaved with some simple phase operators one can induce polynomial transformations to the singular values of a particular (not necessarily rectangular) block-matrix of the unitary U. The main idea behind the qubitization approach is to lift the quantum signal processing results presented in the previous section. One can do so by defining some two-dimensional invariant subspaces within which the results of quantum signal processing apply, thereby "qubitizing"[7] the problem. Then by understanding how the two-dimensional subspaces behave, one can infer the higher-dimensional behavior.

The original qubitization approach can be understood along the lines of C. Jordan's Lemma [Jor75] about the common invariant subspaces of two reflections.[8] Jordan's result is most often presented stating that the product of two reflections decomposes to one- and two-dimensional invariant subspaces, such that the operator has eigenvalue ±1 on the one-dimensional subspaces, and the operator acts as a rotation on the two-dimensional subspaces. This higher dimensional insight lies at the heart of Szegedy's quantum walk results [Sze04] as well as Marriott and Watrous' QMA amplification scheme [MW05].

Motivated by a series of prior work on quantum search algorithms [Gro05, Hø00, YLC14] the original qubitization approach of Low and Chuang [LC16] replaced one of the reflections in Jordan's Lemma by a phase-gate, such as in Figure 1b. They examined the operators arising by iterative application of the reflection- and phase-operator with applying possibly different phases in each step. In this paper we go one step further and replace the other reflection[9] by an arbitrary unitary

[7] Another justification for the term "qubitization" is that the involved higher-dimensional phase operations reduce to carefully choosing a single qubit phase gate, see Figure 1b.

[8] By reflection we mean a Hermitian operator having only ±1 eigenvalues, possibly having multiple −1 eigenvalues.

[9] One could also merge U into one of the projectors, leading to a product of reflections as in Jordan's Lemma [Jor75].

operator U, and analyze the procedure with carefully chosen one- and two-dimensional subspaces coming from singular value decomposition.

Definition 11 (Singular value decomposition of a projected unitary)**.** *Let $\mathcal{H}_U$ be a finite-dimensional Hilbert space and let $U, \Pi, \widetilde{\Pi} \in \mathrm{End}(\mathcal{H}_U)$ be linear operators on $\mathcal{H}_U$ such that U is a unitary, and $\Pi, \widetilde{\Pi}$ are orthogonal projectors, and let*

$$A = \widetilde{\Pi} U \Pi.$$

Let $d := \mathrm{rank}(\Pi)$, $\tilde{d} := \mathrm{rank}\Big(\widetilde{\Pi}\Big)$, $d_{\min} := \min(d, \tilde{d})$. By singular value decomposition we know that there exist orthonormal bases $(|\psi_i\rangle \colon i \in [d])$, $\Big(|\tilde{\psi}_i\rangle \colon i \in [\tilde{d}]\Big)$ of the subspaces $\mathrm{img}(\Pi)$ and $\mathrm{img}\Big(\widetilde{\Pi}\Big)$ respectively, such that

$$A = \sum_{i=1}^{d_{\min}} \varsigma_i |\tilde{\psi}_i\rangle\langle\psi_i|, \tag{17}$$

and[10] *for all $i \in [d_{\min}] \colon \varsigma_i \in \mathbb{R}_0^+$. Moreover, $\varsigma_i \geq \varsigma_j$ for all $i \leq j \in [d_{\min}]$.*

Definition 12 (Invariant subspaces associated to a singular value decomposition)**.** *Let $U, \Pi, \widetilde{\Pi}, A$ be as in Definition 11, and let us use its notation. Let $k \in [d_{\min}]$ be the largest index for which $\varsigma_k = 1$, and let $r = \mathrm{rank}(A)$. For*

$$\begin{aligned}
&i \in [k] \textit{ let} && \mathcal{H}_i := \mathrm{Span}(|\psi_i\rangle) \textit{ and} && \tilde{\mathcal{H}}_i := \mathrm{Span}\Big(|\tilde{\psi}_i\rangle\Big),\\
&i \in [r] \setminus [k] \textit{ let} && \mathcal{H}_i := \mathrm{Span}\Big(|\psi_i\rangle, |\psi_i^\perp\rangle\Big) \textit{ where} && |\psi_i^\perp\rangle := \frac{(I-\Pi)U^\dagger|\tilde{\psi}_i\rangle}{\Big\|(I-\Pi)U^\dagger|\tilde{\psi}_i\rangle\Big\|} = \frac{(I-\Pi)U^\dagger|\tilde{\psi}_i\rangle}{\sqrt{1-\varsigma_i^2}},\\
&i \in [r] \setminus [k] \textit{ let} && \tilde{\mathcal{H}}_i := \mathrm{Span}\Big(|\tilde{\psi}_i\rangle, |\tilde{\psi}_i^\perp\rangle\Big) \textit{ where} && |\tilde{\psi}_i^\perp\rangle := \frac{(I-\widetilde{\Pi})U|\psi_i\rangle}{\Big\|(I-\widetilde{\Pi})U|\psi_i\rangle\Big\|} = \frac{(I-\widetilde{\Pi})U|\psi_i\rangle}{\sqrt{1-\varsigma_i^2}},\\
&i \in [d] \setminus [r] \textit{ let} && \mathcal{H}_i^R := \mathrm{Span}(|\psi_i\rangle) \textit{ and} && \tilde{\mathcal{H}}_i^R := \mathrm{Span}(U|\psi_i\rangle),\\
&i \in [\tilde{d}] \setminus [r] \textit{ let} && \mathcal{H}_i^L := \mathrm{Span}\Big(U^\dagger|\tilde{\psi}_i\rangle\Big) \textit{ and} && \tilde{\mathcal{H}}_i^L := \mathrm{Span}\Big(|\tilde{\psi}_i\rangle\Big).
\end{aligned}$$

Finally let

$$\mathcal{H}_\perp := \left(\bigoplus_{i\in[r]} \mathcal{H}_i \oplus \bigoplus_{i\in[d]\setminus[r]} \mathcal{H}_i^R \oplus \bigoplus_{i\in[\tilde{d}]\setminus[r]} \mathcal{H}_i^L\right)^\perp \textit{ and } \tilde{\mathcal{H}}_\perp := \left(\bigoplus_{i\in[r]} \tilde{\mathcal{H}}_i \oplus \bigoplus_{i\in[d]\setminus[r]} \tilde{\mathcal{H}}_i^R \oplus \bigoplus_{i\in[\tilde{d}]\setminus[r]} \tilde{\mathcal{H}}_i^L\right)^\perp.$$

Now we show that the subspaces $\mathcal{H}_i \colon i \in [k]$, $\mathcal{H}_i \colon i \in [r]\setminus[k]$, $\mathcal{H}_i^R \colon i \in [d]\setminus[r]$ and $\mathcal{H}_i^L \colon i \in [\tilde{d}]\setminus[r]$ are indeed pairwise orthogonal, by proving that their spanning bases described in Definition 12 form an orthonormal system of vectors. (By symmetry it also implies that the spanning bases of the $\tilde{\mathcal{H}}$ subspaces form also an orthonormal system of vectors.) The proof is summarized in Table 2, relying

[10] In singular value decomposition one usually requires that the diagonal elements of Σ are non-negative. Here we could also allow negative reals, all the proofs of this section would go through with minor modifications, mostly defining the ordering of the singular values with decreasing absolute value. This would enable one to treat spectral decompositions of Hermitian matrices also as singular value decompositions.

on the following observations:

$$\forall i,j \in [d] \qquad \langle\psi_i|\psi_j\rangle = \delta_{ij} \tag{18}$$

$$\forall i \in [d], j \in [r]\setminus[k] \qquad \langle\psi_i|\psi_j^\perp\rangle \propto \langle\psi_i|(I-\Pi)U^\dagger|\tilde{\psi}_j\rangle \propto \langle\psi_i|(I-\Pi) = 0 \tag{19}$$

$$\forall i \in [d], j \in [\tilde{d}]\setminus[r] \qquad \langle\psi_i|U^\dagger|\tilde{\psi}_j\rangle = \langle\psi_i|\Pi U^\dagger\widetilde{\Pi}|\tilde{\psi}_j\rangle = \langle\psi_i|A^\dagger|\tilde{\psi}_j\rangle \propto A^\dagger|\tilde{\psi}_j\rangle = 0 \tag{20}$$

$$\forall i,j \in [r]\setminus[k] \qquad \langle\psi_i^\perp|\psi_j^\perp\rangle = \frac{\langle\tilde{\psi}_i|U(I-\Pi)U^\dagger|\tilde{\psi}_j\rangle}{\sqrt{(1-\varsigma_i^2)(1-\varsigma_j^2)}} = \frac{\delta_{ij}-\langle\tilde{\psi}_i|AA^\dagger|\tilde{\psi}_j\rangle}{\sqrt{(1-\varsigma_i^2)(1-\varsigma_j^2)}} = \delta_{ij} \tag{21}$$

$$\forall i \in [r]\setminus[k], j \in [\tilde{d}]\setminus[r] \qquad \langle\psi_i^\perp|U^\dagger|\tilde{\psi}_j\rangle = \frac{\langle\tilde{\psi}_i|U(I-\Pi)U^\dagger|\tilde{\psi}_j\rangle}{\sqrt{(1-\varsigma_i^2)}} = \frac{\delta_{ij}-\langle\tilde{\psi}_i|AA^\dagger|\tilde{\psi}_j\rangle}{\sqrt{(1-\varsigma_i^2)}} = 0 \tag{22}$$

$$\forall i,j \in [\tilde{d}] \qquad \langle\tilde{\psi}_i|\tilde{\psi}_j\rangle = \delta_{ij} \tag{23}$$

$\mathcal{H}_i \perp \mathcal{H}_j$	$\lvert\psi_j\rangle \in \mathcal{H}_j$ $j \in [k]$	$\lvert\psi_j\rangle \in \mathcal{H}_j$ $j \in [r]\setminus[k]$	$\lvert\psi_j^\perp\rangle \in \mathcal{H}_j$ $j \in [r]\setminus[k]$	$\lvert\psi_j\rangle \in \mathcal{H}_j^R$ $j \in [d]\setminus[r]$	$U^\dagger\lvert\tilde{\psi}_j\rangle \in \mathcal{H}_j^L$ $j \in [\tilde{d}]\setminus[r]$
$\lvert\psi_i\rangle \in \mathcal{H}_i$ $i \in [k]$	by (18) $\langle\psi_i\lvert\psi_j\rangle = \delta_{ij}$	by (18) $\langle\psi_i\lvert\psi_j\rangle = 0$	by (19) $\langle\psi_i\lvert\psi_j^\perp\rangle = 0$	by (18) $\langle\psi_i\lvert\psi_j\rangle = 0$	by (20) $\langle\psi_i\lvert U^\dagger\lvert\tilde{\psi}_j\rangle = 0$
$\lvert\psi_i\rangle \in \mathcal{H}_i$ $i \in [r]\setminus[k]$		by (18) $\langle\psi_i\lvert\psi_j\rangle = \delta_{ij}$	by (19) $\langle\psi_i\lvert\psi_j^\perp\rangle = 0$	by (18) $\langle\psi_i\lvert\psi_j\rangle = 0$	by (20) $\langle\psi_i\lvert U^\dagger\lvert\tilde{\psi}_j\rangle = 0$
$\lvert\psi_i^\perp\rangle \in \mathcal{H}_i$ $i \in [r]\setminus[k]$			by (21) $\langle\psi_i^\perp\lvert\psi_j^\perp\rangle = \delta_{ij}$	by (19) $\langle\psi_i^\perp\lvert\psi_j\rangle = 0$	by (22) $\langle\psi_i^\perp\lvert U^\dagger\lvert\tilde{\psi}_j\rangle = 0$
$\lvert\psi_i\rangle \in \mathcal{H}_i^R$ $i \in [d]\setminus[r]$				by (18) $\langle\psi_i\lvert\psi_j\rangle = \delta_{ij}$	by (20) $\langle\psi_i\lvert U^\dagger\lvert\tilde{\psi}_j\rangle = 0$
$U^\dagger\lvert\tilde{\psi}_i\rangle \in \mathcal{H}_i^L$ $i \in [\tilde{d}]\setminus[r]$					by (23) $\langle\tilde{\psi}_i\lvert UU^\dagger\lvert\tilde{\psi}_j\rangle = \delta_{ij}$

Table 2: Proof of the orthonormality of the spanning bases described in Definition 12.

Now we introduce some notation for matrices that represent linear maps acting between different subspaces. This will enable us to conveniently express matrices in a block-diagonal form. We will use the subspaces of Definition 12, because they enable us to block-diagonalize the unitaries used for implementing singular value transformation.

Definition 13 (Notation for matrices of linear maps between different vector spaces)**.** *For two vector (sub)spaces $\mathcal{H}, \mathcal{H}'$ let us denote by $[\,\cdot\,]_{\mathcal{H}'}^{\mathcal{H}}$ the matrix of a linear map that maps $\mathcal{H} \mapsto \mathcal{H}'$. Moreover, if the subspaces are as in Definition 12 and we explicitly write down matrix elements, they are meant to be interpreted in the spanning bases we used for defining $\mathcal{H}, \mathcal{H}'$ in Definition 12.*

Lemma 14 (Invariant subspace decomposition of a projected unitary)**.** *Let $\mathcal{H}_U$ be a finite-dimensional Hilbert-space and $U, \Pi, \widetilde{\Pi} \in \mathrm{End}(\mathcal{H}_U)$ be as in Definition 11. Then using the singular value decomposition of Definition 12 we have that*

$$U = \bigoplus_{i\in[k]} [\varsigma_i]^{\mathcal{H}_i}_{\tilde{\mathcal{H}}_i} \oplus \bigoplus_{i\in[r]\setminus[k]} \begin{bmatrix} \varsigma_i & \sqrt{1-\varsigma_i^2} \\ \sqrt{1-\varsigma_i^2} & -\varsigma_i \end{bmatrix}^{\mathcal{H}_i}_{\tilde{\mathcal{H}}_i} \oplus \bigoplus_{i\in[d]\setminus[r]} [1]^{\mathcal{H}_i^R}_{\tilde{\mathcal{H}}_i^R} \oplus \bigoplus_{i\in[\tilde{d}]\setminus[r]} [1]^{\mathcal{H}_i^L}_{\tilde{\mathcal{H}}_i^L} \oplus [\,\cdot\,]^{\mathcal{H}_\perp}_{\tilde{\mathcal{H}}_\perp}. \tag{24}$$

Moreover,

$$2\Pi - I = \bigoplus_{i\in[k]} [1]^{\mathcal{H}_i}_{\mathcal{H}_i} \oplus \bigoplus_{i\in[r]\setminus[k]} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}^{\mathcal{H}_i}_{\mathcal{H}_i} \oplus \bigoplus_{i\in[d]\setminus[r]} [1]^{\mathcal{H}_i^R}_{\mathcal{H}_i^R} \oplus \bigoplus_{i\in[d]\setminus[r]} [-1]^{\mathcal{H}_i^L}_{\mathcal{H}_i^L} \oplus [\,\cdot\,]^{\mathcal{H}_\perp}_{\mathcal{H}_\perp}, \tag{25}$$

$$e^{i\phi(2\Pi-I)} = \bigoplus_{i\in[k]} \left[e^{i\phi}\right]^{\mathcal{H}_i}_{\mathcal{H}_i} \oplus \bigoplus_{i\in[r]\setminus[k]} \begin{bmatrix} e^{i\phi} & 0 \\ 0 & e^{-i\phi} \end{bmatrix}^{\mathcal{H}_i}_{\mathcal{H}_i} \oplus \bigoplus_{i\in[d]\setminus[r]} \left[e^{i\phi}\right]^{\mathcal{H}_i^R}_{\mathcal{H}_i^R} \oplus \bigoplus_{i\in[d]\setminus[r]} \left[e^{-i\phi}\right]^{\mathcal{H}_i^L}_{\mathcal{H}_i^L} \oplus [\,\cdot\,]^{\mathcal{H}_\perp}_{\mathcal{H}_\perp}, \tag{26}$$

and

$$2\widetilde{\Pi} - I = \bigoplus_{i\in[k]} [1]^{\tilde{\mathcal{H}}_i}_{\tilde{\mathcal{H}}_i} \oplus \bigoplus_{i\in[r]\setminus[k]} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}^{\tilde{\mathcal{H}}_i}_{\tilde{\mathcal{H}}_i} \oplus \bigoplus_{i\in[d]\setminus[r]} [-1]^{\tilde{\mathcal{H}}_i^R}_{\tilde{\mathcal{H}}_i^R} \oplus \bigoplus_{i\in[d]\setminus[r]} [1]^{\tilde{\mathcal{H}}_i^L}_{\tilde{\mathcal{H}}_i^L} \oplus [\,\cdot\,]^{\tilde{\mathcal{H}}_\perp}_{\tilde{\mathcal{H}}_\perp}, \tag{27}$$

$$e^{i\phi(2\widetilde{\Pi}-I)} = \bigoplus_{i\in[k]} \left[e^{i\phi}\right]^{\tilde{\mathcal{H}}_i}_{\tilde{\mathcal{H}}_i} \oplus \bigoplus_{i\in[r]\setminus[k]} \begin{bmatrix} e^{i\phi} & 0 \\ 0 & e^{-i\phi} \end{bmatrix}^{\tilde{\mathcal{H}}_i}_{\tilde{\mathcal{H}}_i} \oplus \bigoplus_{i\in[d]\setminus[r]} \left[e^{-i\phi}\right]^{\tilde{\mathcal{H}}_i^R}_{\tilde{\mathcal{H}}_i^R} \oplus \bigoplus_{i\in[d]\setminus[r]} \left[e^{i\phi}\right]^{\tilde{\mathcal{H}}_i^L}_{\tilde{\mathcal{H}}_i^L} \oplus [\,\cdot\,]^{\tilde{\mathcal{H}}_\perp}_{\tilde{\mathcal{H}}_\perp}. \tag{28}$$

Proof. For all $i \in [r] \setminus [k]$ we can verify that

$$U|\psi_i\rangle = \widetilde{\Pi}U|\psi_i\rangle + (I-\widetilde{\Pi})U|\psi_i\rangle = \underbrace{\widetilde{\Pi}U\Pi}_{A}|\psi_i\rangle + (I-\widetilde{\Pi})U|\psi_i\rangle = \varsigma_i|\tilde{\psi}_i\rangle + \sqrt{1-\varsigma_i^2}|\tilde{\psi}_i^\perp\rangle, \tag{29}$$

and

$$\begin{aligned}\sqrt{1-\varsigma_i^2}U|\psi_i^\perp\rangle = U(I-\Pi)U^\dagger|\tilde{\psi}_i\rangle = |\tilde{\psi}_i\rangle - U\Pi U^\dagger|\tilde{\psi}_i\rangle = |\tilde{\psi}_i\rangle - U\underbrace{\Pi U^\dagger\widetilde{\Pi}}_{A^\dagger}|\tilde{\psi}_i\rangle = |\tilde{\psi}_i\rangle - U\varsigma_i|\psi_i\rangle \\ = (1-\varsigma_i^2)|\tilde{\psi}_i\rangle - \varsigma_i\sqrt{1-\varsigma_i^2}|\tilde{\psi}_i^\perp\rangle,\end{aligned} \tag{30}$$

where in the last equality we used (29). Since U is unitary, it preserves the inner product and therefore maps $\mathcal{H}_\perp$ onto $\tilde{\mathcal{H}}_\perp$. Now equation (24) directly follows from (29)-(30). The other statements trivially follow from Definition 11. □

Definition 15 (Alternating phase modulation sequence)**.** *Let $\mathcal{H}_U$ be a finite-dimensional Hilbert space and let $U, \Pi, \widetilde{\Pi} \in \mathrm{End}(\mathcal{H}_U)$ be linear operators on $\mathcal{H}_U$ such that U is a unitary, and $\Pi, \widetilde{\Pi}$ are orthogonal projectors. Let $\Phi \in \mathbb{R}^n$, then we define the* phased alternating sequence *U_Φ as follows*

$$U_\Phi := \begin{cases} e^{i\phi_1(2\widetilde{\Pi}-I)}U\prod_{j=1}^{(n-1)/2}\left(e^{i\phi_{2j}(2\Pi-I)}U^\dagger e^{i\phi_{2j+1}(2\widetilde{\Pi}-I)}U\right) & \text{if } n \text{ is odd, and} \\ \prod_{j=1}^{n/2}\left(e^{i\phi_{2j-1}(2\Pi-I)}U^\dagger e^{i\phi_{2j}(2\widetilde{\Pi}-I)}U\right) & \text{if } n \text{ is even.} \end{cases} \tag{31}$$

Definition 16 (Singular value transformation by even/odd functions)**.** *Let $f:\mathbb{R}\to\mathbb{C}$ be an even or odd function. Let $A\in\mathbb{C}^{\tilde{d}\times d}$, let $d_{\min}:=\min(d,\tilde{d})$ and let*

$$A=\sum_{i=1}^{d_{\min}}\varsigma_i|\tilde{\psi}_i\rangle\langle\psi_i|$$

be a singular value decomposition of A.

We define the polynomial singular value transformation *of A, for odd function f as*

$$f^{(SV)}(A):=\sum_{i=1}^{d_{\min}}f(\varsigma_i)|\tilde{\psi}_i\rangle\langle\psi_i|,$$

and for even f as

$$f^{(SV)}(A):=\sum_{i=1}^{d}f(\varsigma_i)|\psi_i\rangle\langle\psi_i|,$$

where for $i\in[d]\setminus[d_{\min}]$ we define $\varsigma_i:=0$.

The following theorem is a generalized and improved version of the "Flexible quantum signal processing" result of Low and Chuang [LC17a, Theorem 4]. Our result is more general because it works for arbitrary matrices as opposed to only working for Hermitian (or normal) matrices. Another improvement is that we remove the constraint $P_{\Re}(0)=0$ for even d, which stems from our improved treatment of Theorem 5 and Corollary 8. Also note that the following theorem can be viewed as a generalization of the quantum walk techniques introduced by Szegedy [Sze04].

Theorem 17 (Singular value transformation by alternating phase modulation)**.** *Let $\mathcal{H}_U$ be a finite-dimensional Hilbert space and let $U,\Pi,\widetilde{\Pi}\in\mathrm{End}(\mathcal{H}_U)$ be linear operators on $\mathcal{H}_U$ such that U is a unitary, and $\Pi,\widetilde{\Pi}$ are orthogonal projectors. Let $P\in\mathbb{C}[x]$ and $\Phi\in\mathbb{R}^n$ is as in Corollary 8, then*

$$P^{(SV)}(\widetilde{\Pi}U\Pi)=\begin{cases}\widetilde{\Pi}U_\Phi\Pi & \textit{if } n \textit{ is odd, and}\\ \Pi U_\Phi\Pi & \textit{if } n \textit{ is even.}\end{cases}\tag{32}$$

Proof. We first prove the odd case. Observe that $P(1)=\prod_{j=1}^{n}e^{i\phi_j}$, and let $e^{i\phi_0}:=e^{i\sum_{j=1}^{n}(-1)^n\phi_j}$

$$\begin{aligned}U_\Phi&=e^{i\phi_1(2\widetilde{\Pi}-I)}U\prod_{j=1}^{n/2}\left(e^{i\phi_{2j}(2\Pi-I)}U^\dagger e^{i\phi_{2j+1}(2\widetilde{\Pi}-I)}U\right)\\&=\bigoplus_{i\in[k]}[\varsigma_k^nP(1)]_{\tilde{\mathcal{H}}_i}^{\mathcal{H}_i}\oplus\bigoplus_{i\in[r]\setminus[k]}\left[\prod_{j=1}^{n}\left(e^{i\phi_j\sigma_z}R(\varsigma_\ell)\right)\right]_{\tilde{\mathcal{H}}_i}^{\mathcal{H}_i}\oplus\bigoplus_{i\in[d]\setminus[r]}\left[e^{i\phi_0}\right]_{\tilde{\mathcal{H}}_i^R}^{\mathcal{H}_i^R}\oplus\bigoplus_{i\in[\tilde{d}]\setminus[r]}\left[e^{-i\phi_0}\right]_{\tilde{\mathcal{H}}_i^L}^{\mathcal{H}_i^L}\oplus[\,\cdot\,]_{\tilde{\mathcal{H}}_\perp}^{\mathcal{H}_\perp}\\&\hspace{30em}\text{(by Lemma 14)}\\&=\bigoplus_{i\in[k]}[P(\varsigma_i)]_{\tilde{\mathcal{H}}_i}^{\mathcal{H}_i}\oplus\bigoplus_{i\in[r]\setminus[k]}\begin{bmatrix}P(\varsigma_i)&\cdot\\\cdot&\cdot\end{bmatrix}_{\tilde{\mathcal{H}}_i}^{\mathcal{H}_i}\oplus\bigoplus_{i\in[d]\setminus[r]}\left[e^{i\phi_0}\right]_{\tilde{\mathcal{H}}_i^R}^{\mathcal{H}_i^R}\oplus\bigoplus_{i\in[\tilde{d}]\setminus[r]}\left[e^{-i\phi_0}\right]_{\tilde{\mathcal{H}}_i^L}^{\mathcal{H}_i^L}\oplus[\,\cdot\,]_{\tilde{\mathcal{H}}_\perp}^{\mathcal{H}_\perp}.\\&\hspace{30em}\text{(by Corollary 8)}\end{aligned}$$

Finally equation (32) follows from the fact that $\Pi=\sum_{i=1}^{d}|\psi_i\rangle\langle\psi_i|$ and $\widetilde{\Pi}=\sum_{i=1}^{\tilde{d}}|\tilde{\psi}_i\rangle\langle\tilde{\psi}_i|$, therefore

$$\begin{aligned}\widetilde{\Pi}U_\Phi\Pi &= \bigoplus_{i\in[k]}[P(\varsigma_i)]_{\tilde{\mathcal{H}}_i}^{\mathcal{H}_i} \oplus \bigoplus_{i\in[r]\setminus[k]}\left[\begin{array}{cc} P(\varsigma_i) & 0\\ 0 & 0\end{array}\right]_{\tilde{\mathcal{H}}_i}^{\mathcal{H}_i} \oplus \bigoplus_{i\in[d]\setminus[r]}[0]_{\tilde{\mathcal{H}}_i^R}^{\mathcal{H}_i^R} \oplus \bigoplus_{i\in[\tilde{d}]\setminus[r]}[0]_{\tilde{\mathcal{H}}_i^L}^{\mathcal{H}_i^L} \oplus [0]_{\tilde{\mathcal{H}}_\perp}^{\mathcal{H}_\perp}\\ &= \sum_{i=1}^{d_{\min}} P(\varsigma_i)|\tilde{\psi}_i\rangle\langle\psi_i|.\end{aligned}$$

The last equality follows from the observation that for n odd P is odd, therefore $P(0)=0$.

For the even case we can similarly derive that

$$U_\Phi = \bigoplus_{i\in[k]}[P(\varsigma_i)]_{\mathcal{H}_i}^{\mathcal{H}_i} \oplus \bigoplus_{i\in[r]\setminus[k]}\left[\begin{array}{cc} P(\varsigma_i) & \cdot\\ \cdot & \cdot\end{array}\right]_{\mathcal{H}_i}^{\mathcal{H}_i} \oplus \bigoplus_{i\in[d]\setminus[r]}\left[e^{-i\phi_0}\right]_{\mathcal{H}_i^R}^{\mathcal{H}_i^R} \oplus \bigoplus_{i\in[\tilde{d}]\setminus[r]}\left[e^{i\phi_0}\right]_{\mathcal{H}_i^L}^{\mathcal{H}_i^L} \oplus [\,\cdot\,]_{\mathcal{H}_\perp}^{\mathcal{H}_\perp}. \qquad \text{(by Corollary 8)}$$

Finally equation (32) follows from the fact that $\Pi=\sum_{i=1}^{d}|\psi_i\rangle\langle\psi_i|$, and therefore

$$\begin{aligned}\Pi U_\Phi\Pi &= \bigoplus_{i\in[k]}[P(\varsigma_i)]_{\mathcal{H}_i}^{\mathcal{H}_i} \oplus \bigoplus_{i\in[r]\setminus[k]}\left[\begin{array}{cc} P(\varsigma_i) & 0\\ 0 & 0\end{array}\right]_{\mathcal{H}_i}^{\mathcal{H}_i} \oplus \bigoplus_{i\in[d]\setminus[r]}\left[e^{-i\phi_0}\right]_{\mathcal{H}_i^R}^{\mathcal{H}_i^R} \oplus \bigoplus_{i\in[\tilde{d}]\setminus[r]}[0]_{\mathcal{H}_i^L}^{\mathcal{H}_i^L} \oplus [0]_{\mathcal{H}_\perp}^{\mathcal{H}_\perp}\\ &= \sum_{i=1}^{d} P(\varsigma_i)|\psi_i\rangle\langle\psi_i|.\end{aligned}$$

The last equality uses the observation that for n even $P(0)=e^{-i\phi_0}$, as shown by Corollary 8. □

Corollary 18 (Singular value transformation by real polynomials). *Let $U,\Pi,\widetilde{\Pi}$ be as in Theorem 17. Suppose that $P_{\Re}\in\mathbb{R}[x]$ is a degree-n polynomial satisfying that*

- *$P_{\Re}$ has parity-($n \mod 2$) and*
- *for all $x\in[-1,1]$: $|P_{\Re}(x)|\leq 1$.*

Then there exist $\Phi\in\mathbb{R}^n$, such that

$$P_{\Re}^{(SV)}\left(\widetilde{\Pi}U\Pi\right)=\begin{cases}\left(\langle+|\otimes\widetilde{\Pi}\right)\left(|0\rangle\langle 0|\otimes U_\Phi+|1\rangle\langle 1|\otimes U_{-\Phi}\right)\left(|+\rangle\otimes\Pi\right) & \text{if } n \text{ is odd, and}\\ \left(\langle+|\otimes\Pi\right)\left(|0\rangle\langle 0|\otimes U_\Phi+|1\rangle\langle 1|\otimes U_{-\Phi}\right)\left(|+\rangle\otimes\Pi\right) & \text{if } n \text{ is even.}\end{cases} \tag{33}$$

Proof. By Corollary 10 we can find a $\Phi\in\mathbb{R}^n$ such that $\Re[P]=P_{\Re}$. Observe that $-\Phi$ gives rise to P^* in Corollary 8 as can be seen from equation (14). Let $\Pi'=\widetilde{\Pi}$ for n odd and let $\Pi'=\Pi$ for n even. Then by Theorem 17 we get that $P^{(SV)}\left(\widetilde{\Pi}U\Pi\right)=\Pi' U_\Phi\Pi$, and $P^{*(SV)}\left(\widetilde{\Pi}U\Pi\right)=\Pi' U_{-\Phi}\Pi$. Therefore

$$\begin{aligned}\left(\langle+|\otimes\Pi'\right)\left(|0\rangle\langle 0|\otimes U_\Phi\right)\left(|+\rangle\otimes\Pi\right) &= P^{(SV)}\left(\widetilde{\Pi}U\Pi\right)/2\\ \left(\langle+|\otimes\Pi'\right)\left(|1\rangle\langle 1|\otimes U_{-\Phi}\right)\left(|+\rangle\otimes\Pi\right) &= P^{*(SV)}\left(\widetilde{\Pi}U\Pi\right)/2.\end{aligned}$$

We can conclude by observing that $P_{\Re} = (P + P^*)/2$, and therefore

$$P_{\Re}^{(SV)}\left(\widetilde{\Pi} U \Pi\right) = \left(P^{(SV)}\left(\widetilde{\Pi} U \Pi\right) + P^{*(SV)}\left(\widetilde{\Pi} U \Pi\right)\right)/2.$$

□

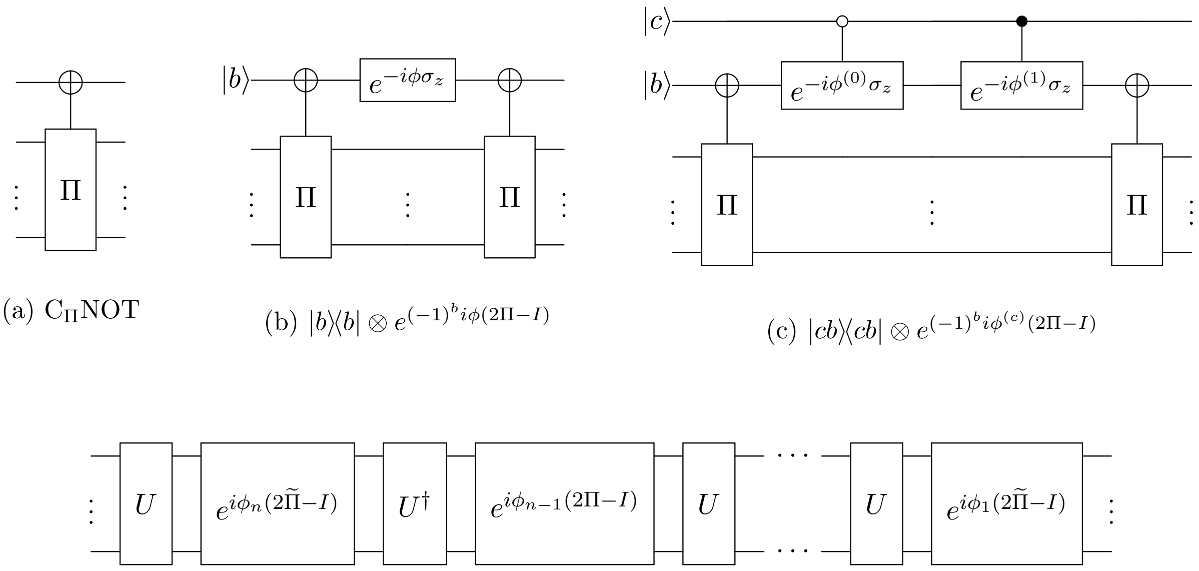
Note that the above result is essentially optimal in the sense that the requirements are necessary. It is obvious that the polynomial needs to be bounded within $[-1, 1]$ since the matrix must have norm at most 1 as it is a projected unitary. Also one cannot implement a degree d Chebyshev polynomial with $d-1$ uses of the unitary U, since $T_d(x)$ takes value 1 at 1 with derivative d^2. Substituting $y := 1$ and $x := 1 - \delta$ for some small δ to equation (68) in Theorem 73 shows that exactly implementing $T_d(x)$ requires at least d uses of U. Finally, about the parity constraint, note that every result in this subsection would stay valid if we would extend the concept of singular values by allowing negative values as well. But then by changing a singular vector/singular value term $\varsigma|\phi\rangle\langle\psi|$ to $-\varsigma|-\phi\rangle\langle\psi|$ would be again a valid decomposition, where singular value transformation could be applied with a polynomial P. It would require that $P(\varsigma)|\psi\rangle\langle\psi| = P(-\varsigma)|\psi\rangle\langle\psi|$, and $P(\varsigma)|\phi\rangle\langle\psi| = -P(-\varsigma)|-\phi\rangle\langle\psi|$ for consistency, showing the necessity of the even/odd constraint. Equations (34)-(35) in the proof of Corollary 21 also show that the even/odd case separation for singular value transformation is quite natural.

What remains is to discuss how to efficiently implement an alternating phase modulation sequences. Observe that with a single ancilla qubit, two uses of $\mathrm{C}_{\Pi}\mathrm{NOT}$, and a single-qubit phase gate $e^{-i\phi\sigma_z}$ we can implement the operator $e^{i\phi(2\Pi-I)}{=}\mathrm{C}_{\Pi}\mathrm{NOT}\left(I \otimes e^{-i\phi\sigma_z}\right)\mathrm{C}_{\Pi}\mathrm{NOT}$, which leads to an efficient implementation of U_Φ, see Figure 1b.

Lemma 19 (Efficient implementation of alternating phase modulation sequences)**.** *Let $\Phi \in \mathbb{R}^n$, then the alternating phased sequence U_Φ of Definition 15 can implemented using a single ancilla qubit with n uses of U and $U^\dagger$, n uses of $C_\Pi NOT$ and n uses of $C_{\widetilde{\Pi}}NOT$ gates and n single qubit gates. A controlled version of U_Φ can be built similarly just replacing the n single qubit gates by controlled gates, and in case n is odd replacing one U gate with a controlled U gate. For a set of vectors $\{\Phi^{(k)} \in \mathbb{R}^n \colon k \in \{0,1\}^m\}$ a multi-controlled alternating phased sequence $\sum_{k\in\{0,1\}^m} |k\rangle\langle k| \otimes U_{\Phi^{(k)}}$ can be implemented similarly by replacing the single qubit gates with multiply controlled single qubit gates of the form $\sum_{k\in\{0,1\}^m} |k\rangle\langle k| \otimes e^{i\phi^{(k)}}$.*

Proof. See the constructions of Figure 1. □

Note that Figure 1 also explains the term “qubitization”: the fine-tuned driving of the circuit giving rise to the required polynomial transformation is achieved by cleverly chosen Pauli-z rotations of a single ancilla qubit. The rotations of the single ancilla qubit induce rotations on the common two-dimensional invariant subspaces of $U, \Pi, \widetilde{\Pi}$ cf. Lemma 14.



Π
$|b\rangle$
$e^{-i\phi\sigma_z}$
Π
Π
$|c\rangle$
$|b\rangle$
$e^{-i\phi^{(0)}\sigma_z}$
$e^{-i\phi^{(1)}\sigma_z}$
Π
Π
U
$e^{i\phi_n(2\tilde{\Pi}-I)}$
$U^\dagger$
$e^{i\phi_{n-1}(2\Pi-I)}$
U
U
$e^{i\phi_1(2\tilde{\Pi}-I)}$

(a) $\mathrm{C}_\Pi\mathrm{NOT}$

(b) $|b\rangle\langle b| \otimes e^{(-1)^b i\phi(2\Pi-I)}$

(c) $|cb\rangle\langle cb| \otimes e^{(-1)^b i\phi^{(c)}(2\Pi-I)}$

(d) $U_\Phi = e^{i\phi_1(2\tilde{\Pi}-I)}U\prod_{j=1}^{(n-1)/2}\left(e^{i\phi_{2j}(2\Pi-I)}U^\dagger e^{i\phi_{2j+1}(2\tilde{\Pi}-I)}U\right)$ (for odd n)

Figure 1: Gates and gate sequences used for singular value transformation in Theorem 17. Figure 1a shows how to implement a $\mathrm{C}_\Pi\mathrm{NOT}$ gate, and Figure 1b shows how to implement $e^{i\phi(2\Pi-I)}$ using a single ancilla qubit, two $\mathrm{C}_\Pi\mathrm{NOT}$ gates and an $e^{-i\phi\sigma_z}$ gate. Figure 1c demonstrates how to implement a controlled version of the gate $e^{i\phi^{(c)}(2\Pi-I)}$, by only controlling the single qubit gate $e^{-i\phi^{(c)}\sigma_z}$. Finally, Figure 1d summarizes the complete circuit used in Theorem 17.

3.3 Robustness of singular value transformation

In this subsection we will prove results about the robustness of singular value transformation. More precisely we prove bounds on how big can be difference $\left\|P^{(SV)}(A)-P^{(SV)}(\tilde{A})\right\|$ in terms of the magnitude of "perturbation" $\left\|A-\tilde{A}\right\|$.

First consider the generalization of ordinary $\mathbb{R}\to\mathbb{C}$ functions to Hermitian matrices. One is tempted to think that if such a function is Lipschitz-continuous, then the induced operator function is also Lipschitz-continuous, however it turns out to be false. For a recent survey on the topic see the work of Aleksandrov and Peller [AP16].

Although the Lipschitz property cannot be saved directly, one may not lose more than some logarithmic factors. We invoke a nice result form the theory of operator functions, quantifying this claim. The following theorem is due to Farforovskaya and Nikolskaya [FN09, Theorem 10].

Theorem 20 (Robustness of eigenvalue transformation)**.** *Suppose that $f\colon[-1,1]\to\mathbb{C}$ is a function such that $\omega\colon[0,2]\to[0,\infty]$ is a modulus of continuity, i.e., for all $x,x'\in[-1,1]$*

$$|f(x)-f(x')|\leq\omega(|x-x'|).$$

Then for all A, B Hermitian matrices such that $\|A\|, \|B\| \leq 1$, we have that

$$\|f(A) - f(B)\| \leq 4\left[\ln\left(\frac{2}{\|A-B\|} + 1\right) + 1\right]^2 \omega(\|A - B\|).$$

Now we show how this general theorem implies a general robustness result for singular value transformation.

Corollary 21 (Robustness of singular value transformation 1)**.** *If $f\colon [-1,1] \to \mathbb{C}$ is an even or odd function such that $\omega\colon [0,2] \to [0,\infty]$ is a modulus of continuity, and $A, \tilde{A} \in \mathbb{C}^{\tilde{d}\times d}$ are matrices of operator norm at most 1, then we have that*

$$\left\|f^{(SV)}(A) - f^{(SV)}(\tilde{A})\right\| \leq 4\left[\ln\left(\frac{2}{\left\|A-\tilde{A}\right\|} + 1\right) + 1\right]^2 \omega\left(\left\|A - \tilde{A}\right\|\right).$$

Proof. Let us assume that f is an even function and that $\tilde{d} \leq d$. Then, using singular value decomposition, we can rewrite A as

$$A = W\begin{bmatrix}\Sigma & 0\end{bmatrix}V^\dagger,$$

where $W \in \mathbb{C}^{\tilde{d}\times\tilde{d}}$, $V \in \mathbb{C}^{d\times d}$ are unitaries and $\Sigma \in \mathbb{R}^{\tilde{d}\times\tilde{d}}$ is a diagonal matrix with nonnegative diagonal entries. Let $\overline{A} := \begin{bmatrix}0 & A\\ A^\dagger & 0\end{bmatrix} \in \mathbb{C}^{(\tilde{d}+d)\times(\tilde{d}+d)}$ be the Hermitian matrix obtained from A. We claim that

$$f(\overline{A}) = \begin{bmatrix}f^{(SV)}(A^\dagger) & 0\\ 0 & f^{(SV)}(A)\end{bmatrix}. \tag{34}$$

To prove this claim, first note that

$$\overline{A} = \begin{bmatrix}0 & A\\ A^\dagger & 0\end{bmatrix} = \begin{bmatrix}0 & W\begin{bmatrix}\Sigma & 0\end{bmatrix}V^\dagger\\ V\begin{bmatrix}\Sigma\\ 0\end{bmatrix}W^\dagger & \begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix}\end{bmatrix} = \begin{bmatrix}W & 0\\ 0 & V\end{bmatrix}\begin{bmatrix}0 & \Sigma & 0\\ \Sigma & 0 & 0\\ 0 & 0 & 0\end{bmatrix}\begin{bmatrix}W^\dagger & 0\\ 0 & V^\dagger\end{bmatrix}$$

and that

$$\begin{bmatrix}0 & \Sigma\\ \Sigma & 0\end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix}I & I\\ I & -I\end{bmatrix}\begin{bmatrix}\Sigma & 0\\ 0 & -\Sigma\end{bmatrix}\frac{1}{\sqrt{2}}\begin{bmatrix}I & I\\ I & -I\end{bmatrix}.$$

Therefore, if we denote

$$U = \begin{bmatrix}W & 0\\ 0 & V\end{bmatrix}\begin{bmatrix}\frac{1}{\sqrt{2}}\begin{bmatrix}I & I\\ I & -I\end{bmatrix} & \begin{matrix}0\\ 0\end{matrix}\\ \begin{matrix}0 & 0\end{matrix} & I\end{bmatrix},$$

we get

$$\overline{A} = U\begin{bmatrix}\Sigma & 0 & 0\\ 0 & -\Sigma & 0\\ 0 & 0 & 0\end{bmatrix}U^\dagger,$$

which implies that

$$\begin{aligned}f(\overline{A}) =& U\begin{bmatrix}f(\Sigma) & 0 & 0\\ 0 & f(-\Sigma) & 0\\ 0 & 0 & f(0)I\end{bmatrix}U^\dagger = U\begin{bmatrix}f(\Sigma) & 0 & 0\\ 0 & f(\Sigma) & 0\\ 0 & 0 & f(0)I\end{bmatrix}U^\dagger\\ =& \begin{bmatrix}Wf(\Sigma)W^\dagger & \begin{matrix}0 & 0\end{matrix}\\ \begin{matrix}0\\ 0\end{matrix} & V\begin{bmatrix}f(\Sigma) & 0\\ 0 & f(0)I\end{bmatrix}V^\dagger\end{bmatrix} = \begin{bmatrix}f^{(SV)}(A^\dagger) & 0\\ 0 & f^{(SV)}(A)\end{bmatrix}.\end{aligned}$$

Thus, using Theorem 20 we get that

$$\begin{aligned}
\left\|f^{(SV)}(A)-f^{(SV)}(\tilde{A})\right\| &\leq \left\|f(\overline{A})-f(\overline{\tilde{A}})\right\| \\
&\leq 4\left[\ln\left(\frac{2}{\left\|\overline{A}-\overline{\tilde{A}}\right\|}+1\right)+1\right]^2\omega\left(\left\|\overline{A}-\overline{\tilde{A}}\right\|\right) \\
&= 4\left[\ln\left(\frac{2}{\left\|A-\tilde{A}\right\|}+1\right)+1\right]^2\omega\left(\left\|A-\tilde{A}\right\|\right),
\end{aligned}$$

which completes the proof for the case where f is an even function and that $\tilde{d}\leq d$. The case $\tilde{d}\geq d$ can be handled by symmetry. Finally, the remaining case where f is odd can be handled similarly by observing that

$$f(\overline{A})=\begin{bmatrix} 0 & f^{(SV)}(A) \\ f^{(SV)}(A^{\dagger}) & 0 \end{bmatrix}. \tag{35}$$

□

We can also prove robustness results by bootstrapping our exact (non-robust) results, enabling us to remove the log factor from the above corollary under certain circumstances. We study two cases. First we make no extra assumptions, and establish error bounds that scale with the square root of the initial error. Then we improve the dependence to linear under the assumption that the singular values are bounded away from 1 in absolute value.

Lemma 22 (Robustness of singular value transformation 2)**.** *If $P\in\mathbb{C}[x]$ is a degree-n polynomial satisfying the requirements of Corollary 8, moreover $A,\tilde{A}\in\mathbb{C}^{\tilde{d}\times d}$ are matrices of operator norm at most 1, then we have*[11] *that*

$$\left\|P^{(SV)}(A)-P^{(SV)}(\tilde{A})\right\|\leq 4n\sqrt{\left\|A-\tilde{A}\right\|}.$$

Proof. Let $\varepsilon=\left\|\tilde{A}-A\right\|$, and let $B,\tilde{B}\in\mathbb{C}^{(d+\tilde{d})\times(d+\tilde{d})}$ be the matrices such that

$$B:=\left[\begin{array}{cc} A & 0 \\ 0 & 0 \end{array}\right],\quad \tilde{B}:=\left[\begin{array}{cc} \frac{\tilde{A}-A}{\varepsilon} & 0 \\ 0 & 0 \end{array}\right],$$

and let $U\in\mathbb{C}^{4(d+\tilde{d})\times 4(d+\tilde{d})}$ be a unitary such that[12]

$$U=\left[\begin{array}{cccc} B & 0 & \cdot & \cdot \\ 0 & \tilde{B} & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{array}\right].$$

[11] Let us do a sanity check for $d=\tilde{d}=1$. For large d we have that $T_d(1)-T_d(1-\frac{1}{2d^2})\approx 1-\cos(1)\approx 0.46$, whereas our upper bound gives $2\sqrt{2}$, showing that the upper bound is tight up to a constant factor, for arbitrary large d and for arbitrary small ε. (However, the joint dependence on d and ε might not be optimal.)

[12] We denote by $\cdot$ arbitrary matrix blocks and elements that are irrelevant for our presentation.

Such U must exist because $\|B\| \leq 1$ and $\left\|\tilde{B}\right\| \leq 1$. Let Π be the orthogonal projector projecting to the first d coordinates, and let $\widetilde{\Pi}$ be the orthogonal projector projecting to the first $\tilde{d}$ coordinates. Observe that $\widetilde{\Pi}U\Pi = A$. Let $W \in \mathbb{C}^{4(d+\tilde{d})\times 4(d+\tilde{d})}$ be the unitary

$$W := \begin{bmatrix} \sqrt{\frac{1}{1+\varepsilon}}I & -\sqrt{\frac{\varepsilon}{1+\varepsilon}}I & 0 & 0 \\ \sqrt{\frac{\varepsilon}{1+\varepsilon}}I & \sqrt{\frac{1}{1+\varepsilon}}I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{bmatrix}.$$

Let $\bar{U} := W^{\dagger}UW$, and observe that $\widetilde{\Pi}\bar{U}\Pi = \tilde{A}/(1+\varepsilon)$. Also observe that

$$\|W - I\| = \sqrt{2 - 2/\sqrt{1+\varepsilon}} \leq \sqrt{\varepsilon},$$

therefore $\left\|U - \bar{U}\right\| \leq 2\sqrt{\varepsilon}$. Let $\Pi' = \widetilde{\Pi}$ if n is odd, and let $\Pi' = \Pi$ for n even. Let Φ be as in Corollary 8, then Theorem 17 implies that

$$\left\|P^{(SV)}(A) - P^{(SV)}(\tilde{A}/(1+\varepsilon))\right\| = \left\|\Pi' U_{\Phi}\Pi - \Pi'\bar{U}_{\Phi}\Pi\right\| \leq \left\|U_{\Phi} - \bar{U}_{\Phi}\right\| \leq n\left\|U - \bar{U}\right\| \leq 2n\sqrt{\left\|A - \tilde{A}\right\|}.$$

Let $B' \in \mathbb{C}^{(d+\tilde{d})\times(d+\tilde{d})}$ be the matrix such that

$$B' := \begin{bmatrix} \tilde{A} & 0 \\ 0 & 0 \end{bmatrix},$$

and let $U' \in \mathbb{C}^{4(d+\tilde{d})\times 4(d+\tilde{d})}$ be a unitary such that

$$U' = \begin{bmatrix} B' & 0 & . & . \\ 0 & 0 & . & . \\ . & . & . & . \\ . & . & . & . \end{bmatrix}.$$

Observe that $\widetilde{\Pi}U'\Pi = \tilde{A}$, and $\bar{U}' := W^{\dagger}\tilde{V}W$ is such that $\widetilde{\Pi}\bar{U}'\Pi = \tilde{A}/(1+\varepsilon)$. By the same argument as before we get that

$$\left\|P^{(SV)}(\tilde{A}) - P^{(SV)}(\tilde{A}/(1+\varepsilon))\right\| \leq 2n\sqrt{\left\|A - \tilde{A}\right\|}.$$

We can conclude using the triangle inequality. □

Now we establish another lemma which improves on the previous results for example in the case when the singular values are bounded away from 1 in absolute value.

Lemma 23 (Robustness of singular value transformation 3)**.** *If $P \in \mathbb{C}[x]$ is a degree-n polynomial satisfying the requirements of Corollary 8, moreover $A, \tilde{A} \in \mathbb{C}^{\tilde{d}\times d}$ are matrices of operator norm at most* 1*, such that*

$$\left\|A - \tilde{A}\right\| + \left\|\frac{A + \tilde{A}}{2}\right\|^2 \leq 1,$$

then we have that

$$\left\|P^{(SV)}(A)-P^{(SV)}(\tilde{A})\right\| \leq n\sqrt{\frac{2}{1-\left\|\frac{A+\tilde{A}}{2}\right\|^2}}\left\|A-\tilde{A}\right\|.$$

Proof. Let $B,\tilde{B}\in\mathbb{C}^{(d+\tilde{d})\times(d+\tilde{d})}$ be the matrices such that

$$B := \begin{bmatrix} \frac{A+\tilde{A}}{\|A+\tilde{A}\|} & 0 \\ 0 & 0 \end{bmatrix},\ \ \tilde{B} := \begin{bmatrix} \frac{A-\tilde{A}}{\|A-\tilde{A}\|} & 0 \\ 0 & 0 \end{bmatrix}.$$

Let $x>1$ and let $U\in\mathbb{C}^{4(d+\tilde{d})\times4(d+\tilde{d})}$ be a unitary such that

$$U = \begin{bmatrix} \sqrt{\frac{x-1}{x}}B & \sqrt{\frac{1}{x}}\tilde{B} & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix}.$$

Let $C:=\sqrt{\frac{x-1}{x}}B\oplus\sqrt{\frac{1}{x}}\tilde{B}$ be top-left block of U. It is easy to see that

$$\|C\|^2 \leq \frac{x-1}{x}\|B\|^2+\frac{1}{x}\left\|\tilde{B}\right\|^2 = \frac{x-1}{x}+\frac{1}{x}=1,$$

therefore a unitary U must exist with C being the top-left block. Suppose that

$$\frac{x}{x-1}\frac{\left\|A+\tilde{A}\right\|^2}{4}+x\frac{\left\|A-\tilde{A}\right\|^2}{4}=1. \tag{36}$$

Let $W_{\pm}\in\mathbb{C}^{4(d+\tilde{d})\times4(d+\tilde{d})}$ be the unitary

$$W_{\pm} := \begin{bmatrix} \sqrt{\frac{x}{x-1}}\frac{\|A+\tilde{A}\|}{2}I & \mp\sqrt{x}\frac{\|A-\tilde{A}\|}{2}I & 0 & 0 \\ \pm\sqrt{x}\frac{\|A-\tilde{A}\|}{2}I & \sqrt{\frac{x}{x-1}}\frac{\|A+\tilde{A}\|}{2}I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{bmatrix}.$$

Let Π be the orthogonal projector projecting to the first d coordinates, and let $\widetilde{\Pi}$ be the orthogonal projector projecting to the first $\tilde{d}$ coordinates. Observe that $\widetilde{\Pi}UW_{+}\Pi=A$ and $\widetilde{\Pi}UW_{-}\Pi=\tilde{A}$. Also observe that $\|W_{+}-W_{-}\|=\sqrt{x}\left\|A-\tilde{A}\right\|$, thus $\|UW_{+}-UW_{-}\|=\sqrt{x}\left\|A-\tilde{A}\right\|$.

Let $\varepsilon:=\left\|A-\tilde{A}\right\|^2$ and let $\delta:=4-\left\|A+\tilde{A}\right\|^2$. We can rewrite (36) as

$$\frac{x}{x-1}\frac{4-\delta}{4}+x\frac{\varepsilon}{4}=1, \tag{37}$$

which has a solution

$$x=\frac{4}{\delta+\varepsilon}\left(1+\frac{\left(1-\frac{8\varepsilon}{(\delta+\varepsilon)^2}\right)-\sqrt{1-\frac{16\varepsilon}{(\delta+\varepsilon)^2}}}{\frac{8\varepsilon}{(\delta+\varepsilon)^2}}\right). \tag{38}$$

Now let $y := 8\varepsilon/(\delta+\varepsilon)^2$, it is easy to see that for $y \leq \frac{1}{2}$ we have that $\frac{(1-y)-\sqrt{1-2y}}{y} \leq 1$. It is also easy to see that if $\varepsilon \leq \delta^2/16$, then $y \leq \frac{1}{2}$. Thus for $\varepsilon \leq \delta^2/16$ we get that $x \leq 8/(\delta+\varepsilon)$, and therefore $\|UW_+ - UW_-\| = \sqrt{8/(\delta+\varepsilon)}\left\|A-\tilde{A}\right\| \leq \sqrt{8/\delta}\left\|A-\tilde{A}\right\|$.

Now we proceed similarly to the proof of Lemma 22. Let $\Pi' = \widetilde{\Pi}$ if n is odd, and let $\Pi' = \Pi$ for n even. Let Φ be as in Corollary 8 and let $U^{(\pm)} := UW_\pm$, then Theorem 17 implies that

$$\left\|P^{(SV)}(A) - P^{(SV)}(\tilde{A})\right\| = \left\|\Pi' U_\Phi^+ \Pi - \Pi' U_\Phi^- \Pi\right\| \leq \left\|U_\Phi^+ - U_\Phi^-\right\| \leq n\left\|U^+ - U^-\right\| = n\sqrt{\frac{8}{\delta}}\left\|A-\tilde{A}\right\|.$$

Finally note that $\varepsilon \leq \delta^2/16$ is equivalent to $4\sqrt{\varepsilon} \leq \delta$, which by definition is equivalent to

$$\left\|A-\tilde{A}\right\| + \left\|\frac{A+\tilde{A}}{2}\right\|^2 \leq 1.$$

□

3.4 Singular vector transformation and singular value amplification

In this subsection we derive some corollaries of singular value transformation. We call the first corollary projected singular vector transformation, because it implements a unitary that transforms the right singular vectors to the left singular vectors above some singular value threshold. Then we show how to quickly derive advanced amplitude amplification results using this general technique. Finally, we develop a corollary called singular value amplification, which shows how to uniformly amplify the singular values of a matrix represented as a projected unitary.

First we define singular value threshold projectors which are slight modifications of the singular value projectors of Definition 1.

Definition 24 (Singular value threshold projectors)**.** *Let $A = \widetilde{\Pi}U\Pi = W\Sigma V^\dagger$ be a singular value decomposition of a projected unitary. For $S \subseteq \mathbb{R}$ we define $\Pi_S := \Pi V \Sigma_S V^\dagger \Pi$, and similarly $\widetilde{\Pi}_S := \widetilde{\Pi} W \Sigma_S W^\dagger \widetilde{\Pi}$. For $\delta \in \mathbb{R}$ we define $\Pi_{\geq\delta} := \Pi_{[\delta,\infty)}$, also we define $\Pi_{>\delta}, \Pi_{\leq\delta}, \Pi_{<\delta}, \Pi_{=\delta}$ and $\widetilde{\Pi}_{>\delta}, \widetilde{\Pi}_{\leq\delta}, \widetilde{\Pi}_{<\delta}, \widetilde{\Pi}_{=\delta}$ analogously.*

Then we invoke a result of Low and Chuang [LC17a, Corollary 6] about constructive polynomial approximations of the sign function – the error of the optimal approximation, studied by Eremenko and Yuditskii [EY07], achieves similar scaling but is non-constructive.

Lemma 25 (Polynomial approximations of the sign function)**.** *For all $\delta > 0$, $\varepsilon \in (0, 1/2)$ there exists an efficiently computable odd polynomial $P \in \mathbb{R}[x]$ of degree $n = \mathcal{O}\left(\frac{\log(1/\varepsilon)}{\delta}\right)$, such that*

- *for all $x \in [-2, 2]$: $|P(x)| \leq 1$, and*
- *for all $x \in [-2, 2] \setminus (-\delta, \delta)$: $|P(x) - \mathrm{sign}(x)| \leq \varepsilon$.*

Now we are ready to prove our result about singular value transformation. Our singular vector transformation implements a unitary which maps a right singular vector having singular value at least δ to the corresponding left singular vector.

Theorem 26 (Singular vector transformation)**.** *Let $U, \Pi, \widetilde{\Pi}$ be as in Theorem 17 and let $\delta > 0$. Suppose that $\widetilde{\Pi} U \Pi = W \Sigma V^\dagger$ is a singular value decomposition. Then there is an $m = \mathcal{O}\left(\frac{\log(1/\varepsilon)}{\delta}\right)$ and a $\Phi \in \mathbb{R}^m$ such that $\left\| \widetilde{\Pi}_{\geq\delta} U_\Phi \Pi_{\geq\delta} - \widetilde{\Pi}_{\geq\delta}(WV^\dagger)\Pi_{\geq\delta} \right\| \leq \varepsilon$. Moreover, U_Φ can be implemented using a single ancilla qubit with m uses of U and $U^\dagger$, m uses of $C_\Pi NOT$ and m uses of $C_{\widetilde{\Pi}} NOT$ gates and m single qubit gates.*

Proof. By Lemma 25 we can construct an odd polynomial $P_\Re \in \mathbb{R}[x]$ of degree $m = \mathcal{O}\left(\frac{\log(1/\varepsilon^2)}{\delta}\right)$ that approximates the sign function with $\varepsilon^2/2$ precision on the domain $[-1,1] \setminus (-\delta, \delta)$. By Corollary 10 we know that there exists a polynomial P of the same degree as $P_\Re$ such that $\Re[P] = P_\Re$, moreover P satisfies the conditions of Corollary 8. Use singular value transformation Theorem 17 to construct a $\Phi \in \mathbb{R}^m$ such that $\widetilde{\Pi} U_\Phi \Pi = P^{(SV)}\left(\widetilde{\Pi} U \Pi\right)$ up to precision ε and observe that $\left\| \widetilde{\Pi}_{\geq\delta} P^{(SV)}\left(\widetilde{\Pi} U \Pi\right) \Pi_{\geq\delta} - \widetilde{\Pi}_{\geq\delta}(WV^\dagger)\Pi_{\geq\delta} \right\| \leq \varepsilon$. Conclude the gate complexity using Lemma 19. □

As an easy corollary we recover and improve upon fixed-point amplitude amplification results [Hø00, Gro05, AC12, YLC14] by combining the advantages of prior art. On one hand, the query complexity of $\mathcal{O}(\frac{1}{\delta}\,\mathrm{poly}(1/\varepsilon))$ by [AC12] is optimal with respect to target state overlap δ, but converges slowly with respect to error ε. On the other hand, the query complexity of $\mathcal{O}(\frac{1}{\delta}\log(1/\varepsilon))$ by [YLC14] is optimal and exhibits exponentially fast convergence with respect to the error, but it introduces an unknown phase on the amplified state. Our presented approach has the same optimal asymptotic scaling and also ensures that this phase error is ϵ-close to 0.

Theorem 27 (Fixed-point amplitude amplification)**.** *Let U be a unitary and Π be an orthogonal projector such that $a|\psi_G\rangle = \Pi U |\psi_0\rangle$, and $a > \delta > 0$. There is a unitary circuit $\tilde{U}$ such that $\left\| |\psi_G\rangle - \tilde{U}|\psi_0\rangle \right\| \leq \varepsilon$, which uses a single ancilla qubit and consists of $\mathcal{O}\left(\frac{\log(1/\varepsilon)}{\delta}\right)$ $U, U^\dagger$, $C_\Pi NOT$, $C_{|\psi_0\rangle\langle\psi_0|} NOT$ and $e^{i\phi\sigma_z}$ gates.*

Proof. Set $\widetilde{\Pi} := \Pi$ and $\Pi' := |\psi_0\rangle\langle\psi_0|$ and observe that

$$\widetilde{\Pi} U \Pi' = a|\psi_G\rangle\langle\psi_0|.$$

Now use Theorem 26 in order to get an algorithm $\tilde{U}$ that satisfies

$$\left\| |\psi_G\rangle\langle\psi_G|\tilde{U}|\psi_0\rangle\langle\psi_0| - |\psi_G\rangle\langle\psi_0| \right\| \leq \varepsilon.$$

□

Another easy to derive corollary of our machinery is robust oblivious amplitude amplification.[13]

Theorem 28 (Robust oblivious amplitude amplification)**.** *Let $n \in \mathbb{N}_+$ be odd, let $\varepsilon \in \mathbb{R}_+$, let U be a unitary, let $\widetilde{\Pi}, \Pi$ be orthogonal projectors, and let $W : \mathrm{img}(\Pi) \mapsto \mathrm{img}\left(\widetilde{\Pi}\right)$ be an isometry, such that*

$$\left\| \sin\left(\frac{\pi}{2n}\right) W|\psi\rangle - \widetilde{\Pi} U |\psi\rangle \right\| \leq \varepsilon \tag{39}$$

[13] Note that we could also easily derive a fixed-point version of oblivious amplitude amplification based on Theorem 26, but we state the usual version instead for readability.

for all $|\psi\rangle \in \mathrm{img}(\Pi)$. Then we can construct a unitary $\tilde{U}$ such that for all $|\psi\rangle \in \mathrm{img}(\Pi)$

$$\left\| W|\psi\rangle - \widetilde{\Pi}\tilde{U}|\psi\rangle \right\| \leq 2n\varepsilon,$$

which uses a single ancilla qubit, with n uses of U and $U^\dagger$, n uses of $C_\Pi NOT$ and n uses of $C_{\widetilde{\Pi}} NOT$ gates and n single qubit gates.

Proof. First we prove the $\varepsilon = 0$ case. We prove this case by reproducing the polynomials stemming from ordinary amplitude amplification. Let $T_n \in \mathbb{R}[x]$ be the degree-n Chebyshev polynomial of the first kind. As discussed after Corollary 8 there is an easy to describe $\Phi \in \mathbb{R}^n$ which corresponds to T_n in equation (14).

Now observe that by (39) we have that $\widetilde{\Pi}U\Pi = \sin(\frac{\pi}{2n})W$. We can apply singular value transformation using T_n to obtain U_Φ such that

$$\widetilde{\Pi}U_\Phi\Pi = T_n\Big(\sin\Big(\frac{\pi}{2n}\Big)\Big)W = T_n\Big(\cos\Big(\frac{\pi}{2} - \frac{\pi}{2n}\Big)\Big)W = \cos\Big(\frac{n-1}{2}\pi\Big)W = \pm W.$$

After correcting the global phase ± 1 (which depends on the parity of $(n-1)/2$) we get $\tilde{U} := \pm U_\Phi$ such that for all $|\psi\rangle \in \mathrm{img}(\Pi)$ we have $\tilde{U}|\psi\rangle = W|\psi\rangle$. The complexity statement follows from Lemma 19.

In the $\varepsilon \neq 0$ case we first handle some trivial cases. If $n = 1$ or $\varepsilon > \frac{1}{3}$ we simply take $\tilde{U} := U$. Otherwise if $n \geq 3$ and $\varepsilon \in [0, \frac{1}{3}]$ the error bounds follow from Lemma 23, in the following way: Let $A := \sin(\frac{\pi}{2n})W$ and let $\tilde{A} := \widetilde{\Pi}U\Pi$, by (39) we have that $\left\|A - \tilde{A}\right\| \leq \varepsilon$. Then

$$\left\|A + \tilde{A}\right\| \leq \|A\| + \|A\| + \left\|\tilde{A} - A\right\| = 2\sin\Big(\frac{\pi}{2n}\Big) + \varepsilon \leq 2\sin\Big(\frac{\pi}{6}\Big) + \frac{1}{3} = \frac{4}{3},$$

thus $\left\|\frac{A+\tilde{A}}{2}\right\|^2 \leq \frac{4}{9}$ and $\left\|A - \tilde{A}\right\| + \left\|\frac{A+\tilde{A}}{2}\right\|^2 \leq \frac{7}{9} < 1$. This also implies that $\sqrt{\frac{2}{1-\left\|\frac{A+\tilde{A}}{2}\right\|^2}} \leq \sqrt{\frac{2}{1-\frac{4}{9}}} = \sqrt{\frac{18}{5}} < 2$, and therefore by Lemma 23 we get that $\left\|W - \widetilde{\Pi}\tilde{U}\Pi\right\| \leq 2n\varepsilon$. □

Now we turn to solving the linear singular value amplification problem. That is, given a matrix in a projected encoding form, construct a projected encoding of a matrix which have singular values that are γ times larger than the original singular values.

In order to proceed we first construct some polynomials similarly that can be used in combination with our singular value transformation results.

Lemma 29 (Polynomial approximations of the rectangle function)**.** *Let $\delta', \varepsilon' \in (0, \frac{1}{2})$ and $t \in [-1, 1]$. There exist an even polynomial $P' \in \mathbb{R}[x]$ of degree $\mathcal{O}\big(\log(\frac{1}{\varepsilon'})/\delta'\big)$, such that $|P'(x)| \leq 1$ for all $x \in [-1, 1]$, and*

$$\begin{cases} P'(x) \in \quad [0, \varepsilon'] & \text{for all } x \in [-1, -t-\delta'] \cup [t+\delta', 1], \text{ and} \\ P'(x) \in [1-\varepsilon', 1] & \text{for all } x \in [-t+\delta', t-\delta']. \end{cases} \tag{40}$$

Proof. First let us take a real polynomial P which $\frac{\varepsilon'}{2}$-approximates the sign function on the interval $[-2, 2] \setminus (-\delta', \delta')$, moreover for all $x \in [-2, 2]$: $|P(x)| \leq 1$. Such a polynomial of degree $\mathcal{O}\big(\frac{1}{\delta'}\log\big(\frac{1}{\varepsilon'}\big)\big)$ can be efficiently constructed by Lemma 25. Now take the polynomial

$$P'(x) := (1-\varepsilon')\frac{P(x+t) + P(-x+t)}{2} + \varepsilon'.$$

It is easy to see that by construction $P'(x)$ is an even polynomial of degree $\mathcal{O}\big(\frac{1}{\delta'}\log\big(\frac{1}{\varepsilon'}\big)\big)$. Moreover $|P'(x)|\leq 1$ for all $x\in[-1,1]$ and (40) also holds. □

Now we prove our result about uniform singular value amplification, which is a common generalization of the results of Low and Chuang [LC17a, Theroems 2,8].

Theorem 30 (Uniform singular value amplification)**.** *Let $U,\Pi,\widetilde{\Pi}$ be as in Theorem 17, let $\gamma>1$ and let $\delta,\varepsilon\in(0,\frac{1}{2})$. Suppose that $\widetilde{\Pi}U\Pi=W\Sigma V^\dagger=\sum_i\varsigma_i|w_i\rangle\langle v_i|$ is a singular value decomposition. Then there is an $m=\mathcal{O}\big(\frac{\gamma}{\delta}\log\big(\frac{\gamma}{\varepsilon}\big)\big)$ and an efficiently computable $\Phi\in\mathbb{R}^m$ such that*[14]

$$\left(\langle+|\otimes\widetilde{\Pi}_{\leq\frac{1-\delta}{\gamma}}\right)U_\Phi\left(|+\rangle\otimes\Pi_{\leq\frac{1-\delta}{\gamma}}\right)=\sum_{i\colon\varsigma_i\leq\frac{1-\delta}{\gamma}}\tilde{\varsigma}_i|w_i\rangle\langle v_i|,\quad\textit{where }\left\|\frac{\tilde{\varsigma}_i}{\gamma\varsigma_i}-1\right\|\leq\varepsilon. \tag{41}$$

Moreover, U_Φ can be implemented using a single ancilla qubit with m uses of U and $U^\dagger$, m uses of $C_\Pi NOT$ and m uses of $C_{\widetilde{\Pi}}NOT$ gates and m single qubit gates.

Proof. Let us set in Lemma 29 $t:=\frac{1-\delta/2}{\gamma}$, $\delta':=\frac{\delta}{2\gamma}$ and $\varepsilon':=\frac{\varepsilon}{\gamma}$ in order to get an even polynomial P of degree $\mathcal{O}\big(\frac{\gamma}{\delta}\log\big(\frac{\gamma}{\varepsilon}\big)\big)$ that is an $\frac{\varepsilon}{\gamma}$-approximation of the rectangle function. Let $P_{\Re}(x):=\gamma\cdot x\cdot P(x)$, which is an odd polynomial of degree $m=\mathcal{O}\big(\frac{\gamma}{\delta}\log\big(\frac{\gamma}{\varepsilon}\big)\big)$. It is easy to see that $P_{\Re}$ approximates the linear function $\gamma\cdot x$ with ε-multiplicative-precision on the domain $\left[\frac{-1+\delta}{\gamma},\frac{1-\delta}{\gamma}\right]$, and observe that $|P_{\Re}(x)|\leq 1$ for all $x\in[-1,1]$, thereby it satisfies the requirements of Corollary 18. We use singular value transformation Corollary 18 to construct a $\Phi\in\mathbb{R}^m$ such that

$$(\langle+|\otimes\widetilde{\Pi})U_\Phi(|+\rangle\otimes\Pi)=P_{\Re}^{(SV)}\Big(\widetilde{\Pi}U\Pi\Big)=\sum_i P_{\Re}(\sigma_i)|w_i\rangle\langle v_i|$$

which shows that equation (41) is satisfied because $\frac{P_{\Re}(x)}{\gamma\cdot x}$ is ε-close to 1 on the domain $\left[\frac{-1+\delta}{\gamma},\frac{1-\delta}{\gamma}\right]$. We conclude the gate complexity using Lemma 19. □

Finally, note that if $\|\Sigma\|\leq\frac{1-\delta}{\gamma}$ in the above theorem then we get that

$$\Big\|\gamma\widetilde{\Pi}U\Pi-(\langle+|\otimes\widetilde{\Pi})U_\Phi(|+\rangle\otimes\Pi)\Big\|\leq\varepsilon,$$

thereby this procedure gives an efficient way to magnify a projected unitary encoding.

3.5 Singular value discrimination, quantum walks and the fast OR lemma

First we show how to efficiently implement approximate singular value threshold projectors, which will be the main tool of this section.

Theorem 31 (Implementing singular value threshold projectors)**.** *Let $U,\Pi,\widetilde{\Pi}$ be as in Theorem 17 and let $t,\delta>0$. Suppose that $\widetilde{\Pi}U\Pi=W\Sigma V^\dagger$ is a singular value decomposition. Then there is an $m=\mathcal{O}\Big(\frac{\log(1/\varepsilon)}{\delta}\Big)$ and a $\Phi\in\mathbb{R}^m$ such that we have $\|\Pi_{\geq t+\delta}U_\Phi\Pi_{\geq t+\delta}-\Pi_{\geq t+\delta}\|\leq\varepsilon$, and*[14] *$\|(\langle+|\otimes\Pi_{\leq t-\delta})U_\Phi(|+\rangle\otimes\Pi_{\leq t-\delta})\|\leq\varepsilon$. Moreover, U_Φ can be implemented using a single ancilla qubit with m uses of U and $U^\dagger$, m uses of $C_\Pi NOT$ and m uses of $C_{\widetilde{\Pi}}NOT$ gates and m single qubit gates.*

[14] Here we implicitly assumed that U_Φ is implemented as in Figure 1, with the phase gates as in Figure 1b and the the $|+\rangle$ ancilla state corresponds to the ancilla qubit in Figure 1b.

Proof. By Lemma 29 we can construct an even polynomial $P_{\Re} \in \mathbb{R}[x]$ of degree $m = \mathcal{O}\Big(\frac{\log(1/\varepsilon^2)}{\delta}\Big)$ that approximates the rectangle function with $\varepsilon^2/2$ precision on the domain $[-1,1] \setminus (-t-\delta, -t+\delta) \cup (t-\delta, t+\delta)$. By Corollary 10 we know that there exists a polynomial P of the same degree as $P_{\Re}$ such that $\Re[P] = P_{\Re}$, moreover P satisfies the conditions of Corollary 8. Use singular value transformation Theorem 17 to construct a $\Phi \in \mathbb{R}^m$ such that $\widetilde{\Pi} U_\Phi \Pi = P^{(SV)}\Big(\widetilde{\Pi} U \Pi\Big)$ up to precision ε and observe that $\|\Pi_{\geq t+\delta} U_\Phi \Pi_{\geq t+\delta} - \Pi_{\geq t+\delta}\| \leq \varepsilon$, and $\|(\langle +| \otimes \Pi_{\leq t-\delta}) U_\Phi (|+\rangle \otimes \Pi_{\leq t-\delta})\| \leq \varepsilon$. Conclude the gate complexity using Lemma 19. □

Note that the above complexity can be improved up to quadratically in terms of scaling with δ, when the threshold t is close to 1, see, e.g., Lemma 35. For the error of the optimal polynomial approximation of the step function see the results of Eremenko and Yuditskii [EY11].

We define the singular value discrimination problem as to find out whether a given quantum state has singular value at most a or it at least b. As we indicated above whenever a and b are $\mathcal{O}(|a-b|)$ close to 1 we can get a quadratic improvement. A simple way to achieve this quadratic improvement is to perform singular value projection on the complementary singular values rather than on the original ones, by replacing the matrix $\widetilde{\Pi} U \Pi$ by the complementary projection $(I - \widetilde{\Pi}) U \Pi$.

Theorem 32 (Efficient singular value discrimination)**.** *Let $0 \leq a < b \leq 1$, and let $A = \widetilde{\Pi} U \Pi$ be a projected unitary encoding. Let $|\psi\rangle$ be a given unknown quantum state, with the promise that $|\psi\rangle$ is a right singular vector of A with singular value at most a or at least b. Then we can distinguish the two cases with error probability at most ε using singular value transformation of degree*

$$\mathcal{O}\left(\frac{1}{\max[b-a, \sqrt{1-a^2} - \sqrt{1-b^2}]} \log\left(\frac{1}{\varepsilon}\right)\right).$$

Moreover, if $a = 0$ or $b = 1$ we can make the error one sided.

Proof. Let us assume that $b - a \geq \sqrt{1-a^2} - \sqrt{1-b^2}$. First we apply an $\sqrt{\varepsilon}$-approximate singular value projector on $|\psi\rangle$ using Theorem 31, with choosing $t := \frac{a+b}{2}$ and $\delta := \frac{b-a}{2}$, at the end measuring the projector $|+\rangle\langle +| \otimes \Pi$. If we find the state in the image of $|+\rangle\langle +| \otimes \Pi$ we conclude that the singular value is at least b, otherwise we conclude that it is at most a. The correctness and the complexity follows from Theorem 31. If $a = 0$ then we make the error one-sided by using singular vector transformation Theorem 26 with setting $\delta := b$, and measuring $\widetilde{\Pi}$ at the end. Similarly as before if we find the state in the image of $\widetilde{\Pi}$ we conclude that the singular value is at lest b, otherwise we conclude that it is 0. The error becomes one-sided because Theorem 26 uses an odd-degree singular value transformation which always preserves 0 singular values.

The proof of the $b - a < \sqrt{1-a^2} - \sqrt{1-b^2}$ case works analogously just changing $\widetilde{\Pi}$ to $\Pi' := I - \widetilde{\Pi}$ in the proof, which leads to $A' := \Pi' U \Pi$. It is easy to see by Lemma 12 that $|\psi\rangle$ is a singular vector of A' with singular value at least $\sqrt{1-a^2}$ in the first case or with singular value at most $\sqrt{1-b^2}$ in the second case. Also if $b = 1$ we can make the error one sided since then the corresponding singular value of $|\psi\rangle$ with respect to A' is 0.

Finally note that if $a = 0$, then $b - a = b \geq 1 - \sqrt{1-b^2} = \sqrt{1-a^2} - \sqrt{1-b^2}$, and if $b = 1$, then $b - a = 1 - a \leq \sqrt{1-a^2} = \sqrt{1-a^2} - \sqrt{1-b^2}$, therefore we covered all cases. □

The above result can also be used when the input state is promised to be a superposition of singular values, with the promised bounds. Also in order to distinguish the two cases with constant success probability it is enough if most of the amplitude is on singular vectors with singular vectors satisfying the promise.

3.5.1 Relationship to quantum walks

Now we show how to quickly derive the quadratic speed-ups of Markov chain based search algorithms using our singular value transformation and discrimination results. Before doing so we introduce some definitions and notation for Markov Chains.

Let $\mathcal{P} \in \mathbb{R}^{n\times n}$ be a time-independent Markov chain on discrete state space X with $|X| = n$, which sends an element $x \in X$ to $y \in X$ with probability p_{xy}, thereby $\mathcal{P}$ is a row-stochastic matrix. We say that $\mathcal{P}$ is *ergodic* if for a large enough $t \in \mathbb{N}$ all elements of $\mathcal{P}^t$ are non-zero. For an ergodic $\mathcal{P}$ there exists a unique stationary distribution π such that $\pi\mathcal{P} = \pi$, and we define the *time-reversed* Markov chain as $\mathcal{P}^* := \mathrm{diag}(\pi)^{-1} \cdot \mathcal{P}^T \cdot \mathrm{diag}(\pi)$. We say that $\mathcal{P}$ is *reversible* if it is ergodic and $\mathcal{P}^* = \mathcal{P}$. For an ergodic Markov chain $\mathcal{P}$ we define the discriminant matrix $D(\mathcal{P})$ such that its xy entry is $\sqrt{p_{xy}p^*_{yx}}$, where p^*_{yx} stands for entries of the time-reversed chain. It is easy to see that

$$D(\mathcal{P}) = \mathrm{diag}(\pi)^{\frac{1}{2}} \cdot \mathcal{P} \cdot \mathrm{diag}(\pi)^{-\frac{1}{2}}.$$

This form has several important consequences. First of all the spectrum of $\mathcal{P}$ and $D(\mathcal{P})$ coincide, moreover the vector $\sqrt{\pi}$, that we get from π by taking the square root element-wise, is a left eigenvector of $D(\mathcal{P})$ with eigenvalue 1. Also from the definition $\sqrt{p_{xy}p^*_{yx}}$ of the xy entry it follows that for reversible Markov chains $D(\mathcal{P})$ is a symmetric matrix, therefore its singular values and eigenvalues coincide after taking their absolute value.

In the literature [Sze04, MNRS11, KMOR16] quantum walk based search methods are usually analyzed with the help of this discriminant matrix. Here we directly use the discriminant matrix as opposed to the associated quantum walk, significantly simplifying the analysis. Before deriving our versions of the Markov chain speed-up results we introduce some definitions regarding sets of marked elements.

For a set of marked element $M \subseteq X$, we denote by $D_M(\mathcal{P})$ the matrix that we get after setting to zero the rows and columns of $D(\mathcal{P})$ corresponding to the marked elements. For an ergodic Markov chain $\mathcal{P}$ we define the hitting time $HT(\mathcal{P}, M)$ as the expected number of step of the Markov chain before reaching the first marked element, if started from the stationary[15] distribution π. We denote the probability that an element is marked in the stationary distribution by $p_M := \sum_{x\in M} \pi_x$. Now we invoke some results about the connection between the hitting time and the discriminant matrix, which are proven for example in [KMOR16, Proposition 2] and [Gil14, Lemma 10].

Lemma 33 (Relationship between the hitting time and the discriminant matrix)**.** *Let $\mathcal{P}$ be a reversible Markov chain and M a set of marked elements. Let (v_i, λ_i) be the eigenvector-eigenvalue pairs of $D_M(\mathcal{P})$, then*

$$HT(\mathcal{P}, M) = \sum_{i=1}^{n} \frac{|\langle v_i, \sqrt{\pi}\rangle|^2}{1-\lambda_i} - p_M, \qquad \textit{and} \qquad \sum_{i=1}^{n} \frac{|\langle v_i, \sqrt{\pi}\rangle|^2}{1-|\lambda_i|} \leq 2\sum_{i=1}^{n} \frac{|\langle v_i, \sqrt{\pi}\rangle|^2}{1-\lambda_i} \tag{42}$$

The following result shows how the presence of marked elements can be detected quadratically faster using singular value discrimination compared to using the corresponding classical Markov chain. A slightly less general version of this result was proven by Szegedy [Sze04].

[15] Here we follow the convention of Szegedy [Sze04, Equation (15)], and define hitting time without conditioning on the stationary distribution to unmarked vertices.

Corollary 34 (Detecting marked elements in a reversible Markov chain)**.** *Let $\mathcal{P}$ be a reversible Markov chain, and $M \subseteq X$ a set of marked elements. Let U be a unitary and $\tilde{\Pi}, \Pi$ orthogonal projectors. Suppose that $B, \tilde{B}$ are orthogonal bases, such that representing the matrix of $\tilde{\Pi}U\Pi$ in the bases $B \to \tilde{B}$ we have that*

$$\tilde{\Pi}U\Pi = \begin{bmatrix} D_M(\mathcal{P}) & 0 \\ 0 & . \end{bmatrix}.$$

Suppose that we are given a copy of $|\pi\rangle := \sum_{x\in X} \sqrt{\pi_x}|x\rangle$, where $|x\rangle\colon x \in X$ are the first n basis elements in B. Then we can distinguish with constant one sided error the case $\mathrm{HT}(P, M) \leq K$ from the case $M = \emptyset$ (i.e., $\mathrm{HT}(P, M) = \infty$) with singular value transformation of degree $\mathcal{O}\big(\sqrt{K+1}\big)$.

Proof. Suppose that $M \neq \emptyset$. Let $(|v_i\rangle, \lambda_i)\colon i \in [n]$ be the eigenvector and eigenvalue pairs of the $D_M(\mathcal{P})$ block of $\tilde{\Pi}U\Pi$. By Lemma 33 we have that

$$\sum_{i=1}^{n} \frac{|\langle v_i|\pi\rangle|^2}{1 - |\lambda_i|} \leq 2\mathrm{HT}(P, M) + 2p_M \leq 2(K + 1).$$

By Markov's inequality we have that

$$\sum_{i\colon |\lambda_i| \geq 1 - \frac{1}{12(K+1)}} |\langle v_i|\pi\rangle|^2 \leq \frac{1}{6},$$

and so $\left\|\Pi_{\leq 1-\frac{1}{12(K+1)}}|\pi\rangle\right\|^2 \geq \frac{5}{6}$. On the other hand if $M = \emptyset$, then $D_M(\mathcal{P}) = D(\mathcal{P})$, and $\|D(\mathcal{P})|\pi\rangle\| = 1$. Therefore we can apply our singular value discrimination result Theorem 32 to distinguish the two cases $M = \emptyset$ and $\mathrm{HT}(P, M) \leq K$ using singular value transformation of degree $\mathcal{O}\big(\sqrt{K+1}\big)$. □

The above result shows how to detect the presence of marked elements quadratically faster than the classical hitting time. In practice one usually also wants to find a marked element, and quantum walks are also good at solving this problem. In order to show the connection to the literature of quantum walk based search algorithms we define some additional notation.

First we define the standard implementation procedures for Markov chains together with their associated costs following Magniez et al. [MNRS11]. We slightly generalize the usual approach fitting our singular value transformation framework. Let us fix an orthogonal basis B, such that the first n elements of B are labeled by $|x\rangle_d\colon x \in X$. We define the following costs an operations with their matrices represented in the basis B.

U : Update cost U. The cost of implementing $\mathrm{C}_\Pi\mathrm{NOT}$ and U gates such that

$$\Pi U\Pi = \begin{bmatrix} D(\mathcal{P}) & 0 \\ 0 & . \end{bmatrix}. \tag{43}$$

C : Checking cost C. The cost of implementing a $\mathrm{C}_{\Pi_M}\mathrm{NOT}$ gate such that for all $x \in M\colon \Pi_M|x\rangle_d = |x\rangle_d$ and for all $x \in X \setminus M\colon \Pi_M|x\rangle_d = 0$. This implies that

$$(I - \Pi_M)\Pi U\Pi(I - \Pi_M) = \begin{bmatrix} D_M(\mathcal{P}) & 0 \\ 0 & . \end{bmatrix}.$$

S : Setup cost S. The cost of preparing the stationary state in basis B: $|\pi\rangle := \sum_{x\in X}\sqrt{\pi_x}|x\rangle_d$.

First we would like to describe how these operators are usually implemented in the literature. Usually $\mathcal{P}$ is represented using basis elements $|x\rangle_d = |0\rangle|x\rangle|d_x\rangle$, where the $|d_x\rangle$ register stores some data associated with the vertex x, which enables efficient implementation of the update procedure. The unitary U is usually implemented using a product of state preparation unitaries $U = U_L^\dagger U_R$:

$$U_R : |0\rangle|x\rangle|d_x\rangle \to \sum_{y\in[n]} \sqrt{p_{xy}}|x\rangle|y\rangle|d_{xy}\rangle \qquad \forall x \in X \tag{44}$$

$$U_L : |0\rangle|y\rangle|d_y\rangle \to \sum_{x\in[n]} \sqrt{p^*_{yx}}|x\rangle|y\rangle|d_{xy}\rangle \qquad \forall y \in X \tag{45}$$

We assume for simplicity that $0 \notin X$, resulting in a helpful "free" label, and also let us assume that the first register is $n+1$ dimensional and the second register is n dimensional. If the third register is one-dimensional (i.e., we can just trivially omit it), by equations (44)-(45) we get that

$$(|0\rangle\langle 0| \otimes I)U(|0\rangle\langle 0| \otimes I) = \begin{bmatrix} D(\mathcal{P}) & 0 \\ 0 & 0 \end{bmatrix}.$$

If the data structure register is non-trivial, we can still conclude that

$$(|0\rangle\langle 0| \otimes I)U(|0\rangle\langle 0| \otimes I) = \begin{bmatrix} D(\mathcal{P}) & \cdot \\ \cdot & \cdot \end{bmatrix},$$

however we need a slightly stronger assumption about U. We assume that U_L, U_R are implemented such that the block-matrices next to $D(\mathcal{P})$ are 0 as in (43). This is implicitly assumed[16] in [MNRS11], and a sufficient condition is presented in the work of Childs et al. [CJKM13].

Before solving the above problem, we invoke a useful polynomial approximation result due to Dolph [Dol46].

Lemma 35 (Optimal polynomial approximation of a windowing function on a bounded interval)**.** *For all $\varepsilon \in (0,1]$ and $n \in \mathbb{N}$ we have that*[17]

$$\underset{P\in\mathbb{R}[x]}{\operatorname{argmax}}\Big(\max\big\{\lambda\colon \|P(x)\|_{[-\lambda,\lambda]} \leq \varepsilon\big\}\Big) = T_n(xT_{1/n}(1/\varepsilon)),$$

where argmax *is over all real degree-n polynomials satisfying $\|P\|_{[-1,1]} \leq 1$, and $P(\pm 1) = (\pm 1)^n$. Moreover, for any $\delta \in (0,1)$ for some $n = \mathcal{O}\Big(\frac{1}{\sqrt{\delta}}\log(\frac{1}{\varepsilon})\Big)$ we have that*

$$\big\|T_n(xT_{1/n}(1/\epsilon))\big\|_{[-1+\delta,1-\delta]} \leq \varepsilon.$$

Notably, the phase sequence required to implement this polynomial using quantum signal processing is expressed in closed-form in the work of Yoder et al. [YLC14].

[16] This assumption is necessary for the correctness of the analysis in [MNRS11], however it is not explicitly stated.

[17] The standard generalization of Chebyshev polynomials to non-integer degree y is $T_y(x) \equiv \cosh(y\,\mathrm{arccosh}(x)) \equiv \cos(y\arccos(x))$.

Theorem 36 (Quadratic speed-up for finding marked elements of a Markov chain)**.** *Let $\mathcal{P}$ be a reversible Markov chain, such that the singular value gap*[18] *of $D(\mathcal{P})$ is at least δ, and the set of marked elements M is such that $p_M \geq \varepsilon$. Then we can find a marked element with high probability in complexity of order $\mathsf{S} + \frac{1}{\sqrt{\varepsilon}}\left(\mathsf{C} + \sqrt{\frac{1}{\delta}}\log(\frac{1}{\varepsilon})\mathsf{U}\right)$.*

Proof. First we apply singular value transform on $\Pi U \Pi$ using an ε-approximation of the zero-function given by Lemma 35 in order to get U_Φ with all $\neq 1$ singular values of $D(\mathcal{P})$ shrunk below a level of $\mathcal{O}(\varepsilon)$. Then the top-left block of $\Pi U_\Phi \Pi$ is $\mathcal{O}(\varepsilon)$-close to $|\pi\rangle\langle\pi|$, and the implementation of U_Φ has complexity $\mathcal{O}\left(\sqrt{\frac{1}{\delta}}\log(\frac{1}{\varepsilon})\mathsf{U}\right)$. We pretend that the top-left block is $|\pi\rangle\langle\pi|$, in which case we seem to have that $\Pi_M \Pi U_\Phi \Pi = \sqrt{p_M}|\pi_M\rangle\langle\pi|$, where $|\pi_M\rangle := \frac{\sum_{x\in M}\sqrt{\pi_x}|x\rangle_d}{\sqrt{p_M}}$. Then we apply singular vector transform to get a constant approximation of $|\pi_M\rangle\langle\pi|$ in the top-left block, and apply it to the state $|\pi\rangle$ in order to find a marked element with high probability. Finally, we use the robustness of singular value transformation Lemma 22 to show that we can indeed dismiss the $\mathcal{O}(\varepsilon)$ discrepancy between $\Pi U_\Phi \Pi$ and $|\pi\rangle\langle\pi|$. □

Note that the above algorithm is simpler and more efficient than the phase estimation based algorithm of [MNRS11]. However note, that Magniez et al. [MNRS11] showed how to remove the $\log(\frac{1}{\varepsilon})$ factor completely using a more involved procedure.

Finally note that it is known that a unique marked element can be found using a quantum walk quadratically faster than the hitting time. However, it is an open question whether in the presence of multiple marked elements the quadratic advantage can be retained.[19] For more details see the work of Krovi et al. [KMOR16]. They use an interpolated matrix between $D(\mathcal{P})$ and $D_M(\mathcal{P})$ – which is an idea very naturally fitting our framework, see for example Lemma 52. We believe that the algorithms of Krovi et al. [KMOR16] for finding marked elements can also be cast in our singular value transformation framework, but we leave the discussion of these algorithms for future work.

3.5.2 Fast QMA amplification and fast quantum OR lemma

We show how the fast QMA amplification result of Nagaj et al. [NWZ09] follows directly from our singular value discrimination results. In order to state the result we invoke the definition of the language class QMA.

Definition 37 (The language class QMA)**.** *Let $L \subseteq \{0,1\}^*$ be a language of yes and no instances $L = L_{yes} \dot{\cup} L_{yes}$. The language L belongs to the class QMA if there exists a uniform family of quantum verifier circuits V working on $n = \mathrm{poly}(|x|)$ qubits using $m = \mathrm{poly}(|x|)$ ancillae and two numbers $0 \leq b < a \leq 1$ satisfying $\frac{1}{a-b} = \mathcal{O}(\mathrm{poly}(|x|))$ such that for all x in*

L_{yes} : *there exists an n-qubit witness $|\psi\rangle$ such that upon measuring the state $V|\psi\rangle|0\rangle^m$ the probability of finding the first qubit in state $|1\rangle$ has probability at least a.*

[18] We define the singular value gap as the difference between the two largest singular values. For a reversible Markov chain the singular values of $D(\mathcal{P})$ are the same as the absolute values of the eigenvalues of $\mathcal{P}$. Note however, that this is not strictly necessary, we could work with the eigenvalues too using eigenvalue transformation results, such as Theorem 56.

[19] One can show that $p_M = \Omega\left(\frac{1}{HT(\mathcal{P},M)}\right)$, therefore using amplitude amplification one can find a marked element quadratically faster, however with a large S cost. The appeal of the quantum walk algorithms is that they only use the setup procedure a very few times.

L_{no} : *for any n-qubit state $|\phi\rangle$ upon measuring the state $V|\phi\rangle|0\rangle^m$ the probability of finding the first qubit in state $|1\rangle$ has probability at most b.*

Now we are ready to reprove the result of Nagaj et al. [NWZ09]:

Theorem 38 (Fast QMA amplification). *Suppose that we have a language in* QMA *as in Definition 37. We can modify the verifier circuit V such that the acceptance probabilities become $a' := 1-\varepsilon$ and $b' := \varepsilon$ using singular value transformation of degree $\mathcal{O}\Big(\frac{1}{\max[\sqrt{a}-\sqrt{b},\sqrt{1-b}-\sqrt{1-a}]}\log\big(\frac{1}{\varepsilon}\big)\Big)$.*

Proof. Observe that by Definition 37 for all $x \in L_{\text{yes}}$ we have that

$$\left\|(|1\rangle\langle 1| \otimes I_{n+m-1})V\left(I_n \otimes |0\rangle\langle 0|^{\otimes m}\right)\right\| \geq \sqrt{a},$$

and for all $x \in L_{\text{no}}$ we have that

$$\left\|(|1\rangle\langle 1| \otimes I_{n+m-1})V\left(I_n \otimes |0\rangle\langle 0|^{\otimes m}\right)\right\| \leq \sqrt{b}.$$

After applying a singular value discrimination circuit for discriminating singular values below $\sqrt{b}$ and above $\sqrt{a}$ we get a circuit that in the former case accepts some witness $|\psi\rangle$ with probability at least $1-\varepsilon$ and in the latter case rejects every state $|\phi\rangle$ with probability at least $1-\varepsilon$. □

Finally we show how to quickly derive a slightly improved version of the fast quantum OR lemma of Brandão et al. [BKL+17]. We use the main ideas of the proof of the original quantum OR lemma of Harrow et al. [HLM17] in a way similar to the approach of Brandão et al. [BKL+17].

Theorem 39 (Fast quantum OR lemma). *Let $m \in \mathbb{N}$, let $\Pi_i : i \in [m]$ be orthogonal projectors and let $\eta, \nu \in (0, \frac{1}{2}]$. Suppose we are given one copy of a quantum state ρ with the promise that either*

(i) there exists some $i \in [m]$ such $\mathrm{Tr}[\rho\Pi_i] \geq 1-\eta$, or

(ii) $\frac{1}{m}\sum_{j=1}^m \mathrm{Tr}[\rho\Pi_j] \leq \nu$.

Suppose that we have access[20] to an operator V such that $(\langle i|\otimes I)V(|i\rangle\otimes I) = C_{\Pi_i}NOT$ for all $i \in [m]$. Then for all $\varepsilon \in (0, \frac{1}{2}]$ we can construct an algorithm which, in case (i) accepts ρ with probability at least $\frac{(1-\eta)^2}{4}-\varepsilon$, and in case (ii) it accepts ρ with probability at most $5m\nu+\varepsilon$. Moreover, the algorithm uses V and its inverse a total number of $\mathcal{O}\big(\sqrt{m}\log\big(\frac{1}{\varepsilon}\big)\big)$ times and uses $\mathcal{O}\big(\sqrt{m}\log(m)\log\big(\frac{1}{\varepsilon}\big)\big)$ other gates and $\mathcal{O}(\log(m))$ ancilla qubits.

Proof. Let us define $A := \frac{1}{m}\sum_{i=1}^m (I-\Pi_i)$. First observe that $I-\Pi_i = (|0\rangle\langle 0| \otimes I)\mathrm{C}_{\Pi_i}\mathrm{NOT}(|0\rangle\langle 0| \otimes I)$. Let $a := \lceil \log_2(m+1)\rceil + 1$ and let U be a unitary that implements the map $|0\rangle^{a-1} \to \frac{1}{\sqrt{m}}\sum_{i=1}^m |i\rangle$, and let us define $\tilde{V} := \big(U^\dagger \otimes I\big)V(U \otimes I)$ and $\Pi := |0\rangle\langle 0|^a \otimes I$. Then it is easy to see that $A = \Pi\tilde{V}\Pi$.

Now let $\lambda := \frac{1-\eta}{2m}$, in case (i) Harrow et al. [HLM17, Corollary 11] proved that

$$\mathrm{Tr}[\rho\Pi_{\leq 1-\lambda}] \geq (1-\eta)^2/4. \tag{46}$$

[20] Brandão et al. [BKL+17] assumes access to a multiply-controlled reflection operator instead of V, but it is easy to see that such an operator can be easily transformed to the operator required here using a single qubit by applying phase-kickback.

On the other hand in case (ii) we have that $\mathrm{Tr}[\rho A] \geq 1-\nu$. Using Markov's inequality we get

$$\mathrm{Tr}\left[\rho \Pi_{\leq 1-\frac{4}{5}\lambda}\right] \leq \frac{\nu}{\frac{4}{5}\lambda} = \frac{5m\nu}{2(1-\eta)} \leq 5m\nu. \tag{47}$$

Finally, we apply ε-precise singular value discrimination on ρ with $a := 1-\lambda$ and $b := 1-\frac{4}{5}\lambda$. The correctness follows from (46)-(47) and Theorem 32. The complexity statement follows from Theorem 32, Lemma 19 and the fact that U can be implemented using $\mathcal{O}(\log(m))$ one- and two-qubit gates. □

3.6 "Non-commutative measurements" and singular value estimation

Preparing ground states of local Hamiltonians is a notoriously hard problem. However, under the conditions of the quantum Lovász Local Lemma, the local Hamiltonian is guaranteed to be frustration-free as shown by Ambainis et al. [AKS12]. As shown by Sattath and Arad [SA15] and Schwarz et al. [SCV13] under same conditions the problem becomes efficiently solvable when the local Hamiltonian terms commute. The non-commuting case is more difficult, but under a gap-promise Gilyén and Sattath [GS17] showed that a ground state can be efficiently prepared.

Gilyén and Sattath [GS17] essentially reduced the state preparation problem to solving the following task: Given two (non-commuting) orthogonal projectors Π^F and Π_c and quantum state $|\psi\rangle \in \mathrm{img}(\Pi^F)$ perform a "non-commutative" measurement in the following sense. If $|\psi\rangle \in \mathrm{img}(\Pi^F) \cap \ker(\Pi_c)$ then output 0 and leave the state intact, otherwise if $|\psi\rangle$ is a right singular vector of $\Pi_c\Pi^F$ with singular value greater than 0, then output 1 and "rotate" the state $|\psi\rangle$ to the corresponding left singular vector of $\Pi_c\Pi^F$ which in turn lies in $\mathrm{img}(\Pi_c)$. They showed how to implement such a quantum channel using a combination of weak measurements and the quantum Zeno effect, however their quantum channel does not preserve coherence between singular vectors with different singular values. The complexity of their implementation is essentially $\mathcal{O}\left(\frac{\log(1/\varepsilon)}{\varepsilon\varsigma^2}\right)$, where ς is the smallest non-zero singular value of $\Pi_c\Pi^F$, and ε is the desired maximum failure probability.

Gilyén and Sattath [GS17] called exact quantum channel the procedure which solves the above problem, but also preserves coherence between singular vectors with different singular values. In their paper it was unclear how to efficiently implement such a "non-commutative" measurement. However note, that the techniques developed in this paper result in an efficient implementation. Indeed, by setting $A := \Pi_c\Pi^F$ this task can be solved with maximal failure probability ε using singular value transformation Theorem 26, with $\mathcal{O}\left(\frac{\log(1/\varepsilon)}{\varsigma}\right)$ uses of $\mathrm{C}_{\Pi^F}\mathrm{NOT}$, $\mathrm{C}_{\Pi_c}\mathrm{NOT}$ and other two-qubit gates. This improves the ε dependence exponentially and improves the ς dependence quadratically, while solves a qualitatively stronger problem. These improvements also greatly improve the final complexity of the main algorithm presented in [GS17].

Finally, we turn to the singular value estimation results of Kerenidis an Prakash [KP17b]. Kerenidis and Prakash mostly use singular value estimation in order to implement singular vector projectors, with similar complexity to that of Theorem 31. However, to our knowledge, there is an unresolved issue in their implementation procedure, stemming from the ambiguity of the phase labels produced by phase estimation. Using our techniques combined with ideas of Chakraborty et al. [CGJ18], we show an alternative approach to singular value estimation.

Suppose that $A = \widetilde{\Pi}U\Pi$, and we would like to perform singular value estimation of A. The main idea is to first implement an operator V such that $(I \otimes \Pi)V(I \otimes \Pi) = \sum_{t=0}^{2^n-1} |t\rangle\langle t| \otimes T_{2t}^{(SV)}(A)$.

This can be done by using controlled quantum walk steps, i.e., using controlled alternating phase modulation sequences with phases as in Lemma 9. Suppose that $|\psi_j\rangle \in \mathrm{img}(\Pi)$ is a right singular vector of A with singular value $\cos(\theta_j)$, then

$$(I\otimes\Pi)V\big(H^{\otimes n}\otimes I\big)|0\rangle^n|\psi_j\rangle = (I\otimes\Pi)V\frac{1}{\sqrt{2^n}}\sum_{t=0}^{2^n-1}|t\rangle|\psi_j\rangle = \frac{1}{\sqrt{2^n}}\sum_{t=0}^{2^n-1}\cos(2t\theta_j)|t\rangle|\psi_j\rangle.$$

One can show that the norm of this state N_j is always bigger than some constant c. However, the problem is that the norm N_j depends on the singular value $\cos(\theta_j)$. Fortunately we can use singular vector transformation using the projected unitary encoding $(I\otimes\Pi)V(H^{\otimes n}\otimes I)(|0\rangle\langle 0|^{\otimes n}\otimes\Pi)$ by Theorem 26 in order to ε-approximate the map $|0\rangle^n|\psi_j\rangle \to \frac{1}{\sqrt{2^n}}\sum_{j=0}^{2^n-1}\frac{\cos(2t\theta_j)}{N_j}|t\rangle|\psi_j\rangle$. Applying a Fourier transform on the first (time) register, and taking half of the absolute value of the resulting estimation solves the singular value estimation problem. The correctness can be seen using the usual analysis of quantum phase estimation [CEMM98] by utilizing the identity $\cos(x) = \frac{e^{ix}+e^{-ix}}{2}$. For more details see [CGJ18, version 2].

3.7 Direct implementation of the Moore-Penrose pseudoinverse

Suppose $\widetilde{\Pi}U\Pi = A$ and $A = W\Sigma V^\dagger$ is a singular value decomposition. Then the pseudo-inverse of A is $A^+ = V\Sigma^+W^\dagger$, where Σ^+ contains the inverses of the diagonal elements of Σ, except for 0 entries which remain 0.

Now it is pretty straightforward to proceed using our singular value transformation methods. Suppose that all non-zero singular values are at least δ. Let $P_\Re$ be an odd real polynomial that ε-approximates the function $\delta/(2x)$ on the domain $[-1,1]\setminus(-\delta,\delta)$, then $P_\Re^{(SV)}(A^\dagger) = \Pi U_\Phi^\dagger\widetilde{\Pi}$ ε-approximates $\frac{\delta}{2}A^+$. The only thing remaining is to construct a relatively low-degree odd polynomial $P_\Re$ that achieves the desired approximation, and which is bounded by 1 in absolute value on $[-1,1]$, in order to be able to apply Corollary 18. Childs et al. [CKS17, Lemmas 17-19] constructed a useful polynomial for improving the HHL algorithm, which we can use after some adjustments.

Lemma 40. (Polynomial approximations of $1/x$, [CKS17, Lemmas 17-19]) *Let $\kappa > 1$ and $\varepsilon \in (0,\frac{1}{2})$. For $b = \lceil\kappa^2\log(\kappa/\varepsilon)\rceil$ the odd function*

$$f(x) := \frac{1-(1-x^2)^b}{x}$$

is ε-close to $1/x$ on the domain $[-1,1]\setminus(-\frac{1}{\kappa},\frac{1}{\kappa})$. Let $J := \left\lceil\sqrt{b\log(4b/\varepsilon)}\right\rceil$, then the $\mathcal{O}\big(\kappa\log(\frac{\kappa}{\varepsilon})\big)$-degree odd real polynomial

$$g(x) := 4\sum_{j=0}^{J}(-1)^j\left[\frac{\sum_{i=j+1}^{b}\binom{2b}{b+i}}{2^{2b}}\right]T_{2j+1}(x)$$

is ε-close to $f(x)$ on the interval $[-1,1]$, moreover $|P(x)| \le 4J = \mathcal{O}\big(\kappa\log(\frac{\kappa}{\varepsilon})\big)$ on this interval.

Theorem 41 (Implementing the Moore-Penrose pseudoinverse)**.** *Let $U,\Pi,\widetilde{\Pi}$ be as in Theorem 17 and let $0 < \varepsilon \le \delta \le \frac{1}{2}$. Suppose that $A = \widetilde{\Pi}U\Pi = W\Sigma V^\dagger$ is a singular value decomposition. Let*

$\Pi_{0,\geq\delta} := \Pi_{=0} + \Pi_{\geq\delta}$ *and similarly* $\widetilde{\Pi}_{0,\geq\delta} := \widetilde{\Pi}_{=0} + \widetilde{\Pi}_{\geq\delta}$. *Then there is an* $m = \mathcal{O}\big(\frac{1}{\delta}\log(\frac{1}{\varepsilon})\big)$ *and an efficiently computable* $\Phi \in \mathbb{R}^m$ *such that*[14]

$$\left\|\Big(\langle+|\otimes\Pi_{0,\geq\delta}\Big)U_\Phi\Big(|+\rangle\otimes\widetilde{\Pi}_{0,\geq\delta}\Big) - \Pi_{0,\geq\delta}\left(\frac{\delta}{2}\cdot A^+\right)\widetilde{\Pi}_{0,\geq\delta}\right\| \leq \varepsilon. \tag{48}$$

Moreover, U_Φ *can be implemented using a single ancilla qubit with* m *uses of* U *and* $U^\dagger$, m *uses of* $C_\Pi NOT$ *and* m *uses of* $C_{\widetilde{\Pi}}NOT$ *gates and* m *single qubit gates.*

Proof. Using Lemma 40 we can construct an odd polynomial $P(x)$ of degree $\mathcal{O}(\log(1/\varepsilon)/\delta)$ that $\frac{\varepsilon}{3}$-approximates the function $\frac{\delta}{2x}$ on the domain $[-1,1]\setminus(-\frac{\delta}{2},\frac{\delta}{2})$, and is less than 1 on this domain. Let us define $P_{\max} := \max_{x\in[-1,1]}|P(x)|$ and observe that $P_{\max} = \mathcal{O}\big(\log(\frac{1}{\varepsilon})\big)$. Let us also construct an even polynomial P' of degree $\mathcal{O}(\log(1/\varepsilon)/\delta)$ using Lemma 29 setting $t := \frac{3}{4}\delta$, $\delta' := \frac{\delta}{4}$ and $\varepsilon' := \min\Big(\frac{\varepsilon}{3},\frac{1}{P_{\max}}\Big)$ that ε'-approximates the rectangle function. Finally let $P_{\Re} := P\cdot(1-P')$, which is and odd real polynomial of degree $m = \mathcal{O}(\log(1/\varepsilon)/\delta)$. It is easy to see that $P_{\Re}$ ε-approximates $\frac{\delta}{2x}$ on the domain $[-1,1]\setminus(-\frac{\delta}{2},\frac{\delta}{2})$, moreover $P_{\Re}$ is bounded by 1 in absolute value on $[-1,1]$. Finally, we apply real singular value transformation on $A^\dagger = \Pi U^\dagger\widetilde{\Pi}$ using the polynomial $P_{\Re}$ by Corollary 18, and conclude the gate complexity using Lemma 19. □

Note that the $\varepsilon \leq \delta$ assumption in the above statement is quite natural, but it is not necessary, and can be removed by using our general polynomial approximation results, see Corollary 69.

3.8 Applications in quantum machine learning

The ability to transform singular values is central to the operation of many popular machine learning methods. Quantum machine learning methods such as quantum support vector machines [RML14], principal component analysis [LMR14, WK17], regression [HHL09, WBL12, CKS17, CGJ18], slow feature analysis [KL18], Gibbs sampling [CS17, AGGW17] and in turn training Boltzmann machines [AAR+18, KW17] all hinge upon this idea. These results are among the most celebrated in quantum machine learning, showing that singular value transformation has substantial impact on this field of quantum computing.

Many quantum algorithms for basic machine learning problems, such as ordinary least squares, weighted least squares, generalized least squares, were studied in a series of works [HHL09, WBL12, CKS17, CGJ18]. We do not examine these problems case-by-case, but point out that they can all be reduced to implementing the Moore-Penrose pseudoinverse and matrix multiplication, therefore they can be straightforwardly implemented by Theorem 41 and Lemma 53 (to be discussed in Subsection 4.4) within our framework.

We demonstrate how to apply our singular value transformation techniques to quantum machine learning by developing a new quantum algorithm for principal component regression. This machine learning algorithm is closely related to principal component analysis (PCA), which is a tool that is commonly used to reduce the effective dimension of a model by excising irrelevant features from it. The PCA method is quite intuitive, it simply involves computing the covariance matrix for a data set and then diagonalizing it. The eigenvectors of the covariance matrix then represent the independent directions of least or greatest variation in the data. Dimension reduction can be achieved by culling out any components that have negligibly small variation over them. This technique has many applications ranging from anomaly detection to quantitative finance. Quantum algorithms are known for this task and can lead to substantial speed-ups under appropriate assumptions about the data and the oracles used to provide it to the algorithm [LMR14, KLL+17].

Principal component regression is identical in spirit to principal component analysis. Rather than simply truncating small eigenvalues of the covariance matrix for a data set, principal component regression aims to approximately reconstruct a target vector on a domain/range that is spanned by the right or left singular vectors with large singular values. A least-squares type estimation of the target vector within this subspace of the data can be found by performing a singular vector transformation. This can provide a more flexible and powerful approach to dimensionality reduction than ordinary principal component analysis permits.

The problem of principal component regression can be formally stated as follows [FMMS16]: given a matrix $A \in \mathbb{R}^{n\times d}$, a vector $b \in \mathbb{R}^n$ and a threshold value $0 < \varsigma$, find $x \in \mathbb{R}^d$ such that

$$x = \operatorname{argmin}_{x\in\mathbb{R}^d}\left\|\widetilde{\Pi}_{\geq\varsigma} A \Pi_{\geq\varsigma} x - b\right\|, \tag{49}$$

where $\widetilde{\Pi}_{\geq\varsigma}, \Pi_{\geq\varsigma}$ denote left and right singular value threshold projectors. A closed-form expression for the optimal solution of (49) is given by $x = \Pi_{\geq\varsigma} A^+ \widetilde{\Pi}_{\geq\varsigma} b = A^+ \widetilde{\Pi}_{\geq\varsigma} b$.

As the following corollary shows, our singular value transformation techniques give rise to an efficient quantum algorithm for implementing $\Pi_{\geq\varsigma} A^+ \widetilde{\Pi}_{\geq\varsigma}$, and thus principal component regression.

Corollary 42 (Implementing the threshold pseudoinverse)**.** *Let $U, \Pi, \widetilde{\Pi}$ form a projected unitary encoding of the matrix A, and let $\varepsilon, \delta \in (0, \frac{1}{2}]$ and $0 < \varsigma < 1$. Suppose that $A = \widetilde{\Pi} U \Pi = W \Sigma V^\dagger$ is a singular value decomposition of the projected unitary encoding of A. Then there is an $m = \mathcal{O}\big(\frac{1}{\delta}\log(\frac{1}{\varepsilon})\big)$ and an efficiently computable $\Phi \in \mathbb{R}^m$ such that*[14]

$$\left\|\Big(\langle +| \otimes \big(\Pi - \Pi_{[\varsigma-\delta,\varsigma+\delta]}\big)\Big) U_\Phi \Big(|+\rangle \otimes \big(\widetilde{\Pi} - \widetilde{\Pi}_{[\varsigma-\delta,\varsigma+\delta]}\big)\Big) - \Pi_{\geq\varsigma}\Big(\frac{\varsigma}{2}A^+\Big)\widetilde{\Pi}_{\geq\varsigma}\right\| \leq \varepsilon. \tag{50}$$

Moreover, U_Φ can be implemented using a single ancilla qubit with m uses of U and $U^\dagger$, m uses of $C_\Pi NOT$ and m uses of $C_{\widetilde{\Pi}} NOT$ gates and m single qubit gates.

Proof. We can implement this operator by first applying a singular value threshold projector $\widetilde{\Pi}_{\geq\varsigma}$ according to Theorem 31, followed by performing the Moore-Penrose pseudoinverse A^+ as in Theorem 41.

Implementing these two operations separately is actually suboptimal. In order to get the stated result we simply take the polynomials used for singular value transformation in Theorem 31 and Theorem 41, then take their product and implement singular value transformation according to the product polynomial. The complexity statement can be proven similarly to the proofs of Theorem 31 and Theorem 41. □

Given a unitary preparing a quantum state $|b\rangle$ we can approximately solve principal component regression by applying an approximation of $\Pi_{\geq\varsigma}\big(\frac{\varsigma}{2}A^+\big)\widetilde{\Pi}_{\geq\varsigma}$ to $|b\rangle$, and then applying amplitude amplification to get $|x\rangle$. Strictly speaking, in order for this to work as required by (49), we would need to have the promise that $|b\rangle$ does not have an overlap with left singular vectors that have eigenvalues in $[\varsigma-\delta, \varsigma+\delta]$, while it does have a non-negligible overlap with left singular vectors having singular value $> \varsigma + \delta$. In fact, due to the nature of singular value transformation, for a left singular vector $|w_j\rangle$ with singular value $\varsigma_j \in [\varsigma-\delta, \varsigma+\delta]$ the procedure still performs a meaningful operation: it maps $|w_j\rangle \to f(\varsigma_j)|v_j\rangle$, such that $f(\varsigma_j) \in [-1, 1]$. It is plausible to believe that the transition behavior on $[\varsigma-\delta,\varsigma+\delta]$ would in practice not significantly degrade the performance of typical machine learning applications, therefore the promise of not having singular values in $[\varsigma-\delta,\varsigma+\delta]$ is probably not crucial. Also note that an essentially quadratic improvement to the runtime of the above procedure can be achieved using variable-time amplitude amplification techniques [Amb12, CKS17, CGJ18].

Finally, we briefly discuss a recently developed quantum machine learning algorithm which is significantly more complex then the previous algorithm, but can still be easily fitted to our framework. Kerenidis and Luongo recently proposed a quantum algorithm for slow feature analysis [KL18]. The main ingredient of their algorithm is to apply a threshold projection on some input state, i.e., to project onto the subspace spanned by the singular vectors of a matrix with singular values smaller than a certain threshold. Their algorithm is based on singular value estimation, whereas our Theorem 31 approaches the same problem in a more direct way, by transforming the singular values according to a threshold function.

In the first step of the quantum algorithm of Kerenidis and Luongo, the task is to implement $Y := V\Sigma^{-1}V^{\dagger}$ for a given input matrix $X = W\Sigma V^{\dagger}$. In our framework this can be performed analogously to Theorem 41 using singular value transformation; the only difference is that one needs to use an even polynomial approximation of $\frac{1}{x}$, for example given by Corollary 67. In the next step, one needs to implement singular value threshold projection using the matrix $\dot{X}Y$ for a given "derivative" matrix $\dot{X}$. Taking the product[21] of the two matrices can be implemented using Lemma 53, after which we can use our Theorem 31 to implement singular value threshold projection.

4 Matrix Arithmetics using blocks of unitaries

In this section we describe a generic toolbox for implementing matrix calculations on a quantum computer in an operational way, representing the vectors as quantum states. The matrix arithmetics methodology we propose carries out all calculations in an operational way, such that the matrices are represented by blocks of unitary operators of the quantum system, thereby can in principle result in exponential speed-ups in terms of the dimension of the matrices. The methodology we describe is a distilled version of the results of a series of works on quantum algorithms [HHL09, BCC$^+$15, CKS17, LC16, AGGW17, CGJ18].

We present the results in an intuitively structured way. First we define how to represent arbitrary matrices as blocks of unitaries, and show how to efficiently encode various matrices this way. Then we show how to implement addition and subtraction of these matrices, and finally show how to efficiently obtain products of block-encoded matrices. In order to make the results maximally reusable we also give bounds on the propagation of errors arising from inaccurate encodings.

4.1 Block-encoding

We introduce a definition of block-encoding which we are going to work with in the rest of the paper. The main idea is to represents a subnormalized matrix as the upper-left block of a unitary.

$$U = \begin{bmatrix} A/\alpha & . \\ . & . \end{bmatrix} \qquad \Longrightarrow \qquad A = \alpha(\langle 0| \otimes I)U(|0\rangle \otimes I)$$

[21] In case we would have a subnormalized version of $\dot{X}$, in order to get maximal efficiency, it usually worth amplifying $\dot{X}$ using Theorem 30 before taking the product $\dot{X}Y$.

Definition 43 (Block-encoding)**.** *Suppose that A is an s-qubit operator, $\alpha,\varepsilon\in\mathbb{R}_+$ and $a\in\mathbb{N}$, then we say that the $(s+a)$-qubit unitary U is an (α,a,ε)-block-encoding of A, if*

$$\left\|A-\alpha(\langle 0|^{\otimes a}\otimes I)U(|0\rangle^{\otimes a}\otimes I)\right\|\leq\varepsilon.$$

Note that since $\|U\|=1$ we necessarily have $\|A\|\leq\alpha+\varepsilon$. Also note that using the above definition it seems that we can only represent square matrices of size $2^s\times 2^s$. However, this is not really a restriction. Suppose that $A\in\mathbb{C}^{n\times m}$, where $n,m\leq 2^s$. Then we can define an embedding matrix denoted by $A_e\in\mathbb{C}^{2^s\times 2^s}$ such that the top-left block of A_e is A and all other elements are 0. This embedding is a faithful representation of the matrices. Suppose that $A,B\in\mathbb{C}^{n\times m}$ are matrices, then $A_e+B_e=(A+B)_e$. Moreover, suppose $C\in\mathbb{C}^{m\times k}$ for some $k\leq 2^s$, then $A_e\cdot C_e=(A\cdot C)_e$.

The above defined block-encoding is a special case of the projected-encoding of Definition 11, therefore we can later apply our singular value transformation results for block-encoded matrices. In this manner the advantage of block-encoding is that the $\mathrm{C}_\Pi\mathrm{NOT}$ gate which is required in order to implement the gates of Figure 1 is just a Toffoli gate on $a+1$ qubits, which can be implemented by $\mathcal{O}(a+1)$ two-qubit gates and using a single additional ancilla qubit [HLZ+17].

4.2 Constructing block-encodings

Definition 44 (Trivial block-encoding)**.** *A unitary matrix is a $(1,0,0)$-block-encoding of itself.*

If we ε-approximately implement a unitary U using a ancilla qubits via a unitary $\tilde{U}$ acting jointly on the system and the ancilla qubits, then $\tilde{U}$ is an $(1,a,\varepsilon)$-block-encoding of U. This is also a rather trivial encoding. Note that we make a slight distinction between ancilla qubits that are exactly returned to their original state after the computation and the ones that might pick up some error. The latter qubits we will treat as part of the encoding, and the former qubits we usually treat separately as purely ancillary qubits.

Now we present some non-trivial ways for constructing block-encodings, which will serve as a toolbox for efficiently inputting and representing matrices for arithmetic computations on a quantum computer. We will denote by I_w a w-qubit identity operator, and let SWAP_w denote the swap operation of two w-qubit register. We denote by CNOT the controlled not gate that targets the first qubit. When clear from the context we use simply notation $|0\rangle$ to denote $|0\rangle^{\otimes w}$.

First we show following Low and Chuang [LC16], how to create a block-encoding of a purified density operator. This technique can be used in combination with the optimal block-Hamiltonian simulation result Theorem 58, in order to get much better simulation performance, compared to density matrix exponentiation techniques [LMR14, KLL+17] which does not use purification. This result can be generalized for subnormalized density operators too, for more details see [AG18].

Lemma 45 (Block-encoding of density operators)**.** *Suppose that ρ is an s-qubit density operator and G is an $(a+s)$-qubit unitary that on the $|0\rangle|0\rangle$ input state prepares a purification $|0\rangle|0\rangle\to|\rho\rangle$, s.t. $\mathrm{Tr}_a|\rho\rangle\langle\rho|=\rho$. Then $(G^\dagger\otimes I_s)(I_a\otimes\mathrm{SWAP}_s)(G\otimes I_s)$ is a $(1,a+s,0)$-block-encoding of ρ.*

Proof. Let r be the Schmidt-rank of ρ, let $\{|\psi_k\rangle\colon k\in[2^s]\}$ be an orthonormal basis, let $\{|\phi_k\rangle\colon k\in[r]\}$ be an orthonormal system and let $p\in[0,1]^{2^s}$ be such that $|\rho\rangle=\sum_{k=1}^{r}\sqrt{p_k}|\phi_k\rangle|\psi_k\rangle$ and $p_\ell=0$

for all $\ell \in [2^s] \setminus [r]$. Then for all $i, j \in [2^s]$ we have that

$$\begin{aligned}
\langle 0|^{\otimes a+s}\langle\psi_i|(G^\dagger \otimes I_s)(I_a \otimes \mathrm{SWAP}_s)(G^\dagger \otimes I_s)|0\rangle^{\otimes a+s}|\psi_j\rangle = \\
&= \langle\rho|\langle\psi_i|(I_a \otimes \mathrm{SWAP}_s)|\rho\rangle|\psi_j\rangle \\
&= \left(\sum_{k=1}^{r}\sqrt{p_k}\langle\phi_k|\langle\psi_k|\right)\langle\psi_i|(I_a \otimes \mathrm{SWAP}_s)\left(\sum_{\ell=1}^{r}\sqrt{p_\ell}|\phi_\ell\rangle|\psi_\ell\rangle\right)|\psi_j\rangle \\
&= \left(\sum_{k=1}^{r}\sqrt{p_k}\langle\phi_k|\langle\psi_k|\langle\psi_i|\right)\left(\sum_{\ell=1}^{r}\sqrt{p_\ell}|\phi_\ell\rangle|\psi_j\rangle|\psi_\ell\rangle\right) \\
&= \sqrt{p_j p_i}\delta_{ij} \\
&= \langle\psi_i|\rho|\psi_j\rangle.
\end{aligned}$$

□

Apeldoorn and Gilyén [AG18] recently also showed that an implementation scheme for a POVM operator can also be easily transformed to block-encoding of the POVM operators. By an implementation scheme we mean a quantum circuit U that given input ρ and a ancilla qubits, it sets a flag qubit to 0 with probability $\mathrm{Tr}[\rho M]$.

Lemma 46 (Block-encoding of POVM operators)**.** *Suppose that U is an $a+s$ qubit unitary, which implements a POVM operator M with ε-precision such that for all s-qubit density operator ρ*

$$\left|\mathrm{Tr}[\rho M] - \mathrm{Tr}\Big[U\big(|0\rangle\langle 0|^{\otimes a} \otimes \rho\big)U^\dagger\big(|0\rangle\langle 0|^{\otimes 1} \otimes I_{a+s-1}\big)\Big]\right| \leq \varepsilon. \tag{51}$$

Then $(I_1 \otimes U^\dagger)(\mathrm{CNOT} \otimes I_{a+s-1})(I_1 \otimes U)$ is a $(1, 1+a, \varepsilon)$-block-encoding of the matrix M.

Proof. First observe that by the cyclicity of trace we have that

$$\begin{aligned}
\mathrm{Tr}\Big[U\big(|0\rangle\langle 0|^{\otimes a} \otimes \rho\big)U^\dagger(|0\rangle\langle 0| \otimes I_{a+s-1})\Big] &= \mathrm{Tr}\Big[U\big(|0\rangle^{\otimes a} \otimes I\big)\rho\big(\langle 0|^{\otimes a} \otimes I\big)U^\dagger(|0\rangle\langle 0| \otimes I_{a+s-1})\Big] \\
&= \mathrm{Tr}\Big[\rho\big(\langle 0|^{\otimes a} \otimes I\big)U^\dagger(|0\rangle\langle 0| \otimes I_{a+s-1})U\big(|0\rangle^{\otimes a} \otimes I\big)\Big].
\end{aligned}$$

Together with (51) this implies that for all ρ density operator

$$\left|\mathrm{Tr}\Big[\rho\Big(M - \big(\langle 0|^{\otimes a} \otimes I\big)U^\dagger(|0\rangle\langle 0| \otimes I_{a+s-1})U\big(|0\rangle^{\otimes a} \otimes I\big)\Big)\Big]\right| \leq \varepsilon,$$

which is equivalent to saying that $\left\|M - \big(\langle 0|^{\otimes a} \otimes I\big)U^\dagger(|0\rangle\langle 0| \otimes I_{a+s-1})U\big(|0\rangle^{\otimes a} \otimes I\big)\right\| \leq \varepsilon$. We can conclude by observing that

$$\begin{aligned}
\big(\langle 0|^{\otimes a} \otimes I\big)U^\dagger(|0\rangle\langle 0| \otimes I_{a+s-1})U\big(|0\rangle^{\otimes a} \otimes I\big) = \\
&= \Big(\langle 0|^{\otimes 1+a} \otimes I\Big)\Big(I_1 \otimes U^\dagger\Big)(\mathrm{CNOT} \otimes I_{a+s-1})\Big(I_1 \otimes U\Big)\Big(|0\rangle^{\otimes 1+a} \otimes I\Big).
\end{aligned}$$

□

Now we turn to a more traditional way of constructing block-encodings via state preparation. This is a common technique for example to implement quantum walks. Note that we introduce the notation $[n]-1$ to denote the set $\{0, 1, \ldots, n-1\}$.

Lemma 47 (Block-encoding of Gram matrices by state preparation unitaries)**.** *Let U_L and U_R be "state preparation" unitaries acting on $a+s$ qubits preparing the vectors $\{|\psi_i\rangle\colon i\in[2^s]-1\}$, $\{|\phi_j\rangle\colon j\in[2^s]-1\}$, s.t.*

$$U_L\colon |0\rangle|i\rangle \to |\psi_i\rangle$$
$$U_R\colon |0\rangle|j\rangle \to |\phi_j\rangle.$$

Then $U=U_L^\dagger U_R$ is an $(1,a,0)$-block-encoding of the Gram matrix A such that $A_{ij}=\langle\psi_i|\phi_j\rangle$.

Based on the above idea one can efficiently implement block-encodings of sparse-access matrices.

Lemma 48 (Block-encoding of sparse-access matrices)**.** *Let $A\in\mathbb{C}^{2^w\times 2^w}$ be a matrix that is s_r-row-sparse and s_c-column-sparse, and each element of A has absolute value at most 1. Suppose that we have access to the following sparse-access oracles acting on two $(w+1)$ qubit registers*

$$\begin{aligned}\mathrm{O}_r\colon |i\rangle|k\rangle &\to |i\rangle|r_{ik}\rangle && \forall i\in[2^w]-1, k\in[s_r], \text{ and}\\ \mathrm{O}_c\colon |\ell\rangle|j\rangle &\to |c_{\ell j}\rangle|j\rangle && \forall \ell\in[s_c], j\in[2^w]-1, \text{ where}\end{aligned}$$

r_{ij} is the index for the j-th non-zero entry of the i-th row of A, or if there are less than i non-zero entries, then it is $j+2^w$, and similarly c_{ij} is the index for the i-th non-zero entry of the j-th column of A, or if there are less than j non-zero entries, then it is $i+2^w$. Additionally assume that we have access to an oracle O_A that returns the entries of A in a binary description

$$\mathrm{O}_A\colon |i\rangle|j\rangle|0\rangle^{\otimes b} \to |i\rangle|j\rangle|a_{ij}\rangle \qquad \forall i,j\in[2^w]-1, \text{ where}$$

a_{ij} is a b-bit binary description[22] of the ij-matrix element of A. Then we can implement a $(\sqrt{s_r s_c}, w+3, \varepsilon)$-block-encoding of A with a single use of $\mathrm{O}_r, \mathrm{O}_c$, two uses of O_A and additionally using $\mathcal{O}\big(w+\log^{2.5}(\frac{s_r s_c}{\varepsilon})\big)$ one and two qubit gates while using $\mathcal{O}\big(b, \log^{2.5}(\frac{s_r s_c}{\varepsilon})\big)$ ancilla qubits.

Proof. We proceed by constructing state preparation unitaries in the spirit of Lemma 47. We will work with 3-registers the first of which is a single qubit register, and the other two registers have $(w+1)$ qubits. Let D_s be a $(w+1)$-qubit unitary that implements the map $|0\rangle\to\sum_{k=1}^{s}\frac{|k\rangle}{\sqrt{s}}$, it is known that this operator D_s can be implemented with $\mathcal{O}(w)$ quantum gates using $\mathcal{O}(1)$ ancilla qubits. Then we define the $2(w+1)$ qubit unitary $V_L:=\mathrm{O}_r(I_{w+2}\otimes D_{s_r})\mathrm{SWAP}_{w+1}$ such that

$$V_L\colon |0\rangle^{w+2}|i\rangle \to \sum_{k=1}^{s_r}\frac{|i\rangle|r_{ik}\rangle}{\sqrt{s_r}} \qquad \forall i\in[2^w]-1.$$

We implement the operator $V_R:=\mathrm{O}_c(D_{s_c}\otimes I_{w+1})$ in a similar way acting as

$$V_R\colon |0\rangle^{w+2}|j\rangle \to \sum_{\ell=1}^{s_c}\frac{|c_{\ell j}\rangle|j\rangle}{\sqrt{s_c}} \qquad \forall j\in[2^w]-1.$$

It is easy to see that the above unitaries are such that

$$\langle 0|^{w+2}\langle i|V_L^\dagger V_R|0\rangle^{w+2}|j\rangle = \frac{1}{\sqrt{s_r s_c}} \text{ if } a_{ij}\neq 0 \text{ and } 0 \text{ otherwise.}$$

[22] For simplicity we assume here that the binary representation is exact.

Now we define $U_L := I_1 \otimes V_L$ and define U_R as performing the unitary $I_1 \otimes V_R$ followed by some extra computation. After performing V_R we get a superposition of index pairs $|i\rangle|j\rangle$. Given an index pair $|i\rangle|j\rangle$ we query the matrix element $|a_{ij}\rangle$ using the oracle O_A. Then we do some elementary computations in order to implement a single qubit gate $|0\rangle \to a_{ij}|0\rangle + \sqrt{1-|a_{ij}|^2}|1\rangle$ on the first qubit, with precision $\mathcal{O}\Big(\mathrm{poly}\Big(\frac{\varepsilon}{s_r s_c}\Big)\Big)$. This can be executed with the stated complexity, for more details see, e.g., the work of Berry et al. [BCK15]. Finally we also need to uncompute everything which requires one more use of O_A. This way we get a good approximation of

$$U_R\colon |0\rangle^{w+3}|j\rangle \to \sum_{\ell=1}^{s_c} \frac{\Big(a_{c_{\ell j}j}|0\rangle + \sqrt{1-|a_{c_{\ell j}j}|^2}|1\rangle\Big)|c_{\ell j}\rangle|j\rangle}{\sqrt{s_c}} \qquad \forall j \in [2^w]-1.$$

□

Note that in the above method the matrix gets subnormalized by a factor of $\frac{1}{\sqrt{s_r s_c}}$. If we would know that for example $\|A\| \leq \frac{1}{2}$, then we could amplify the block-encoding in order to remove this unwanted subnormalization using singular value amplification Theorem 30 using the block-encoding roughly $\sqrt{s_r s_c}$ times. However, under some circumstances one can defeat the subnormalization more efficiently by doing an amplification at the level of the state preparation unitaries. The idea comes from Low and Chuang [LC17a], who called this technique "Uniform spectral gap amplification". We generalize their results combining with ideas of Kerenidis and Prakash [KP17a] and Chakraborty et al. [CGJ18], who used similar ideas but assumed QROM-access to matrices rather than sparse-access.

Lemma 49 (Preamplified block-encoding of sparse-access matrices)**.** *Let $A \in \mathbb{C}^{2^w \times 2^w}$ be a matrix that is s_r-row-sparse and s_c-column-sparse, and is given using the input oracles defined in Lemma 48. Let $a_{i.}$ denote the i-th row of A and similarly $a_{.j}$ the j-th column. Let $q \in [0,2]$ and suppose that $n_r \in [1, s_r]$ is an upper bound on $\|a_{i.}\|_q^q$ and $n_c \in [1, s_c]$ is an upper bound on $\|a_{.j}\|_{2-q}^{2-q}$.*

Let $m = \max[\frac{s_r}{n_r}, \frac{s_c}{n_c}]$. Then we can implement a $\big(\sqrt{\frac{1}{2n_r n_c}}, w+6, \varepsilon\big)$-block-encoding of A with $\mathcal{O}\Big(\sqrt{\frac{s_r}{n_r}}\log(\frac{s_r s_c}{\varepsilon})\Big)$ uses of O_r, $\mathcal{O}\Big(\sqrt{\frac{s_c}{n_c}}\log(\frac{s_r s_c}{\varepsilon})\Big)$ uses of O_c, $\mathcal{O}\big(\sqrt{m}\log(\frac{s_r s_c}{\varepsilon})\big)$ uses of O_A, and additionally using $\mathcal{O}\big(\sqrt{m}\big(w\log(\frac{s_r s_c}{\varepsilon}) + \log^{3.5}(\frac{s_r s_c}{\varepsilon})\big)\big)$ one and two qubit gates while using $\mathcal{O}\big(b, \log^{2.5}(\frac{s_r s_c}{\varepsilon})\big)$ ancilla qubits.

Proof. The idea is very similar to the proof of Lemma 48, we implement the unitaries V_L, V_R the same way. However, we define U_R, U_L slightly differently. Using a similar method than in Lemma 48, we implement $\mathcal{O}\Big(\mathrm{poly}\Big(\frac{\varepsilon}{s_r s_c}\Big)\Big)$-approximations of the maps

$$U_L\colon |0\rangle^{w+4}|i\rangle \to \sum_{k=1}^{s_r} \frac{\Big(|a_{ir_{ik}}|^{\frac{q}{2}}|0\rangle + \sqrt{1-|a_{ir_{ik}}|^q}|1\rangle\Big)|0\rangle|i\rangle|r_{ik}\rangle}{\sqrt{s_r}} \qquad \forall i \in [2^w]-1,$$

$$U_R\colon |0\rangle^{w+4}|j\rangle \to \sum_{\ell=1}^{s_c} \frac{\frac{a_{c_{\ell j}j}}{|a_{c_{\ell j}j}|}|0\rangle\Big(|a_{c_{\ell j}j}|^{1-\frac{q}{2}}|0\rangle + \sqrt{1-|a_{c_{\ell j}j}|^{2-q}}|1\rangle\Big)|c_{\ell j}\rangle|j\rangle}{\sqrt{s_c}} \qquad \forall j \in [2^w]-1.$$

It is easy to see that the above unitaries are such that

$$\langle 0|^{w+4}\langle i|U_L^\dagger U_R|0\rangle^{w+4}|j\rangle = \frac{a_{ij}}{\sqrt{s_r s_c}} \qquad \forall i,j \in [2^w]-1.$$

We can see that for all $i \in [2^w] - 1$ the modified row vector $\sum_{k=1}^{s_r} \frac{|a_{ir_{ik}}|^{\frac{q}{2}}|0\rangle|0\rangle|i\rangle|r_{ik}\rangle}{\sqrt{s_r}}$ has squared norm at most $\frac{n_r}{s_r}$, and a similar $\frac{n_c}{s_c}$ upper bound holds for the squared norm of the modified column vector. Also observe that

$$(|0\rangle\langle 0| \otimes I_{2w+3})U_L(|0\rangle\langle 0|^{w+4} \otimes I_w) = \sum_{j=0}^{2^w-1}\left(\sum_{k=1}^{s_r} \frac{|a_{ir_{ik}}|^{\frac{q}{2}}|0\rangle|0\rangle|i\rangle|r_{ik}\rangle}{\sqrt{s_r}}\right)\langle 0|^{w+4}\langle j|,$$

which is a singular value decomposition with the singular values being the modified row norms. Therefore we can apply singular value amplification Theorem 30 to with amplification $\gamma_r = \sqrt{\frac{s_r}{\sqrt{2}n_r}}$ and precision $\mathcal{O}\Big(\mathrm{poly}\Big(\frac{\varepsilon}{s_r s_c}\Big)\Big)$ resulting in an $\mathcal{O}\Big(\mathrm{poly}\Big(\frac{\varepsilon}{s_r s_c}\Big)\Big)$ approximation of $\tilde{U}_L$ such that

$$(\langle +| \otimes |0\rangle\langle 0| \otimes I_{2w+3})\tilde{U}_L(|+\rangle \otimes |0\rangle\langle 0|^{w+4} \otimes I_w) = \gamma_r \sum_{j=0}^{2^w-1}\left(\sum_{k=1}^{s_r} \frac{|a_{ir_{ik}}|^{\frac{q}{2}}|0\rangle|0\rangle|i\rangle|r_{ik}\rangle}{\sqrt{s_r}}\right)\langle 0|^{w+4}\langle j|.$$

Similarly we apply singular value amplification with amplification $\gamma_c = \sqrt{\frac{s_c}{\sqrt{2}n_c}}$ and precision $\mathcal{O}\Big(\mathrm{poly}\Big(\frac{\varepsilon}{s_r s_c}\Big)\Big)$ resulting in a $\mathcal{O}\Big(\mathrm{poly}\Big(\frac{\varepsilon}{s_r s_c}\Big)\Big)$ approximation of $\tilde{U}_R$ such that

$$\langle ++|\langle 0|^{w+4}\langle i|\tilde{U}_L^\dagger \tilde{U}_R|++\rangle|0\rangle^{w+4}|j\rangle = \gamma_r\gamma_c \frac{a_{ij}}{\sqrt{s_r s_c}} = \frac{a_{ij}}{\sqrt{2n_r n_c}} \qquad \forall i,j \in [2^w] - 1.$$

Finally adding 4 Hadamard gates we can change the $|+\rangle$ states above to $|0\rangle$ states, resulting in the $\big(\sqrt{\frac{1}{2n_r n_c}}, w+6, \varepsilon\big)$-block-encoding of A. The complexity statement follows similarly as in the proof of Lemma 48, with the extra observation that the singular value amplifications of U_L and U_R can be performed using degree $\mathcal{O}\big(\gamma_r \log(\frac{s_r s_c}{\varepsilon})\big)$ and $\mathcal{O}\big(\gamma_c \log(\frac{s_r s_c}{\varepsilon})\big)$ singular value transformations respectively. □

Finally, for completeness we invoke the results of Kerenidis and Prakash [KP17a] and Chakraborty et al. [CGJ18], who showed how to efficiently implement block-encodings of matrices that are stored in a clever quantum data structures in a quantum accessible RAM.

For $q \in [0,2]$ let us define $\mu_q(A) = \sqrt{n_q(A)n_{(2-q)}(A^T)}$, where $n_q(A) := \max_i \|a_{i.}\|_q^q$ is the q-th power of the maximum q-norm of the rows of A. Let $A^{(q)}$ denote the matrix of the same dimensions as A, with[23] $A_{ij}^{(q)} = \sqrt{a_{ij}^q}$. The following was proven in [KP17a], although not in the language of block-encodings, and was stated in this form by Chakraborty et al. [CGJ18].

Lemma 50 (Block-encodings of matrices stored in quantum data structures). *Let $A \in \mathbb{C}^{2^w \times 2^w}$.*

1. *Fix $q \in [0,2]$. If $A^{(q)}$ and $(A^{(2-q)})^\dagger$ are both stored in quantum accessible data structures[24], then there exist unitaries U_R and U_L that can be implemented in time $\mathcal{O}(\mathrm{poly}(w\log(1/\varepsilon)))$ such that $U_R^\dagger U_L$ is a $(\mu_q(A), w+2, \varepsilon)$-block-encoding of A.*

2. *On the other hand, if A is stored in a quantum accessible data structure[24], then there exist unitaries U_R and U_L that can be implemented in time $\mathcal{O}(\mathrm{poly}(w\log(1/\varepsilon)))$ such that $U_R^\dagger U_L$ is an $(\|A\|_F, w+2, \varepsilon)$-block-encoding of A.*

[23] For complex values we define these non-integer powers using the principal value of the complex logarithm function.

[24] Here we assume that the data-structure stores the matrices with sufficient precision, cf. [CGJ18].

4.3 Addition and subtraction: Linear combination of block-encoded matrices

We use a simple but powerful method for implementing linear combinations of unitary operators on a quantum computer. This technique was introduced by Berry et al. [BCC+15] for exponentially improving the precision of Hamiltonian simulation. Later it was adapted by Childs et al. [CKS17] for exponentially improving the precision of quantum linear equation solving. Here we present this method from the perspective of block-encoded matrices.

First we define state preparation unitaries in order to conveniently state our the result in the following lemma.

Definition 51 (State preparation pair)**.** *Let $y \in \mathbb{C}^m$ and $\|y\|_1 \leq \beta$, the pair of unitaries (P_L, P_R) is called a (β, b, ε)-state-preparation-pair if $P_L|0\rangle^{\otimes b} = \sum_{j=0}^{2^b-1} c_j|j\rangle$ and $P_R|0\rangle^{\otimes b} = \sum_{j=1}^{2^b-1} d_j|j\rangle$ such that $\sum_{j=0}^{m-1} |\beta(c_j^* d_j) - y_j| \leq \varepsilon$ and for all $j \in m, \ldots, 2^b - 1$ we have $c_j^* d_j = 0$.*

Now we show how to implement a block-encoding of a linear combination of block-encoded operators.

Lemma 52 (Linear combination of block-encoded matrices)**.** *Let $A = \sum_{j=1}^{m} y_j A_j$ be an s-qubit operator and $\varepsilon \in \mathbb{R}_+$. Suppose that (P_L, P_R) is a $(\beta, b, \varepsilon_1)$-state-preparation-pair for y, $W = \sum_{j=0}^{m-1} |j\rangle\langle j| \otimes U_j + ((I - \sum_{j=0}^{m-1} |j\rangle\langle j|) \otimes I_a \otimes I_s)$ is an $s + a + b$ qubit unitary such that for all $j \in 0, \ldots, m$ we have that U_j is an $(\alpha, a, \varepsilon_2)$-block-encoding of A_j. Then we can implement a $(\alpha\beta, a + b, \alpha\varepsilon_1 + \alpha\beta\varepsilon_2)$-block-encoding of A, with a single use of W, P_R and $P_L^\dagger$.*

Proof. Observe that $\widetilde{W} = (P_L^\dagger \otimes I_a \otimes I_s) W (P_R \otimes I_a \otimes I_s)$ is a $(\alpha\beta, a+b, \alpha\varepsilon_1 + \alpha\beta\varepsilon_2)$-block-encoding of A:

$$\begin{aligned}
\left\| A - \alpha\beta(\langle 0|^{\otimes b} \otimes \langle 0|^{\otimes a} \otimes I)\widetilde{W}(|0\rangle^{\otimes b} \otimes |0\rangle^{\otimes a} \otimes I) \right\| &= \left\| A - \alpha \sum_{j=0}^{m-1} \beta(c_j^* d_j)((\langle 0|^{\otimes a} \otimes I)U_j(|0\rangle^{\otimes a} \otimes I) \right\| \\
&\leq \alpha\varepsilon_1 + \left\| A - \alpha \sum_{j=0}^{m-1} y_j((\langle 0|^{\otimes a} \otimes I)U_j(|0\rangle^{\otimes a} \otimes I) \right\| \\
&\leq \alpha\varepsilon_1 + \alpha \sum_{j=0}^{m-1} y_j \left\| A_j - ((\langle 0|^{\otimes a} \otimes I)U_j(|0\rangle^{\otimes a} \otimes I) \right\| \\
&\leq \alpha\varepsilon_1 + \alpha \sum_{j=0}^{m-1} y_j \varepsilon_2 \\
&\leq \alpha\varepsilon_1 + \alpha\beta\varepsilon_2.
\end{aligned}$$

□

4.4 Multiplication: Product of block-encoded matrices

In general if we want to take the product of two block encoded matrices we need to treat their ancilla qubits separately. In this case as the following lemma shows the errors simply add up and the block encoding does not introduce any additional errors.

Lemma 53 (Product of block-encoded matrices)**.** *If U is an (α,a,δ)-block-encoding of an s-qubit operator A, and V is an (β,b,ε)-block-encoding of an s-qubit operator B then*[25] $(I_b\otimes U)(I_a\otimes V)$ *is an $(\alpha\beta,a+b,\alpha\varepsilon+\beta\delta)$-block-encoding of AB.*

Proof.

$$\begin{aligned}
&\left\|AB-\alpha\beta(\langle 0|^{\otimes a+b}\otimes I)(I_b\otimes U)(I_a\otimes V)(|0\rangle^{\otimes a+b}\otimes I)\right\|\\
=&\left\|AB-\underbrace{\alpha(\langle 0|^{\otimes a}\otimes I)U(|0\rangle^{\otimes a}\otimes I)}_{\tilde{A}}\underbrace{\beta(\langle 0|^{\otimes b}\otimes I)V(|0\rangle^{\otimes b}\otimes I)}_{\tilde{B}}\right\|\\
=&\left\|AB-\tilde{A}B+\tilde{A}B-\tilde{A}\tilde{B}\right\|\\
=&\left\|(A-\tilde{A})B+\tilde{A}(B-\tilde{B})\right\|\\
\leq&\left\|A-\tilde{A}\right\|\beta+\alpha\left\|B-\tilde{B}\right\|\\
\leq&\alpha\varepsilon+\beta\delta.
\end{aligned}$$

□

In the special case when the encoded matrices are unitaries and their block-encoding does not use any extra scaling factor, then we might reuse the ancilla qubits, however it introduces an extra error term, which can be bounded by the geometrical mean of the two input error bounds.

Lemma 54 (Product of two block-encoded unitaries)**.** *If U is an $(1,a,\delta)$-block-encoding of an s-qubit unitary operator A, and V is an $(1,a,\varepsilon)$-block-encoding of an s-qubit unitary operator B then UV is a $(1,a,\delta+\varepsilon+2\sqrt{\delta\varepsilon})$-block-encoding of the unitary operator AB.*

Proof. It is enough to show that for all s-qubit pure states $|\phi\rangle,|\psi\rangle$ we have that

$$\left|\langle\phi|AB|\psi\rangle-\langle\phi|(\langle 0|^{\otimes a}\otimes I)UV(|0\rangle^{\otimes a}\otimes I)|\psi\rangle\right|\leq\delta+\varepsilon+2\sqrt{\delta\varepsilon}.$$

Observe that

$$\begin{aligned}
&\langle\phi|(\langle 0|^{\otimes a}\otimes I)UV(|0\rangle^{\otimes a}\otimes I)|\psi\rangle\\
=&\langle\phi|(\langle 0|^{\otimes a}\otimes I)U\big((|0\rangle\!\langle 0|^{\otimes a}\otimes I)+\big((I-|0\rangle\!\langle 0|^{\otimes a})\otimes I\big)\big)V(|0\rangle^{\otimes a}\otimes I)|\psi\rangle\\
=&\langle\phi|(\langle 0|^{\otimes a}\otimes I)U(|0\rangle^{\otimes a}\otimes I)(\langle 0|^{\otimes a}\otimes I)V(|0\rangle^{\otimes a}\otimes I)|\psi\rangle\\
&+\langle\phi|(\langle 0|^{\otimes a}\otimes I)U\big((I-|0\rangle\!\langle 0|^{\otimes a})\otimes I\big)V(|0\rangle^{\otimes a}\otimes I)|\psi\rangle
\end{aligned}$$

Now we can see that similarly to the proof of Lemma 53 we have

$$\begin{aligned}
&\left|\langle\phi|AB|\psi\rangle-\langle\phi|(\langle 0|^{\otimes a}\otimes I)U(|0\rangle^{\otimes a}\otimes I)(\langle 0|^{\otimes a}\otimes I)V(|0\rangle^{\otimes a}\otimes I)|\psi\rangle\right|\\
=&\left|\langle\phi|\big(AB-(\langle 0|^{\otimes a}\otimes I)U(|0\rangle^{\otimes a}\otimes I)(\langle 0|^{\otimes a}\otimes I)V(|0\rangle^{\otimes a}\otimes I)\big)|\psi\rangle\right|\\
\leq&\left\|AB-(\langle 0|^{\otimes a}\otimes I)U(|0\rangle^{\otimes a}\otimes I)(\langle 0|^{\otimes a}\otimes I)V(|0\rangle^{\otimes a}\otimes I)\right\|\\
\leq&\delta+\varepsilon.
\end{aligned}$$

[25] The identity operators act on each others ancilla qubits, which is hard to express properly using simple tensor notation, but the reader should read this tensor product this way.

Finally note that

$$\begin{aligned}
&\left|\langle\phi|(\langle 0|^{\otimes a}\otimes I)U\big((I-|0\rangle\langle 0|^{\otimes a})\otimes I\big)V(|0\rangle^{\otimes a}\otimes I)|\psi\rangle\right|\\
=&\left|\langle\phi|(\langle 0|^{\otimes a}\otimes I)U\big((I-|0\rangle\langle 0|^{\otimes a})\otimes I\big)^2V(|0\rangle^{\otimes a}\otimes I)|\psi\rangle\right|\\
\leq&\left\|\big((I-|0\rangle\langle 0|^{\otimes a})\otimes I\big)U(|0\rangle^{\otimes a}\otimes I)|\phi\rangle\right\|\cdot\left\|\big((I-|0\rangle\langle 0|^{\otimes a})\otimes I\big)V(|0\rangle^{\otimes a}\otimes I)|\psi\rangle\right\|\\
=&\sqrt{1-\left\|(|0\rangle\langle 0|^{\otimes a}\otimes I)U(|0\rangle^{\otimes a}\otimes I)|\phi\rangle\right\|^2}\cdot\sqrt{1-\left\|(|0\rangle\langle 0|^{\otimes a}\otimes I)V(|0\rangle^{\otimes a}\otimes I)|\psi\rangle\right\|^2}\\
\leq&\sqrt{1-(1-\delta)^2}\cdot\sqrt{1-(1-\varepsilon)^2}\\
\leq&2\sqrt{\delta\varepsilon}.
\end{aligned}$$

□

The following corollary suggest that if we multiply together multiple block-encoded unitaries, the error may grow super-linearly, but it increases at most quadratically with the number of factors in the product.

Corollary 55 (Product of multiple block-encoded unitaries)**.** *Suppose that U_j is an $(1,a,\varepsilon)$-block-encoding of an s-qubit unitary operator W_j for all $j\in[K]$. Then $\prod_{j=1}^K U_j$ is an $(1,a,4K^2\varepsilon)$-block-encoding of $\prod_{j=1}^K W_j$.*

Proof. First observe that for the product of two matrices we get the precision bound 4ε by the above lemma. If $K=2^k$ for some $k\in\mathbb{N}$. Then we can apply the above observation in a recursive fashion in a binary tree structure, to get the upper bound $4^k\varepsilon$ on the precision, and observe that $4^k=K^2$.

If $2^{k-1}\leq K<2^k$ we can just add identity operators so that we have 2^k matrices to multiply, which gives the precision bound $4^k\varepsilon\leq 4^{1+\log_2 K}\varepsilon=4K^2\varepsilon$. □

5 Implementing smooth functions of Hermitian matrices

In the previous section we developed an efficient methodology to perform basic matrix arithmetics, such as addition and multiplication. In principle all smooth functions of matrices can be approximated arbitrarily precisely using such basic arithmetic operations. In this section we show how to more efficiently transform Hermitian matrices according to smooth functions using singular value transformation techniques. The key observation is that for a Hermitian matrix A we have that $P^{(SV)}(A)=P(A)$, i.e., singular value transformation and eigenvalue transformation coincide.

The following theorem is our improvement of Corollary 18 removing the counter-intuitive parity constraint at the expense of a subnormalization factor 1/2, which is not a problem in most applications.

Theorem 56 (Polynomial eigenvalue transformation of arbitrary parity)**.** *Suppose that U is an (α,a,ε)-encoding of a Hermitian matrix A. If $\delta\geq 0$ and $P_{\Re}\in\mathbb{R}[x]$ is a degree-d polynomial satisfying that*

- *for all $x\in[-1,1]$: $|P_{\Re}(x)|\leq\frac{1}{2}$.*

Then there is a quantum circuit $\tilde{U}$, which is an $(1,a+2,4d\sqrt{\varepsilon/\alpha}+\delta)$-encoding of $P_{\Re}(A/\alpha)$, and consists of d applications of U and $U^\dagger$ gates, a single application of controlled-U and $\mathcal{O}((a+1)d)$ other one- and two-qubit gates. Moreover we can compute a description of such a circuit with a classical computer in time $\mathcal{O}(\mathrm{poly}(d,\log(1/\delta)))$.

Proof. First note that for a Hermitian matrix A and for any even/odd polynomial $P \in \mathbb{C}[x]$ we have that $P^{(SV)}(A) = P(A)$. Now let $P_{\Re}^{(\mathrm{even})}(x) := P_{\Re}(x) + P_{\Re}(-x) \leq 1$, and let $P_{\Re}^{(\mathrm{odd})}(x) := P_{\Re}(x) - P_{\Re}(-x) \leq 1$. Then we can implements both $P_{\Re}^{(\mathrm{even})}(x)$ and $P_{\Re}^{(\mathrm{odd})}(x)$ using Corollary 18 and we can take an equal $\frac{1}{2}$ linear combination of them by Lemma 22. Using the notation of Figure 1 the final circuit is simply $(H \otimes H \otimes I)U_{\Phi(c)}(H \otimes H \otimes I)$, where H denotes the Hadamard gate. □

Note that a similar statement can be proven for arbitrary $P \in \mathbb{C}[x]$ that satisfy $|P(x)| \leq \frac{1}{4}$ for all $x \in [-1,1]$. The only difference is that for implementing the complex part one needs to add a (controlled) phase $e^{i\frac{\pi}{2}}$. The $\leq \frac{1}{4}$ constraint comes form the fact that the implementation is a sum of 4 different terms (even/odd component of the real/imaginary part).

5.1 Optimal Hamiltonian simulation

Let us define for $t \in \mathbb{R}_+$ and $\varepsilon \in (0,1)$ the number $r(t,\varepsilon) \geq t$ as the solution to the equation

$$\varepsilon = \left(\frac{t}{r}\right)^r : r \in (t, \infty). \tag{52}$$

This equation is closely related to the Lambert-W function, and unfortunately we cannot give the solution in terms of elementary functions. However, one can see that the function $\left(\frac{t}{r}\right)^r$ is strictly monotone decreasing for $r \in [t,\infty)$ and in the limit $r \to \infty$ it tends to 0. Since for $r = t$ the function value is 1, therefore the equation (52) has a unique solution. In particular for any $r, R \in [t,\infty)$ such that $\left(\frac{t}{r}\right)^r \geq \varepsilon \geq \left(\frac{t}{R}\right)^R$ we have that $r \leq r(t,\varepsilon) \leq R$. This is an important expression for this section, since Low and Chuang proved [LC17b, LC16] that the complexity of Hamiltonian simulation for time t with precision ε is $\Theta(r(|t|,\varepsilon))$.

Low and Chuang also claimed [LC17b] that for all $t \geq 1$ one gets $r(t,\varepsilon) = \Theta\Big(t + \frac{\log(1/\varepsilon)}{\log(\log(1/\varepsilon))}\Big)$, which led to their complexity statement. Note that we found a subtle issue in their calculations making this formula invalid for some range of values of t. We show in Lemma 59 how to correct the formula by a slight modification of the $\log(\log(1/\varepsilon))$ term. This is the reason why we need to give more complicated expressions for the complexity of block-Hamiltonian simulation. First we show what is the connection between equation (52) and the complexity of Hamiltonian simulation by constructing polynomials of degree $\mathcal{O}(r(|t|,\varepsilon))$, which ε-approximate trigonometric functions with t-times rescaled argument.

Lemma 57 (Polynomial approximations of trigonometric functions)**.** *Let $t \in \mathbb{R} \setminus \{0\}$, $\varepsilon \in (0, \frac{1}{e})$, and let $R = \left\lfloor r\Big(\frac{e|t|}{2}, \frac{5}{4}\varepsilon\Big)/2 \right\rfloor$, then the following $2R$ and $2R+1$ degree polynomials satisfy*

$$\left\| \cos(tx) - J_0(t) + 2\sum_{k=1}^{R}(-1)^k J_{2k}(t) T_{2k}(x) \right\|_{[-1,1]} \leq \varepsilon, \text{ and}$$

$$\left\| \sin(tx) - 2\sum_{k=0}^{R}(-1)^k J_{2k+1}(t) T_{2k+1}(x) \right\|_{[-1,1]} \leq \varepsilon,$$

where $J_m(t)\colon m \in \mathbb{N}$ denote Bessel functions of the first kind.

Proof. We use the Fourier-Chebyshev series of the trigonometric functions given by the Jacobi-Anger expansion [AS74, 9.1.44-45]:

$$\begin{aligned}\cos(tx) &= J_0(t) + 2\sum_{k=1}^{\infty}(-1)^k J_{2k}(t)T_{2k}(x)\\ \sin(tx) &= 2\sum_{k=0}^{\infty}(-1)^k J_{2k+1}(t)T_{2k+1}(x).\end{aligned}$$

The Jacobi-Anger expansion implies that

$$\begin{aligned}\left\|\cos(tx) - J_0(t) + 2\sum_{k=1}^{R}(-1)^k J_{2k}(t)T_{2k}(x)\right\|_{[-1,1]} &= \left\|2\sum_{k=R+1}^{\infty}(-1)^k J_{2k}(t)T_{2k}(x)\right\|_{[-1,1]}\\ &\leq 2\sum_{k=R+1}^{\infty}(-1)^k |J_{2k}|\\ &= 2\sum_{\ell=0}^{\infty}(-1)^k |J_{2R+2+2\ell}|, \qquad (53)\end{aligned}$$

and similarly we can derive that

$$\left\|\sin(tx) - 2\sum_{k=0}^{R}(-1)^k J_{2k+1}(t)T_{2k+1}(x)\right\|_{[-1,1]} \leq 2\sum_{\ell=0}^{\infty}(-1)^k |J_{2R+3+2\ell}|. \qquad (54)$$

It is known [AS74, 9.1.62] that for all $m \in \mathbb{N}_+$ and $t \in \mathbb{R}$ we have

$$|J_m(t)| \leq \frac{1}{m!}\left|\frac{t}{2}\right|^m. \qquad (55)$$

Following [BCK15] we show that it implies that for any positive integer $q \geq |t| - 1$ we have that

$$2\sum_{\ell=0}^{\infty}|J_{(q+2\ell)}(t)| \overset{(55)}{\leq} 2\sum_{\ell=0}^{\infty}\frac{|t/2|^{(q+2\ell)}}{(q+2\ell)!} \leq 2\frac{|t/2|^q}{q!}\sum_{\ell=0}^{\infty}\left(\frac{1}{4}\right)^{\ell} = \frac{8}{3}\frac{|t/2|^q}{q!} \leq \frac{1.07}{\sqrt{q}}\left(\frac{e|t|}{2q}\right)^q, \qquad (56)$$

where in the last inequality we used that by Stirling's approximation $q! \geq \sqrt{2\pi q}\left(\frac{q}{e}\right)^q$. In the inequalities (53)-(54) we can apply the bound of (56) with $q \geq 2(R+1) \geq r\left(\frac{e|t|}{2}, \varepsilon\right)$, so we get the upper bound

$$\frac{1.07}{\sqrt{q}}\left(\frac{e|t|}{2q}\right)^q \leq \frac{1.07}{\sqrt{2}}\left(\frac{e|t|}{2q}\right)^q \leq \frac{5}{4}\left(\frac{e|t|}{2q}\right)^q \leq \varepsilon.$$

□

Now we are ready to prove the optimal block-Hamiltonian simulation result of Low and Chuang. The optimality is discussed in an earlier work of the same authors [LC17b].

Theorem 58. (Optimal block-Hamiltonian simulation [LC16]) *Let $t \in \mathbb{R} \setminus \{0\}$, $\varepsilon \in (0,1)$ and let U be an $(\alpha, a, 0)$-block-encoding of the Hamiltonian H. Then we can implement an ε-precise Hamiltonian simulation unitary V which is an $(1, a+2, \varepsilon)$-block-encoding of e^{itH}, with $3r\left(\frac{e\alpha|t|}{2}, \frac{\varepsilon}{6}\right)$ uses of U or its inverse, 3 uses of controlled-U or its inverse and with $\mathcal{O}\left(ar\left(\frac{e\alpha|t|}{2}, \frac{\varepsilon}{6}\right)\right)$ two-qubit gates and using $\mathcal{O}(1)$ ancilla qubits.*

Proof. Use the polynomials of Lemma 57 and combine the even real polynomial $\frac{\varepsilon}{6}$-approximating $\cos(\alpha t)$ with the odd imaginary polynomial $\frac{\varepsilon}{6}$-approximating $i \cdot \sin(\alpha t)$ using the same method as Theorem 56 in order to get an $(1, a+2, \frac{\varepsilon}{6})$-block-encoding of $e^{itH}/2$. Then use robust oblivious amplitude amplification Corollary 28 in order get an $(1, a+2, \varepsilon)$-block-encoding of e^{itH}. □

Now we prove some bounds on $r(t, \varepsilon)$, in order to make the above result more accessible.

Lemma 59 (Bounds on $r(t, \varepsilon)$)**.** *For $t \in \mathbb{R}_+$ and $\varepsilon \in (0,1)$*

$$r(t, \varepsilon) = \Theta\left(t + \frac{\ln(1/\varepsilon)}{\ln(e + \ln(1/\varepsilon)/t)}\right).$$

Moreover, for all $q \in \mathbb{R}_+$ we have that

$$r(t, \varepsilon) < e^q t + \frac{\ln(1/\varepsilon)}{q}.$$

Proof. First consider the case $t \geq \frac{\ln(1/\varepsilon)}{e}$ and set $r := et$, then we get that

$$\forall t \geq \frac{\ln(1/\varepsilon)}{e} \colon \left(\frac{t}{et}\right)^{et} = \left(\frac{1}{e}\right)^{et} \leq \varepsilon \qquad \Longrightarrow \qquad \forall t \geq \frac{\ln(1/\varepsilon)}{e} \colon r(t, \varepsilon) \leq et. \tag{57}$$

Now we turn to the case $t \leq \frac{\ln(1/\varepsilon)}{e}$, and try to find $r = r(t, \varepsilon)$.

$$\left(\frac{t}{r}\right)^r = \varepsilon \qquad \Longleftrightarrow \qquad \left(\frac{r}{t}\right)^{\frac{r}{t}} = \left(\frac{1}{\varepsilon}\right)^{\frac{1}{t}} \qquad \Longleftrightarrow \qquad \frac{r}{t} \ln\left(\frac{r}{t}\right) = \ln\left(\frac{1}{\varepsilon}\right)\frac{1}{t}. \tag{58}$$

Let us define $x := \frac{r}{t} \geq 1$ and $c := \ln\left(\frac{1}{\varepsilon}\right)\frac{1}{t} \geq e$. We will examine the solution of the equation $x \ln(x) = c$ for $c \geq e$. We see that the function $x \ln(x)$ is monotone increasing on $[1, \infty)$, takes value 0 at 1 and in the $x \to \infty$ limit it tends to infinity, therefore the equation $x \ln(x) = c$ has a unique solution for all $c \in \mathbb{R}_+$. Moreover, if $b, B \in [1, \infty)$ are such that $b \ln(b) \leq c \leq B \ln(B)$, then $b \leq x \leq B$. Therefore we can see that $\frac{c}{\ln(c)} \leq x$ since

$$\frac{c}{\ln(c)} \ln\left(\frac{c}{\ln(c)}\right) = \frac{c}{\ln(c)}(\ln(c) - \ln(\ln(c))) = c\left(1 - \frac{\ln(\ln(c))}{\ln(c)}\right) \leq c.$$

By a similar argument we can see that $x \leq \frac{5}{3}\frac{e+c}{\log(e+c)} \leq \frac{4c}{\log(e+c)}$, since

$$\begin{aligned}\frac{5}{3}\frac{e+c}{\ln(e+c)}\ln\left(\frac{5}{3}\frac{e+c}{\ln(e+c)}\right) &> \frac{5}{3}\frac{e+c}{\ln(e+c)}\ln\left(\frac{e+c}{\ln(e+c)}\right)\\ &= \frac{5}{3}\frac{e+c}{\ln(e+c)}(\ln(e+c)-\ln(\ln(e+c)))\\ &= \frac{5}{3}(e+c)\left(1-\frac{\ln(\ln(e+c))}{\ln(e+c)}\right)\\ &\geq \frac{5}{3}(e+c)\left(1-\frac{1}{e}\right) \qquad \left(\forall y\in\mathbb{R}_+\colon \tfrac{\ln(y)}{y}\leq\tfrac{1}{e}\right)\\ &> e+c\\ &> c.\end{aligned}$$

Thus for $x \geq 1$, $c \geq e$ we get that the solution of the equation $x\log(x) = c$ satisfies

$$\frac{c}{\log(e+c)} \leq \frac{c}{\ln(c)} \leq x \leq \frac{4c}{\log(e+c)}. \tag{59}$$

Using $x = \frac{r}{t} \Rightarrow r = tx$ and $c = \ln\left(\frac{1}{\varepsilon}\right)\frac{1}{t}$ from (58)-(59) we get that

$$\forall t \leq \frac{\ln(1/\varepsilon)}{e}\colon \frac{\ln(1/\varepsilon)}{\ln(e+\ln(1/\varepsilon)/t)} \leq r(t,\varepsilon) \leq \frac{4\ln(1/\varepsilon)}{\ln(e+\ln(1/\varepsilon)/t)}. \tag{60}$$

Combining (57) and (60) while observing $t \leq r(t,\varepsilon)$ and $\frac{\ln(1/\varepsilon)}{\ln(e+\ln(1/\varepsilon)/t)} \leq \ln(1/\varepsilon)$, we get that

$$\forall\varepsilon\in(0,1)\forall t\in\mathbb{R}_+\colon r(t,\varepsilon) = \Theta\left(t+\frac{\ln(1/\varepsilon)}{\ln(e+\ln(1/\varepsilon)/t)}\right). \tag{61}$$

Finally note that for $r_q := e^q t + \ln(1/\varepsilon)/q$, then we get

$$\left(\frac{t}{r_1}\right)^{r_q} \leq \left(e^{-q}\right)^{r_q} \leq e^{-\ln(1/\varepsilon)} = \varepsilon \qquad \Longrightarrow \qquad r(t,\varepsilon) \leq r_q.$$

□

This enables us to conclude the complexity of block-Hamiltonian simulation. Note that for $t \leq \varepsilon$ Hamiltonian simulation with ε-precision is trivial if $\|H\| \leq 1$, therefore we should assume that $t = \Omega(\varepsilon)$ in order to avoid this trivial situation. Apart from this we can conclude the complexity of block-Hamiltonian simulation for entire range of interesting parameters.

Corollary 60 (Complexity of block-Hamiltonian simulation)**.** *Let $\varepsilon \in (0,\frac{1}{2})$, $t \in \mathbb{R}$ and $\alpha \in \mathbb{R}_+$. Let U be an $(\alpha, a, 0)$-block-encoding of the unknown Hamiltonian H. In order to implement an ε-precise Hamiltonian simulation unitary V which is an $(1, a+2, \varepsilon)$-block-encoding of e^{itH}, it is necessary and sufficient to use the unitary U a total number of times*

$$\Theta\left(\alpha|t| + \frac{\log(1/\varepsilon)}{\log(e+\log(1/\varepsilon)/(\alpha|t|))}\right).$$

Proof. The upper bound follows from Theorem 58 and Lemma 59. The lower bound follows from the argument laid out in [LC17b] using Lemma 59. □

Note that the above corollary also covers the range $t \ll 1$, unlike the result of Low and Chuang [LC16] who assumed $t = \Omega(1)$. Also note that this result does not entirely match the complexity stated by Low and Chuang [LC17b, LC16]. For example in the case $t = \frac{\log(1/\varepsilon)}{\log(\log(1/\varepsilon))}$ the above corollary shows that the complexity is $\Theta\left(\frac{\log(1/\varepsilon)}{\log(\log(\log(1/\varepsilon)))}\right)$, whereas the expression of [LC17b, LC16] claims complexity $\mathcal{O}\left(\frac{\log(1/\varepsilon)}{\log(\log(1/\varepsilon))}\right)$.

The following lemma of Chakraborty et al. [CGJ18, Appendix A] helps us to understand error accumulation in Hamiltonian simulation, which enables us to present a slightly improved claim in Theorem 62.

Lemma 61. *Let $t \in \mathbb{R}$ and $H, H' \in \mathbb{C}^{n\times n}$ Hermitian operators, then*

$$\left\| e^{itH} - e^{itH'} \right\| \leq |t| \|H - H'\|.$$

Now we prove a robust version of Theorem 58 using Lemma 61, and also substitute a simple expression of Lemma 59 bounding $r(t,\varepsilon)$ in order to get explicit constants.

Corollary 62. (Robust block-Hamiltonian simulation) *Let $t \in \mathbb{R}$, $\varepsilon \in (0,1)$ and let U be an $(\alpha, a, \varepsilon/|2t|)$-block-encoding of the Hamiltonian H. Then we can implement an ε-precise Hamiltonian simulation unitary V which is an $(1, a+2, \varepsilon)$-block-encoding of e^{itH}, with $6\alpha|t| + 9\log(12/\varepsilon)$ uses of U or its inverse, 3 uses of controlled-U or its inverse, using $\mathcal{O}(a(\alpha|t| + \log(2/\varepsilon)))$ two-qubit gates and using $\mathcal{O}(1)$ ancilla qubits.*

Proof. Let $H' = \alpha(\langle 0|^{\otimes a} \otimes I)U(|0\rangle^{\otimes a} \otimes I)$, then $\|H' - H\| \leq \varepsilon/|2t|$. By Theorem 58 we can implement V an $(1, a+2, \varepsilon/2)$-block-encoding of $e^{itH'}$, with $3r\left(\frac{e\alpha|t|}{2}, \frac{\varepsilon}{12}\right)$ uses of U or its inverse, 3 uses of controlled-U or its inverse and with $\mathcal{O}\left(ar\left(\frac{e\alpha|t|}{2}, \frac{\varepsilon}{12}\right)\right)$ two-qubit gates and using $\mathcal{O}(1)$ ancilla qubits. By Lemma 61 we get that V is an $(1, a+2, \varepsilon)$-block-encoding of e^{itH}. Finally by Lemma 59 choosing $q := \frac{1}{3}$ we get that

$$r\left(\frac{e\alpha|t|}{2}, \frac{\varepsilon}{12}\right) \leq e^{q}\frac{e\alpha|t|}{2} + \frac{\ln(12/\varepsilon)}{q} \leq 2\alpha|t| + 3\ln(12/\varepsilon).$$

□

5.2 Bounded polynomial approximations of piecewise smooth functions

We begin by invoking a slightly surprising result showing how to efficiently approximate monomials on the interval $[-1,1]$ with essentially quadratically smaller degree polynomials than the monomial itself. The following theorem can be found in the survey of Sachdeva and Vishnoi [SV14, Theorem 3.3].

Theorem 63 (Efficient approximation of monomials on $[-1,1]$)**.** *For any positive integers s and d, there exists an efficiently computable degree-d polynomial $P_{s,d} \in \mathbb{R}[x]$ that satisfies*

$$\|P_{s,d}(x) - x^s\|_{[-1,1]} \leq 2e^{-d^2/(2s)}.$$

If one wants to approximate smooth functions on the entire $[-1,1]$ interval this result gives essentially quadratic savings. For example one can easily derive Corollary 64 using the above result as shown in [SV14].

Corollary 64 (Polynomial approximations of the exponential function)**.** *Let $\beta \in \mathbb{R}_+$ and $\varepsilon \in (0, \frac{1}{2}]$. There exists an efficiently constructable polynomial $P \in \mathbb{R}[x]$ such that*

$$\left\| e^{-\beta(1-x)} - P(x) \right\|_{[-1,1]} \leq \varepsilon,$$

and the degree of P is $\mathcal{O}\left(\sqrt{\max\left[\beta, \log(\frac{1}{\varepsilon})\right] \log(\frac{1}{\varepsilon})}\right)$.

However, we often want to implement functions that are smooth only on some compact subset of $C \subseteq [-1,1]$, which requires different techniques. The main difficulty is to achieve a good approximation on C, while keeping the norm of the approximating polynomial bounded on the whole $[-1,1]$ interval. We overcome this difficulty by using Fourier approximations on C, which give rise to bounded functions naturally. Later we convert these Fourier series to a polynomial using Lemma 57 from the previous subsection.

Apeldoorn et al [AGGW17, Appendix B] developed techniques that make it possible to implement smooth-functions of a Hamiltonian H, based on Fourier series decompositions and using the Linear Combinations of Unitaries (LCU) Lemma [BCK15]. The techniques developed in [AGGW17, Appendix B] access H only through controlled-Hamiltonian simulation, which step can be omitted using singular value transformation techniques by constructing the corresponding bounded low-degree polynomials.

Now we invoke one of the main technical results of [AGGW17, Lemma 37] about approximating smooth functions by low-weight Fourier series. By low weight we mean that the 1-norm of the coefficients is small. A notable property of the following result is that the bounds on the Fourier series do not depend on the degrees of the polynomials terms. This can however be expected since the terms that have large degree make negligible contribution due to the restricted domain $x \in [-1+\delta, 1-\delta]$, and therefore we can drop them without loss of generality.

Lemma 65 (Low weight approximation by Fourier series)**.** *Let $\delta, \varepsilon \in (0,1)$ and $f : \mathbb{R} \to \mathbb{C}$ s.t. $\left| f(x) - \sum_{k=0}^{K} a_k x^k \right| \leq \varepsilon/4$ for all $x \in [-1+\delta, 1-\delta]$. Then $\exists c \in \mathbb{C}^{2M+1}$ such that*

$$\left| f(x) - \sum_{m=-M}^{M} c_m e^{\frac{i\pi m}{2}x} \right| \leq \varepsilon$$

for all $x \in [-1+\delta, 1-\delta]$, where $M = \max\left(2\left\lceil \ln\left(\frac{4\|a\|_1}{\varepsilon}\right)\frac{1}{\delta}\right\rceil, 0\right)$ and $\|c\|_1 \leq \|a\|_1$. Moreover c can be efficiently calculated on a classical computer in time poly$(K, M, \log(1/\varepsilon))$.

Our main idea is to combine this result with the polynomial approximations of the trigonometric functions as in Lemma 57. The low weights are useful because they let us reduce the precision required for approximating the Fourier terms.

Corollary 66 (Bonded polynomial approximations based on a local Taylor series)**.** *Let $x_0 \in [-1,1]$, $r \in (0,2]$, $\delta \in (0,r]$ and let $f \colon [-x_0-r-\delta, x_0+r+\delta] \to \mathbb{C}$ and be such that $f(x_0+x) = \sum_{\ell=0}^{\infty} a_\ell x^\ell$*

for all $x \in [-r-\delta, r+\delta]$. Suppose $B > 0$ is such that $\sum_{\ell=0}^{\infty}(r+\delta)^{\ell}|a_{\ell}| \leq B$. Let $\varepsilon \in \left(0, \frac{1}{2B}\right]$, then there is an efficiently computable polynomial $P \in \mathbb{C}[x]$ of degree $\mathcal{O}\left(\frac{1}{\delta}\log\left(\frac{B}{\varepsilon}\right)\right)$ such that

$$\|f(x) - P(x)\|_{[x_0-r, x_0+r]} \leq \varepsilon \tag{62}$$

$$\|P(x)\|_{[-1,1]} \leq \varepsilon + \|f(x)\|_{[x_0-r-\delta/2, x_0+r+\delta/2]} \leq \varepsilon + B \tag{63}$$

$$\|P(x)\|_{[-1,1]\setminus[x_0-r-\delta/2, x_0+r+\delta/2]} \leq \varepsilon. \tag{64}$$

Proof. We proceed similarly to the proof of [AGGW17, Theorem 40]. Let $L(x) := \frac{x-x_0}{r+\delta}$ be the linear transformation taking $[x_0-r-\delta, x_0+r+\delta]$ to $[-1,1]$. Let $g(y) := f(L^{-1}(y))$, and $b_{\ell} := a_{\ell}(r+\delta)^{\ell}$ such that $g(y) = \sum_{\ell=0}^{\infty} b_{\ell} y^{\ell}$. Let $\delta' := \frac{\delta}{2(r+\delta)}$ and let $J = \left\lceil \frac{1}{\delta'}\log(\frac{12B}{\varepsilon})\right\rceil$, then for all $y \in [-1,1]$ we have that

$$\left|g(y) - \sum_{j=0}^{J-1} b_j y^j\right| = \left|\sum_{j=J}^{\infty} b_j y^j\right| \leq \sum_{j=J}^{\infty}|b_j(1-\delta')^j| \leq (1-\delta')^J \sum_{j=J}^{\infty}|b_j| \leq \left(1-\delta'\right)^J B \leq e^{-\delta' J}B \leq \frac{\varepsilon}{12}.$$

Now we construct a Fourier-approximation of g for all $y \in [-1+\delta', 1-\delta']$, with precision $\frac{\varepsilon}{3}$. Let $b' := (b_0, b_1, \ldots, b_{J-1})$ and observe that $\|b'\|_1 \leq \|b\|_1 \leq B$. We apply Lemma 65 to the function g, using the polynomial approximation corresponding to the truncation to the first J terms, i.e., using the coefficients in b'. Then we obtain a Fourier $\frac{\varepsilon}{3}$-approximation $\tilde{g}(y) := \sum_{m=-M}^{M} \tilde{c}_m e^{\frac{i\pi m}{2}y}$ of g, with

$$M = \mathcal{O}\left(\frac{1}{\delta'}\log\left(\frac{\|b'\|_1}{\varepsilon'}\right)\right) = \mathcal{O}\left(\frac{r}{\delta}\log\left(\frac{B}{\varepsilon}\right)\right),$$

such that the vector of coefficients $\tilde{c} \in \mathbb{C}^{2M+1}$ satisfies $\|\tilde{c}\|_1 \leq \|b'\|_1 \leq \|b\|_1 \leq B$. Let

$$\tilde{f}(x) := \tilde{g}(L(x)) = \tilde{g}\left(\frac{x-x_0}{r+\delta}\right) = \sum_{m=-M}^{M} \tilde{c}_m e^{\frac{i\pi m}{2(r+\delta)}(x-x_0)} = \sum_{m=-M}^{M} \tilde{c}_m e^{-\frac{i\pi m}{2(r+\delta)}x_0} e^{\frac{i\pi m}{2(r+\delta)}x}.$$

Since $g(y) = f(L^{-1}(y))$ we have that $f(x) = g(L(x))$ thus we can see that $\tilde{f}$ is an $\frac{\varepsilon}{3}$-precise Fourier approximation of f on the interval $[x_0 - r - \frac{\delta}{2}, x_0 + r + \frac{\delta}{2}]$. Now we define $\tilde{P}$ as the polynomial that we get by replacing each of the Fourier terms $e^{\frac{i\pi m}{2(r+\delta)}x}$ by $\frac{\varepsilon}{3B}$-approximating polynomials given by Lemma 57. Using a tiny rescaling we can assure that the polynomial approximations of $e^{\frac{i\pi m}{2(r+\delta)}x}$ have absolute value at most 1 on $[-1,1]$. Moreover by Lemma 59 we know that the degree of these polynomials are $\mathcal{O}\left(\frac{M}{r+\delta} + \log\left(\frac{B}{\varepsilon}\right)\right) = \mathcal{O}\left(\frac{1}{\delta}\log\left(\frac{B}{\varepsilon}\right)\right)$. Since $\|\tilde{c}\| \leq B$, we get that the absolute value of the polynomial $\tilde{P}$ is bounded by B on the interval $[-1,1]$. Finally we define P as the product of $\tilde{P}$ and an approximation polynomial of the rectangle function that is $\frac{\varepsilon}{3B}$-close to 1 on the interval $[x_0 - r, x_0 + r]$, and is $\frac{\varepsilon}{3B}$-close to 0 on the interval $[-1,1] \setminus [x_0 - r - \frac{\delta}{2}, x_0 + r + \frac{\delta}{2}]$, finally which is bounded by 1 on the interval $[-1,1]$ in absolute value. By Lemma 29 we can construct such a polynomial of degree $\mathcal{O}\left(\frac{1}{\delta}\log\left(\frac{B}{\varepsilon}\right)\right)$. As we can see P has degree $\mathcal{O}\left(\frac{1}{\delta}\log\left(\frac{B}{\varepsilon}\right)\right)$, and by construction satisfies the required properties (62)-(64). □

Combining this polynomial approximation result with Theorem 56 we can efficiently implement smooth functions of Hermitian matrices. As an application, motivated by the work of Chakraborty et al. [CGJ18] we show how to construct low-degree polynomial approximations of power functions.

Corollary 67 (Polynomial approximations of negative power functions)**.** *Let $\delta, \varepsilon \in (0, \frac{1}{2}]$, $c > 0$ and let $f(x) := \frac{\delta^c}{2} x^{-c}$, then there exist even/odd polynomials $P, P' \in \mathbb{R}[x]$, such that $\|P - f\|_{[\delta,1]} \leq \varepsilon$, $\|P\|_{[-1,1]} \leq 1$ and similarly $\|P' - f\|_{[\delta,1]} \leq \varepsilon$, $\|P'\|_{[-1,1]} \leq 1$, moreover the degree of the polynomials are $\mathcal{O}\Big(\frac{\max[1,c]}{\delta} \log\big(\frac{1}{\varepsilon}\big)\Big)$.*

Proof. First note that for all $y \in (-1, 1)$ we have that $(1 + y)^{-c} = \sum_{k=0}^{\infty} \binom{-c}{k} y^k$. We first find a polynomial $\tilde{P} \in \mathbb{C}[x]$ such that $\Big\|\tilde{P} - f\Big\|_{[\delta,1]} \leq \frac{\varepsilon}{2}$, $\Big\|\tilde{P}\Big\|_{[-1,0]} \leq \frac{\varepsilon}{2}$ and $\Big\|\tilde{P}\Big\|_{[-1,1]} \leq 1$. We construct such a polynomial of degree $\mathcal{O}\Big(\frac{\max[1,c]}{\delta} \log\big(\frac{1}{\varepsilon}\big)\Big)$ using Corollary 66, with choosing $x_0 := 0$, $r := 1-\delta$, $\delta' := \frac{\delta}{2\max[1,c]}$ and $B := 1$. The choice of B is justified by the observation that

$$\begin{aligned}
\frac{\delta^c}{2} \sum_{k=0}^{\infty} \left| \binom{-c}{k} \right| (r + \delta')^k = \frac{\delta^c}{2} \sum_{k=0}^{\infty} \binom{-c}{k} (-r - \delta')^k &= \frac{\delta^c}{2} (1 - r - \delta')^{-c} \\
&= \frac{\delta^c}{2} (\delta - \delta')^{-c} = \frac{1}{2} \left(1 - \frac{\delta'}{\delta}\right)^{-c} \\
&= \frac{1}{2} \left(1 - \frac{1}{2\max[1, c]}\right)^{-c} \leq 1.
\end{aligned}$$

Finally, we define P as the even real part of $\tilde{P}$, and define P' as the odd real part of $\tilde{P}$. □

Given a $(1, a, 0)$-block-encoding of A, with the promise that the spectrum of A lies in $[\delta, 1]$, using the above polynomials and Theorem 56 we can implement a $(1, a + 2, \varepsilon)$-block-encoding of $\frac{\delta^c}{2} A^{-c}$ with $\mathcal{O}\Big(\frac{\max[1,c]}{\delta} \log\big(\frac{1}{\varepsilon}\big)\Big)$ uses of the block-encoding of A. Since the derivative of the function $\frac{\delta^c}{2} x^{-c}$ at $x = \delta$ is $-\frac{c}{2\delta}$, we get by Theorem 73 that the δ and c dependence of the complexity of this procedure is optimal.

Finally we develop a theorem that is analogous to [AGGW17, Corollary 42], and shows that any function that has quickly converging local Taylor-series can in principle be ε-approximated with complexity $\propto \log\big(\frac{1}{\varepsilon}\big)$.

Theorem 68 (Bounded polynomial approximation based on multiple local Taylor series)**.** *Let $J \in \mathbb{N}$, $(x_j, r_j, \delta_j) \in [-1, 1]^J \times (0, 2]^J \times (0, 1]^J$, such that $x_j\colon j \in [J]$ is monotone increasing, and $\delta_j \leq r_j$ for all $j \in [J]$. Let $I := \bigcup_{j\in[J]} [x_j - r_j, x_j + r_j]$ be the union of the intervals $[x_j - r_j, x_j + r_j]$, and suppose that for all $i < j \in [J]$ such that $j - i \geq 2$ we have that $r_j + r_j < x_j - x_j$. Let $\delta = \min\big[\min_{j\in[J]} \delta_j, \min_{j\in[J-1]} |x_{j+1} - x_j - (r_{j+1} + r_j)|\big]$. Let $f : I + [-\frac{\delta}{2}, \frac{\delta}{2}] \to \mathbb{C}$, $B \in \mathbb{R}_+$ be such that for all $j \in [J]$ we have $f(x_j + x) = \sum_{k=0}^{\infty} a_k^{(j)} x^k$ for all $x \in [x_j - r_j - \frac{\delta_j}{2}, x_j + r_j + \frac{\delta_j}{2}]$ and $\sum_{k=0}^{\infty} (r_j + \delta_j)^k |a_k^{(j)}| \leq B$. Let $\varepsilon \in \big(0, \frac{1}{2BJ}\big]$, then there is an efficiently computable polynomial $P \in \mathbb{C}[x]$ of degree $\mathcal{O}\big(\frac{J}{\delta} \log\big(\frac{BJ}{\varepsilon}\big)\big)$ such that*

$$\begin{aligned}
\|f(x) - P(x)\|_I &\leq \varepsilon \\
\|P(x)\|_{[-1,1]} &\leq \|f(x)\|_{I+[-\delta/2,\delta/2]} \\
\|P(x)\|_{[-1,1]\setminus(I+[-\delta/2,\delta/2])} &\leq \varepsilon.
\end{aligned}$$

Proof. Use Corollary 66 to construct polynomials $f_j\colon j \in [J]$ of degree $\mathcal{O}\big(\frac{1}{\delta}\log\big(\frac{BJ}{\varepsilon}\big)\big)$ such that

$$\begin{aligned}
\|f(x) - f_j(x)\|_{[x_j - r_j, x_j + r_j]} &\le \frac{\varepsilon}{4J} \\
\|f_j(x)\|_{[-1,1]} &\le \|f(x)\|_{I + [-\delta/2, \delta/2]} \\
\|f_j(x)\|_{[-1,1]\setminus([x_j - r_j, x_j + r_j] + [-\delta/2,\delta/2])} &\le \varepsilon.
\end{aligned}$$

Let us introduce a notation for the union of the intervals $[x_i - r_i, x_i + r_i]\colon i \in \{j, j+1, \ldots, k\}$ as

$$I_{[j,k]} := \bigcup_{i\in\{j,j+1,\ldots,k\}} [x_i - r_i, x_i + r_i].$$

We show inductively how to construct polynomials $f_{[j,k]}$ of degree $\mathcal{O}\Big(\frac{k-j+1}{\delta}\log\big(\frac{BJ}{\varepsilon}\big)\Big)$ such that

$$\left\|f(x) - f_{[j,k]}(x)\right\|_{I_{[j,k]}} \le \frac{2(k-j+1)\varepsilon}{2J} \tag{65}$$

$$\left\|f_{[j,k]}(x)\right\|_{[-1,1]} \le \|f(x)\|_{I+[-\delta/2,\delta/2]} \tag{66}$$

$$\left\|f_{[j,k]}(x)\right\|_{[-1,1]\setminus\left(I_{[j,k]}+[-\delta/2,\delta/2]\right)} \le \varepsilon. \tag{67}$$

We already showed how to construct $f_{[j,j]} := f_j\colon j \in [J]$. Suppose that we already constructed $f_{[1,j]}$, then we construct $f_{[1,j+1]}$ as follows. We take a polynomial $S(x)$ of degree $\mathcal{O}\big(\frac{1}{\delta}\log\big(\frac{BJ}{\varepsilon}\big)\big)$ that approximates the shifted sign function s.t. $\left\|S(x) - \mathrm{sign}\Big(x - \frac{x_i+x_j}{2}\Big)\right\|_{[-1,1]\setminus\left[\frac{x_i+x_j-\delta}{2},\frac{x_i+x_j+\delta}{2}\right]} \le \frac{\varepsilon}{8BJ}$, moreover $\|S\|_{[-1,1]} \le 1$. Then we define $f_{[1,j+1]} := \frac{1-S(x)}{2} f_{[1,j]} + \frac{1+S(x)}{2} f_{[j+1,j+1]}$. It satisfies (66)-(67), since $f_{[1,j+1]}$ is a point-wise convex combination of $f_{[1,j]}$ and $f_{[j+1,j+1]}$. Similarly (65) is also easy to verify. Therefore by induction we can finally construct $P := f_{[1,J]}$, which satisfies (65)-(67) and therefore also the requirements of the theorem.[26] □

A direct corollary of this theorem is for example that for all $\varepsilon \in (0, \frac{1}{2}]$ the function $\frac{\delta}{x}$ can be ε-approximated on the domain $[-1,1]\setminus[-\delta,\delta]$ with a polynomial of degree $\mathcal{O}\big(\frac{1}{\delta}\log\big(\frac{1}{\varepsilon}\big)\big)$. Although this also follows from Corollary 67, we prove it directly using Theorem 68, in order to illustrate its usefulness.

Corollary 69 (Polynomial approximations of $\frac{1}{x}$)**.** *Let $\varepsilon, \delta \in (0, \frac{1}{2}]$, then there is an odd polynomial $P \in \mathbb{R}[x]$ of degree $\mathcal{O}\big(\frac{1}{\delta}\log\big(\frac{1}{\varepsilon}\big)\big)$ that is ε-approximating $f(x) = \frac{3}{4}\frac{\delta}{x}$ on the domain $I = [-1,1]\setminus[-\delta,\delta]$, moreover it is bounded 1 in absolute value.*

Proof. Take $J := 2$, $(x_1 := -1, r_1 := 1-\delta, \delta_1 := \frac{\delta}{2})$, $(x_2 := 1, r_2 := 1-\delta, \delta_2 := \frac{\delta}{2})$ and $B = 1$ in Theorem 68, observing that $f(1+x) = \frac{3\delta}{4}\sum_{k=0}^{\infty} -(1)^k x^k = -f(-1+x)$. Define P as the odd real part of the polynomial given by Theorem 68. □

[26] Note that this approach could be further improved to produce a degree $\mathcal{O}\Big(\frac{\log(J)}{\delta}\log\Big(\frac{B\log(J)}{\varepsilon}\Big)\Big)$ approximating polynomial by combining the polynomial approximations on the different intervals in a binary tree structure. Since $J = \mathcal{O}\big(\frac{1}{\delta}\big)$, $\log(J) = \log\big(\frac{1}{\delta}\big)$ and then this gives at most a logarithmic overhead.

5.3 Applications: fractional queries and Gibbs sampling

Scott Aaronson listed as one of "The ten most annoying questions in quantum computing" [Aar06] the following problem: given a unitary U, can we implement $\sqrt{U}$? This was positively answered by Sheridan et al. [SMM09]. We show how to improve the complexity of the result of Sheridan et al. exponentially in terms of the error dependence. We proceed follow ideas of Low and Chuang [LC17a].

Suppose that we have access to a unitary $U = e^{iH}$, where H is a Hamiltonian of norm at most $\frac{1}{2}$. Low and Chuang [LC17a] showed how to get a $(1, 2, \varepsilon)$-block-encoding of H with $\mathcal{O}\big(\log\big(\frac{1}{\varepsilon}\big)\big)$ uses of U. We reprove this result; our proof becomes quite simple thanks to Corollary 66.

Lemma 70 (Polynomial approximations of $\arcsin(x)$)**.** *Let $\delta, \varepsilon \in (0, \frac{1}{2}]$, there is an efficiently computable odd real polynomial $P \in \mathbb{R}[x]$ of degree $\mathcal{O}\big(\frac{1}{\delta}\log\big(\frac{1}{\varepsilon}\big)\big)$ such that $\|P\|_{[-1,1]} \leq 1$ and*

$$\left\| P(x) - \frac{2}{\pi}\arcsin(x) \right\|_{[-1+\delta, 1-\delta]} \leq \varepsilon.$$

Proof. Observe that $\frac{2}{\pi}\arcsin(x) = \sum_{\ell=0}^{\infty} \binom{2\ell}{\ell} \frac{2^{-2\ell}}{2\ell+1} \frac{2}{\pi} x^{2\ell+1}$ for all $x \in [-1, 1]$. Therefore we also have $\sum_{\ell=0}^{\infty} \left| \binom{2\ell}{\ell} \frac{2^{-2\ell}}{2\ell+1} \frac{2}{\pi} \right| = 1$. The result immediately follows by taking the odd real part of the polynomial given by Corollary 66. □

Corollary 71 (Implementing the logarithm of unitaries)**.** *Suppose that $U = e^{iH}$, where H is a Hamiltonian of norm at most $\frac{1}{2}$. Let $\varepsilon \in (0, \frac{1}{2}]$, then we can implement a $(\frac{2}{\pi}, 2, \varepsilon)$-block-encoding of H with $\mathcal{O}\big(\log\big(\frac{1}{\varepsilon}\big)\big)$ uses of controlled-U and its inverse, using $\mathcal{O}\big(\log\big(\frac{1}{\varepsilon}\big)\big)$ two-qubit gates and using a single ancilla qubit.*

Proof. Let cU denote the controlled version of U controlled by the first qubit. Then

$$\sin(H) = -i(\langle +| \otimes I) cU^{\dagger} (ZX \otimes I) cU (|+\rangle \otimes I).$$

Now we apply singular value transformation Corollary 18 using an ε-approximating polynomial of $\frac{2}{\pi}\arcsin(x)$ on the domain $[-\frac{1}{2}, \frac{1}{2}]$. □

Combining the above result with block-Hamiltonian simulation techniques Corollary 62 we can implement fractional queries of unitaries with complexity $\mathcal{O}\big(\log^2\big(\frac{1}{\varepsilon}\big)\big)$. As we show in the following corollary this complexity can be reduced to $\mathcal{O}\big(\log\big(\frac{1}{\varepsilon}\big)\big)$ by directly implementing[27] Hamiltonian simulation using a block-encoding of $\sin(H)$ rather than H.

Corollary 72 (Implementing fractional queries)**.** *Suppose that $U = e^{iH}$, where H is a Hamiltonian of norm at most $\frac{1}{2}$. Let $\varepsilon \in (0, \frac{1}{2}]$ and $t \in [-1, 1]$, then we can implement an ε-approximation of $U^t = e^{itH}$ with $\mathcal{O}\big(\log\big(\frac{1}{\varepsilon}\big)\big)$ uses of controlled-U and its inverse, using $\mathcal{O}\big(\log\big(\frac{1}{\varepsilon}\big)\big)$ two-qubit gates and using $\mathcal{O}(1)$ ancilla qubits.*

Proof. As we have shown in the proof of Corollary 71, one can implement a block-encoding of $\sin(H)$ with a constant number of queries to U. Let us look at the Taylor series of $e^{it\arcsin(x)}$. One can see that the 1-norm of the coefficients of the Taylor series of $t\arcsin(x)$ is $|t|\arcsin(1) = |t|\frac{\pi}{2}$. Therefore, for $t \in [-\frac{2}{\pi}, \frac{2}{\pi}]$ we get that the 1-norm of the Taylor series of $e^{it\arcsin(x)}$ is at most $e^1 = e$.

[27] The method we describe uses block-encoding formalism, but in fact one could implement it more directly using a Fourier series based approach similarly to the one used for Hamiltonian simulation by Low and Chuang [LC17b].

Thereby, using Theorem 68 we can construct polynomial $\mathcal{O}(\varepsilon)$-approximations of $\sin(t\arcsin(x))$ and $\cos(t\arcsin(x))$ of degree $\mathcal{O}\big(\log\big(\frac{1}{\varepsilon}\big)\big)$, which are bounded by 1 in absolute value on the interval $[-1,1]$. We can combine these polynomials in a similar way as in the proof of Theorem 58. This way we can implement an ε-approximation of U^t for all $t \in [-\frac{2}{\pi}, \frac{2}{\pi}]$ with complexity $\mathcal{O}\big(\log\big(\frac{1}{\varepsilon}\big)\big)$. Implementing U^t for all $t \in [-1,1]$ can be achieved by implementing $U^{\frac{t}{2}}$ twice an taking their product. □

Note that the above technique can be combined with an initial phase estimation in order to implement fractional queries under the weaker promise $\|H\| \leq \pi - \delta$. It suffices to perform a δ-precise phase estimation with success probability $1 - \mathrm{poly}(\varepsilon)$, then implement fractional queries using Corollary 72 and then undo the initial phase estimation. This leads to complexity $\mathcal{O}\big(\frac{1}{\delta}\log\big(\frac{1}{\varepsilon}\big)\big)$, which exponentially improves the complexity $\mathcal{O}\big(\max[\frac{1}{\delta}, \frac{1}{\varepsilon}]\log\big(\frac{1}{\varepsilon}\big)\big)$ of Sheridan et al. [SMM09] in the case of $\delta = \Theta(1)$. Note that Sheridan et al. [SMM09] also proved a lower bound on this problem, which shows that the δ dependence of this algorithm is actually optimal. We believe that the $\log(\frac{1}{\varepsilon})$ dependence in the runtime is also necessary, therefore this algorithm is probably fairly close to optimal.

After implementing a fractional query, such that $\|H\| \leq \frac{1}{2}$ is satisfied, one can use Corollary 72 to implement the logarithm of the unitary. Also note the gap promise on the spectrum of U is necessary for implementing the fractional queries, but it is not important that the gap is exactly at $e^{i\pi}$, one can just add a phase gate for example to U in order to rotate the spectrum.

Finally, we briefly describe how these techniques can be used for Gibbs sampling. If one first prepares a maximally entangled state on two registers and applies the map $e^{-\frac{\beta}{2}(H+I)}$ on the first register, then one gets a subnormalized Gibbs state $e^{\beta(H+I)}$ on the first register. Then, using (fixed-point) amplitude amplification one gets a purification of a Gibbs-state. Each of these steps can be compactly performed using singular value transformation techniques, providing an efficient implementation.

An ε-approximation of the map $e^{-\frac{\beta}{2}(H+I)}$ can be implemented using Theorem 56 and Corollary 64 with query complexity $\mathcal{O}\big(\sqrt{\beta}\log(1/\epsilon)\big)$, and it suffices to use $\mathcal{O}\Big(\sqrt{\frac{n}{\mathcal{Z}}}\Big)$ amplitude amplification steps in order to prepare the Gibbs-state with constant success probability, where n is dimension of H and $\mathcal{Z} := \mathrm{Tr}\big[e^{-\beta(H+I)}\big]$ is the partition function. In the case when H does not have an eigenvalue close to -1, but say $\lambda_{\min}$ is the smallest eigenvalue, then one should implement an approximation of $e^{-\beta(H-\lambda_{\min}I)}$ on the domain $[\lambda_{\min}, 1]$ in order to avoid unnecessary subnormalization. However note, that this in general increases to complexity and gives a linear dependence on β. For more details see, e.g., the work of Appeldoorn et al. [AGGW17, AG18].

In the special case when one has access to the square root of H, and H has an eigenvalue close to 0, then one can still achieve quadratically improved scaling with β as shown by Chowdhury and Somma [CS17]. This can be easily shown using our techniques observing that $e^{-\beta H} = e^{-\beta\left(\sqrt{H}\right)^2}$, and that the function $e^{-\beta x^2}$ can be ε-approximated on the interval $[0,1]$ using a polynomial of degree $\mathcal{O}\big(\sqrt{\beta}\log\big(\frac{1}{\varepsilon}\big)\big)$ as follows from Theorem 63 or Corollary 64.

6 Limitations of the smooth function techniques

In the classical literature there are many good techniques for lower bounding the degrees of approximation polynomials [SV14]. There is a intimate relationship between the degrees of approximation polynomials quantum query complexity [BBC+01]. In a recent result Arunachalam et al. [ABP17] showed that for discrete problems certain polynomial approximations characterize the quantum query complexity. There are also some result about lower bounds for continuous problems [Aar09, Bel15, GAW17], however the literature to this end is much more sparse.

To advance the knowledge on lower bounds in the continuous regime, we prove a conceptually simple lower bound on eigenvalue transformations, which guides our intuition about what sort of transformations are possible. Intuitively speaking if a function has derivative d on the domain of interest then we need to use the block-encoding $\Omega(d)$-times in order to implement the eigenvalue transformation corresponding to f. This suggests that Theorem 68 applied together with Theorem 56 often gives optimal results, since the δ parameter usually turns out to be $\propto \frac{1}{d}$, where d is the maximal derivative of the function on the domain of interest.

Theorem 73 (Lower bound for eigenvalue transformation)**.** *Let $I \subseteq [-1,1]$, $a \geq 1$ and suppose U is a $(1,a,0)$-block-encoding of an unknown Hermitian matrix H with the only promise that the spectrum of H lies in I. Let $f: I \to \mathbb{R}$, and suppose that we have a quantum circuit V that implements a $(1,b,\varepsilon)$-block-encoding of $f(H)$ using T applications of U, for all U fulfilling the promise. Then for all $x \neq y \in I \cap [-\frac{1}{2},\frac{1}{2}]$ we have that*

$$T = \Omega\left(\frac{|f(x)-f(y)|-2\varepsilon}{|x-y|}\right).$$

More precisely for all $x, y \in I$ we have that

$$T \geq \frac{\max\left[f(x)-f(y)-2\varepsilon, \sqrt{1-(f(y)-\varepsilon)^2}-\sqrt{1-(f(x)+\varepsilon)^2}\right]}{\sqrt{2}\sqrt{1-xy-\sqrt{(1-x^2)(1-y^2)}}} \tag{68}$$

$$\geq \frac{\max\left[f(x)-f(y)-2\varepsilon, \sqrt{1-(f(y)-\varepsilon)^2}-\sqrt{1-(f(x)+\varepsilon)^2}\right]}{\sqrt{2}\max\left[|x-y|, \left|\sqrt{1-x^2}-\sqrt{1-y^2}\right|\right]}. \tag{69}$$

Proof. First let us examine the case when H is a $d \times d$ matrix, $a=1$ and U is of size $2d \times 2d$. Recall that in (13) we defined the two-dimensional reflection operator

$$R(x) = \begin{bmatrix} x & \sqrt{1-x^2} \\ \sqrt{1-x^2} & -x \end{bmatrix},$$

and note, that for all $x, y \in [0,1]$ we have that

$$\|R(x)-R(y)\| = \sqrt{2}\sqrt{1-xy-\sqrt{(1-x^2)(1-y^2)}} \leq \sqrt{2}\max\left[|x-y|, \left|\sqrt{1-x^2}-\sqrt{1-y^2}\right|\right]. \tag{70}$$

For all $z \in [0,1]$ let $U_z := \bigoplus_{i=1}^d R(z)$, where the direct sum structure is arranged in such a way that U_z is a $(1,1,0)$-block-encoding of $H_z := zI$. Let $V[U_z]$ denote the circuit V when using the input unitary U_z. Since $V[U_z]$ uses U_z a total number of T times we have that

$$\|V[U_x]-V[U_y]\| \leq T\|U_x-U_y\| = T\|R(x)-R(y)\|. \tag{71}$$

By the promise on V we get that $V[U_z]$ is a $(1,b,\varepsilon)$-block-encoding of $f(H_z)=f(z)I$. Let $\varsigma_{\max/\min}$ denote the maximal/minimal singular value of a matrix. Using this notation we get that

$$\varsigma_{\max}\left[(\langle 0|^{\otimes b}\otimes I)V[U_y](|0\rangle^{\otimes b}\otimes I)\right] \leq f(y)+\varepsilon, \tag{72}$$

$$\varsigma_{\min}\left[(\langle 0|^{\otimes b}\otimes I)V[U_x](|0\rangle^{\otimes b}\otimes I)\right] \geq f(x)-\varepsilon. \tag{73}$$

Let use introduce the notation $\Pi_{\overline{|0\rangle\langle 0|}} := \big((I_b - |0\rangle\langle 0|^{\otimes b}) \otimes I\big)$, then by (72)-(73) we have that

$$\begin{aligned}
\|V[U_x]-V[U_y]\| &\geq \left\|(|0\rangle\langle 0|^{\otimes b}\otimes I)V[U_x](|0\rangle\langle 0|^{\otimes b}\otimes I)-(|0\rangle\langle 0|^{\otimes b}\otimes I)V[U_y](|0\rangle\langle 0|^{\otimes b}\otimes I)\right\|\\
&\geq \varsigma_{\min}\Big[(\langle 0|^{\otimes b}\otimes I)V[U_x](|0\rangle^{\otimes b}\otimes I)\Big]-\varsigma_{\max}\Big[(\langle 0|^{\otimes b}\otimes I)V[U_y](|0\rangle^{\otimes b}\otimes I)\Big]\\
&\geq f(x)-f(y)-2\varepsilon,\ \text{ and} \qquad\qquad (74)\\
\|V[U_y]-V[U_x]\| &\geq \left\|\Pi_{\overline{|0\rangle\langle 0|}}V[U_y](|0\rangle\langle 0|^{\otimes b}\otimes I)-\Pi_{\overline{|0\rangle\langle 0|}}V[U_x](|0\rangle\langle 0|^{\otimes b}\otimes I)\right\|\\
&\geq \varsigma_{\min}\Big[\Pi_{\overline{|0\rangle\langle 0|}}V[U_y](|0\rangle^{\otimes b}\otimes I)\Big]-\varsigma_{\max}\Big[\Pi_{\overline{|0\rangle\langle 0|}}V[U_x](|0\rangle^{\otimes b}\otimes I)\Big]\\
&= \sqrt{1-\varsigma_{\max}^2\Big[(\langle 0|^{\otimes b}\otimes I)V[U_y](|0\rangle^{\otimes b}\otimes I)\Big]}-\sqrt{1-\varsigma_{\min}^2\Big[(\langle 0|^{\otimes b}\otimes I)V[U_x](|0\rangle^{\otimes b}\otimes I)\Big]}\\
&\geq \sqrt{1-(f(y)-\varepsilon)^2}-\sqrt{1-(f(x)+\varepsilon)^2}. \qquad\qquad (75)
\end{aligned}$$

By combining (71) and (74)-(75) we get that

$$T \geq \frac{\max\Big[f(x)-f(y)-2\varepsilon,\sqrt{1-(f(y)-\varepsilon)^2}-\sqrt{1-(f(x)+\varepsilon)^2}\Big]}{\|R(x)-R(y)\|}.$$

Combining this inequality with (70) proves (68)-(69). Finally note, that if $a>1$ essentially the same argument can be used to prove (68)-(69), just one needs to define U_z with additional tensor products of identity matrices acting on the new ancilla qubits. □

The above lower bound suggests that the spectrum of H lying closea to 1 is more flexible than the spectrum lying below say $\frac{1}{2}$ in absolute value. Indeed it turns out that the spectrum of H lying close to 1 is quadratically more useful than the spectrum $I \subseteq [-\frac{1}{2},\frac{1}{2}]$, cf. Corollary 64 and Lemma 22. This lower bound also explains why is it so difficult to amplify the spectrum close to 1, cf. Theorem 30. Finally note, that since eigenvalue transformation is a special case of singular value transformation it also gives a lower bound in singular value transformation.

7 Conclusion

Our main contribution in this paper is to provide a paradigm that unifies a host of quantum algorithms ranging from singular value estimation, linear equation solving, quantum simulation to quantum walks. Prior to our contribution each of these fields would have to be understood independently, which makes mastering all of them a challenge. By presenting them all within the framework of quantum singular value transformation, many of the most popular techniques in these fields follow as a direct consequence. This greatly simplifies the learning process while also revealing algorithms that were hitherto unknown.

The main result of this paper is an efficient method for implement singular value transformation, extending earlier qubitization techniques. The paper describes several novel applications to this general result, including an algorithm for performing certain "non-commutative" measurements, an exponentially improved algorithm for simulating fractional queries to an unknown unitary oracle, and an improved algorithm for principal component regression.

We also give a novel view on quantum matrix arithmetics by summarizing known results about block-encoded matrices, showing that they enable performing matrix arithmetic on quantum computers in a simple and efficient manner. The described method in principle can give exponential savings in terms of the dimension of the matrices, and perfectly fits into our framework.

An interesting question for future work involves the recent work by Catalin Dohotaru and Peter Høyer which shows that a wide range of quantum walk algorithms can be unified within a single paradigm called controlled quantum amplification [DH17]. While the structure of the quantum circuits introduced by them bears a strong resemblance to those used in qubitization, it is difficult to place this work within the framework we present here. The question of how to unify their approach with our techniques therefore remains open.

Acknowledgments

A.G. thanks Ronald de Wolf, Robin Kothari, Joran van Apeldoorn, Shantanav Chakraborty, Stacey Jeffrey, Vedran Dunjko and Yimin Ge for inspiring discussions. Y.S. was supported in part by the Army Research Office (MURI award W911NF-16-1-0349); the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Quantum Algorithms Teams program; and the National Science Foundation (grant 1526380). He thanks Andrew Childs, Guoming Wang, Cedric Lin, John Watrous, Ben Reichardt, Guojing Tian and Aaron Ostrander for helpful discussions.

References

[Aar06] Scott Aaronson. The ten most annoying questions in quantum computing, 2006. https://www.scottaaronson.com/blog/?p=112.

[Aar09] Scott Aaronson. Quantum copy-protection and quantum money. In *Proceedings of the 24th IEEE Conference on Computational Complexity (CCC)*, pages 229–242, 2009. arXiv: 1110.5353

[AAR$^+$18] Mohammad H. Amin, Evgeny Andriyash, Jason Rolfe, Bohdan Kulchytskyy, and Roger Melko. Quantum Boltzmann machine. *Physical Review X*, 8(2):021050, 2018. arXiv: 1601.02036

[ABP17] Srinivasan Arunachalam, Jop Briët, and Carlos Palazuelos. Quantum query algorithms are completely bounded forms. arXiv: 1711.07285, 2017.

[AC12] Scott Aaronson and Paul Christiano. Quantum money from hidden subspaces. In *Proceedings of the 44th ACM Symposium on Theory of Computing (STOC)*, pages 41–60. ACM, 2012. arXiv: 1203.4740

[AG18] Joran van Apeldoorn and András Gilyén. Improvements in quantum SDP-solving with applications. arXiv: 1804.05058, 2018.

[AGGW17] Joran van Apeldoorn, András Gilyén, Sander Gribling, and Ronald de Wolf. Quantum SDP-solvers: Better upper and lower bounds. In *Proceedings of the 58th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 403–414, 2017. arXiv: 1705.01843

[AKS12] Andris Ambainis, Julia Kempe, and Or Sattath. A quantum Lovász local lemma. *Journal of the ACM*, 59(5):24, 2012. arXiv: 0911.1696

[Amb12] Andris Ambainis. Variable time amplitude amplification and quantum algorithms for linear algebra problems. In *Symposium on Theoretical Aspects of Computer Science STACS*, pages 636–647, 2012. arXiv: 1010.4458

[AP16] Alexei B. Aleksandrov and Vladimir V. Peller. Operator Lipschitz functions. *Russian Mathematical Surveys*, 71(4):605–702, 2016. arXiv: 1611.01593

[AS74] Milton Abramowitz and Irene A. Stegun. *Handbook of Mathematical Functions, with Formulas, Graphs, and Mathematical Tables*. Dover Publications Inc., New York, NY, USA, 1974.

[BACS07] Dominic W. Berry, Graeme Ahokas, Richard Cleve, and Barry C. Sanders. Efficient quantum algorithms for simulating sparse Hamiltonians. *Communications in Mathematical Physics*, 270(2):359–371, 2007. arXiv: quant-ph/0508139

[BBC+01] Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald de Wolf. Quantum lower bounds by polynomials. *Journal of the ACM*, 48(4):778–797, 2001. Earlier version in FOCS'98. arXiv: quant-ph/9802049

[BCC+15] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma. Simulating hamiltonian dynamics with a truncated taylor series. *Physical Review Letters*, 114(9):090502, 2015. arXiv: 1412.4687

[BCK15] Dominic W. Berry, Andrew M. Childs, and Robin Kothari. Hamiltonian simulation with nearly optimal dependence on all parameters. In *Proceedings of the 56th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 792–809, 2015. arXiv: 1501.01715

[Bel15] Aleksandrs Belovs. Variations on quantum adversary. arXiv: 1504.06943, 2015.

[BHMT02] Gilles Brassard, Peter Høyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. In *Quantum Computation and Quantum Information: A Millennium Volume*, volume 305 of *Contemporary Mathematics Series*, pages 53–74. AMS, 2002. arXiv: quant-ph/0005055

[BKL+17] Fernando G. S. L. Brandão, Amir Kalev, Tongyang Li, Cedric Yen-Yu Lin, Krysta M. Svore, and Xiaodi Wu. Quantum sdp solvers: Large speed-ups, optimality, and applications to quantum learning. arXiv: 1710.02581, 2017.

[BS17] Fernando G. S. L. Brandão and Krysta M. Svore. Quantum speed-ups for solving semidefinite programs. In *Proceedings of the 58th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 415–426, 2017. arXiv: 1609.05537

[CEMM98] Richard Cleve, Artur Ekert, Chiara Macchiavello, and Michele Mosca. Quantum algorithms revisited. *Proceedings of the Royal Society A*, 454(1969):339–354, 1998. arXiv: quant-ph/9708016

[CGJ18] Shantanav Chakraborty, András Gilyén, and Stacey Jeffery. The power of block-encoded matrix powers: improved regression techniques via faster Hamiltonian simulation. arXiv: 1804.01973, 2018.

[CJKM13] Andrew M. Childs, Stacey Jeffery, Robin Kothari, and Frédéric Magniez. A time-efficient quantum walk for 3-distinctness using nested updates. arXiv: 1302.7316, 2013.

[CKS17] Andrew M. Childs, Robin Kothari, and Rolando D. Somma. Quantum algorithm for systems of linear equations with exponentially improved dependence on precision. *SIAM Journal on Computing*, 46(6):1920–1950, 2017. arXiv: 1511.02306

[CMN$^+$17] Andrew M. Childs, Dmitri Maslov, Yunseong Nam, Neil J. Ross, and Yuan Su. Toward the first quantum simulation with quantum speedup. arXiv: 1711.10980, 2017.

[CS17] Anirban Narayan Chowdhury and Rolando D. Somma. Quantum algorithms for Gibbs sampling and hitting-time estimation. *Quantum Information and Computation*, 17(1&2):41–64, 2017. arXiv: 1603.02940

[CW12] Andrew M. Childs and Nathan Wiebe. Hamiltonian simulation using linear combinations of unitary operations. *Quantum Information and Computation*, 12(11&12):901–924, 2012. arXiv: 1202.5822

[DH17] Cătălin Dohotaru and Peter Høyer. Controlled quantum amplification. In *Proceedings of the 44th International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 80, pages 18:1–18:13, 2017.

[Dol46] C. L. Dolph. A current distribution for broadside arrays which optimizes the relationship between beam width and side-lobe level. *Proceedings of the IRE*, 34(6):335–348, 1946.

[EY07] Alexandre Eremenko and Peter Yuditskii. Uniform approximation of sgn(x) by polynomials and entire functions. *Journal d'Analyse Mathématique*, 101(1):313–324, 2007. arXiv: math/0604324

[EY11] Alexandre Eremenko and Peter Yuditskii. Polynomials of the best uniform approximation to sgn(x) on two intervals. *Journal d'Analyse Mathématique*, 114(1):285, 2011. arXiv: 1008.3765

[Fey82] Richard P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6-7):467–488, 1982.

[FMMS16] Roy Frostig, Cameron Musco, Christopher Musco, and Aaron Sidford. Principal component projection without principal component analysis. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pages 2349–2357, 2016. arXiv: 1602.06872

[FN09] Yuliya B. Farforovskaya and Ludmila N. Nikolskaya. Modulus of continuity of operator functions. *St. Petersburg Math. J. – Algebra i Analiz*, 20(3):493–506, 2009.

[GAW17] András P. Gilyén, Srinivasan Arunachalam, and Nathan Wiebe. Optimizing quantum optimization algorithms via faster quantum gradient computation. arXiv: 1711.00465, 2017.

[Gil14] András P. Gilyén. Quantum walk based search methods and algorithmic applications. Master's thesis, Eötvös Loránd University, 2014.

[Gro96] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the 28th ACM Symposium on Theory of Computing (STOC)*, pages 212–219, 1996. arXiv: quant-ph/9605043

[Gro05] Lov K. Grover. Fixed-point quantum search. *Physical Review Letters*, 95(15):150501, 2005. arXiv: quant-ph/0503205

[GS17] András P. Gilyén and Or Sattath. On preparing ground states of gapped Hamiltonians: An efficient quantum Lovász local lemma. In *Proceedings of the 58th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 439–450, 2017. arXiv: 1611.08571

[HHL09] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical Review Letters*, 103(15):150502, 2009. arXiv: 0811.3171

[HLM17] Aram W. Harrow, Cedric Yen-Yu Lin, and Ashley Montanaro. Sequential measurements, disturbance and property testing. In *Proceedings of the 28th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1598–1611, 2017. arXiv: 1607.03236

[HLZ$^+$17] Yong He, Ming-Xing Luo, E. Zhang, Hong-Ke Wang, and Xiao-Feng Wang. Decompositions of n-qubit Toffoli gates with linear circuit complexity. *International Journal of Theoretical Physics*, 56(7):2350–2361, 2017.

[Hø00] Peter Høyer. Arbitrary phases in quantum amplitude amplification. *Physical Review A*, 62(5):052304, 2000. arXiv: quant-ph/0006031

[Jor75] Camille Jordan. Essai sur la géométrie à n dimensions. *Bulletin de la Société Mathématique de France*, 3:103–174, 1875.

[KL18] Iordanis Kerenidis and Alessandro Luongo. Quantum classification of the MNIST dataset via slow feature analysis. arXiv: 1805.08837, 2018.

[KLL$^+$17] Shelby Kimmel, Cedric Yen-Yu Lin, Guang Hao Low, Maris Ozols, and Theodore J. Yoder. Hamiltonian simulation with optimal sample complexity. *npj Quantum Information*, 3(1):13, 2017. arXiv: 1608.00281

[KMOR16] Hari Krovi, Frédéric Magniez, Maris Ozols, and Jérémie Roland. Quantum walks can find a marked element on any graph. *Algorithmica*, 74(2):851–907, 2016. arXiv: 1002.2419

[KP17a] Iordanis Kerenidis and Anupam Prakash. Quantum gradient descent for linear systems and least squares. arXiv: 1704.04992, 2017.

[KP17b] Iordanis Kerenidis and Anupam Prakash. Quantum recommendation systems. In *Proceedings of the 8th Innovations in Theoretical Computer Science Conference (ITCS)*, pages 49:1–49:21, 2017. arXiv: 1603.08675

[KW17] Mária Kieferová and Nathan Wiebe. Tomography and generative training with quantum Boltzmann machines. *Physical Review A*, 96(6):062327, 2017. arXiv: 1612.05204

[LC16] Guang Hao Low and Isaac L. Chuang. Hamiltonian simulation by qubitization. arXiv: 1610.06546, 2016.

[LC17a] Guang Hao Low and Isaac L. Chuang. Hamiltonian simulation by uniform spectral amplification. arXiv: 1707.05391, 2017.

[LC17b] Guang Hao Low and Isaac L. Chuang. Optimal Hamiltonian simulation by quantum signal processing. *Physical Review Letters*, 118(1):010501, 2017. arXiv: 1606.02685

[Llo96] Seth Lloyd. Universal quantum simulators. *Science*, 273(5278):1073–1078, 1996.

[LMR14] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum principal component analysis. *Nature Physics*, 10:631–633, 2014. arXiv: 1307.0401

[LYC16] Guang Hao Low, Theodore J. Yoder, and Isaac L. Chuang. Methodology of resonant equiangular composite quantum gates. *Physical Review X*, 6(4):041067, 2016. arXiv: 1603.03996

[MNRS11] Frédéric Magniez, Ashwin Nayak, Jérémie Roland, and Miklos Santha. Search via quantum walk. *SIAM Journal on Computing*, 40(1):142–164, 2011. arXiv: quant-ph/0608026

[MW05] Chris Marriott and John Watrous. Quantum Arthur–Merlin games. *Computational Complexity*, 14(2):122–152, 2005. arXiv: cs/0506068

[NR96] C. Andrew Neff and John H. Reif. An efficient algorithm for the complex roots problem. *Journal of Complexity*, 12(2):81 – 115, 1996.

[NWZ09] Daniel Nagaj, Pawel Wocjan, and Yong Zhang. Fast amplification of qma. *Quantum Information and Computation*, 9(11&12):1053–1068, 2009. arXiv: 0904.1549

[RML14] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. Quantum support vector machine for big data classification. *Physical Review Letters*, 113(13):130503, 2014. arXiv: 1307.0471

[SA15] Or Sattath and Itai Arad. A constructive quantum Lovász local lemma for commuting projectors. *Quantum Information and Computation*, 15(11&12):987–996, 2015. arXiv: 1310.7766

[SCV13] Martin Schwarz, Toby S. Cubitt, and Frank Verstraete. Quantum information-theoretic proof of the commutative quantum Lovász local lemma. arXiv: 1311.6474, 2013.

[Sho97] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997. Earlier version in FOCS'94. arXiv: quant-ph/9508027

[SMM09] Lana Sheridan, Dmitri Maslov, and Michele Mosca. Approximating fractional time quantum evolution. *Journal of Physics A: Mathematical and Theoretical*, 42(18):185302, 2009. arXiv: 0810.3843

[SV14] Sushant Sachdeva and Nisheeth K. Vishnoi. Faster algorithms via approximation theory. *Found. Trends Theor. Comput. Sci.*, 9(2):125–210, 2014.

[Sze04] Mario Szegedy. Quantum speed-up of Markov chain based algorithms. In *Proceedings of the 45th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 32–41, 2004. arXiv: quant-ph/0401053

[WBL12] Nathan Wiebe, Daniel Braun, and Seth Lloyd. Quantum algorithm for data fitting. *Physical Review Letters*, 109(5):050505, 2012. arXiv: 1204.5242

[WK17] Nathan Wiebe and Ram Shankar Siva Kumar. Hardening quantum machine learning against adversaries. arXiv: 1711.06652, 2017.

[YLC14] Theodore J. Yoder, Guang Hao Low, and Isaac L. Chuang. Fixed-point quantum search with an optimal number of queries. *Physical Review Letters*, 113(21):210501, 2014. arXiv: 1409.3305