Quantum Singular Value Transformation and Beyond: Exponential Improvements for Quantum Matrix Arithmetics

András Gilyén*
QuSoft, CWI and University of Amsterdam
The Netherlands
gilyen@cwi.nl

Yuan Su
Department of Computer Science, Institute for Advanced Computer Studies, and Joint Center for Quantum Information and Computer Science, U. of Maryland
College Park, MD, USA
buptsuyuan@gmail.com

Guang Hao Low
Nathan Wiebe
Quantum Architectures and Computing group, Microsoft Research
Redmond, WA, USA
GuangHao.Low@microsoft.com
nawiebe@microsoft.com

ABSTRACT

An n-qubit quantum circuit performs a unitary operation on an exponentially large, 2^n-dimensional, Hilbert space, which is a major source of quantum speed-ups. We develop a new "Quantum singular value transformation" algorithm that can directly harness the advantages of exponential dimensionality by applying polynomial transformations to the singular values of a block of a unitary operator. The transformations are realized by quantum circuits with a very simple structure – typically using only a constant number of ancilla qubits – leading to optimal algorithms with appealing constant factors.

We show that our framework allows describing many quantum algorithms on a high level, and enables remarkably concise proofs for many prominent quantum algorithms, ranging from optimal Hamiltonian simulation to various quantum machine learning applications. We also devise a new singular vector transformation algorithm, describe how to exponentially improve the complexity of implementing fractional queries to unitaries with a gapped spectrum, and show how to efficiently implement principal component regression. Finally, we also prove a quantum lower bound on spectral transformations.

CCS CONCEPTS

• **Theory of computation → Quantum computation theory**; *Algorithm design techniques.*

KEYWORDS

block-encdoing, qubitization, quantum signal processing

ACM Reference Format:
András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. 2019. Quantum Singular Value Transformation and Beyond: Exponential Improvements for Quantum Matrix Arithmetics. In *Proceedings of the 51st Annual ACM SIGACT Symposium on the Theory of Computing (STOC '19), June 23–26, 2019, Phoenix, AZ, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3313276.3316366

*Supported by ERC Consolidator Grant 615307-QPROGRESS and partially supported by QuantERA project QuantAlgo 680-91-034.

STOC '19, June 23–26, 2019, Phoenix, AZ, USA

ACM ISBN 978-1-4503-6705-9/19/06…$15.00
https://doi.org/10.1145/3313276.3316366

1 INTRODUCTION

It is often said in quantum computing that there are only a few quantum algorithms [52] known to give speed-ups over classical computers. Indeed, a remarkable number of applications stem from a small set of standard quantum algorithmic primitives. The first class of quantum speedups is derived from quantum simulation, originally suggested by Feynman [22]. Such algorithms yield exponential speedups over the best known classical methods for simulating quantum dynamics and are applied to solve electronic structure problems in material science and chemistry. The two most influential quantum algorithms developed later in the '90s are Shor's algorithm [51] (based on the quantum Fourier transform) and Grover's search algorithm [27]. Other examples have emerged over the years. However, arguably the quantum walk algorithm of Ambainis [3] and Szegedy [53] together with the quantum linear systems algorithm of Harrow, Hassidim and Lloyd [30] are the most important other primitives that provide speed-ups relative to classical computing. An important question that remains is "are these primitives truly independent or can they be seen as examples of a deeper underlying concept?" The aim of this paper 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 we call "quantum singular value transformation", which generalizes all the above-mentioned techniques except for the quantum Fourier transform (and thus Shor's algorithm).

Of the aforementioned quantum algorithms, quantum simulation is arguably the most diverse and rapidly developing. Within the last few years, a host of new techniques have led to ever more powerful methods [18]. The problem of quantum simulation is to take an efficient description of a Hamiltonian H, an evolution time t, and an error tolerance ε, and implement a quantum operation V such that $\left\|e^{-iHt} - V\right\| \leq \varepsilon$ using as few resources as possible. The first methods introduced to solve this problem were Trotter formula decompositions [9, 39] and subsequently methods based on linear combinations of unitaries were developed [19] to provide better asymptotic scaling of the cost of simulation.

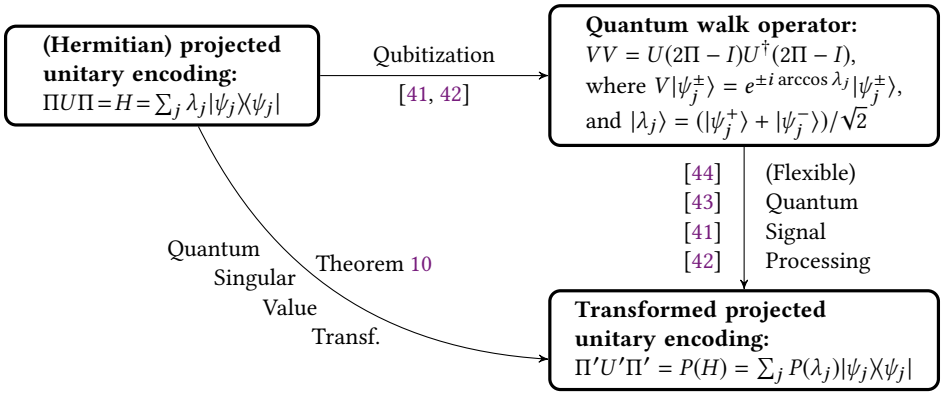


Figure 1: Schematic comparison of QSVT with previous approaches. Remark: QSVT also works for non-Hermitian H.

An alternative strategy developed concurrently used ideas from quantum walks. Asymptotically, this approach is perhaps the favored method for simulating time-independent Hamiltonians as it achieves near-optimal scaling in all relevant parameters. The main tool in this approach is a walk operator that has eigenvalues $\pm e^{\pm i \arcsin(E_k/\alpha)}$, where E_k is the k^{th} eigenvalue of H and α is a normalizing parameter. While early works used phase estimation to invert the arcsin in the spectrum in order to recover the desired eigenvalues e^{-iE_k}, subsequent works achieved better scaling by using a linear combination of quantum walk steps [12]. Recently another approach, called *qubitization* [41], was introduced to create this walk operator for more general inputs, and was combined with the technique of *quantum signal processing* [42, 44] for more efficient spectral transformations.

Quantum simulation is not the only application that benefits from such spectral transformations. Quantum linear systems algorithms [30] as well as algorithms for semi-definite programming [6, 14] use these ideas extensively. Earlier work on linear systems used phase estimation to estimate the eigenvalues of a matrix λ_j and then applied 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 pseudoinverse are straightforward. More recent methods eschew the use of phase estimation in favor of linear combination of unitary methods [17] which typically approximate the inversion using a Fourier series or Chebyshev series expansion. Similar ideas can be used to prepare Gibbs states efficiently [6, 20].

These improvements typically result in exponentially improved scaling in terms of precision. 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 often unavoidable. Therefore these improvements typically "only" result in polynomial savings in the overall 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, where the scaling with accuracy was improved from the initial $O(1/\epsilon^{32})$ [14] to $O(1/\epsilon^4)$ [4].

1.1 A Unifying Perspective

We develop a new technique that we call *quantum singular value transformation*, which unifies and generalizes qubitization and quantum signal processing. The central object for this result is a *projected unitary encoding* of a matrix A of interest. 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 $A := \widetilde{\Pi} U \Pi$. The encoding is called *symmetric* if $\widetilde{\Pi} = \Pi$.

Roughly speaking, qubitization turns a symmetric projected unitary encoding $\Pi U \Pi = H$ of a Hermitian operator H into a unitary V which is the "square-root" of a (Szegedy-type) quantum walk operator $U(2\Pi - I)U^\dagger(2\Pi - I)$, so that each eigenvector $|\psi\rangle$ of H with eigenvalue λ becomes a superposition of two[1] eigenvectors $|\psi^\pm\rangle$ of V with eigenvalues $e^{\pm i \arccos \lambda}$. If U happens to be a Hermitian unitary (i.e., a *reflection operator*), then one can use $V := U(2\Pi - I)$; otherwise one can replace U by a suitable Hermitian unitary $\widetilde{U}$ using a controlled-U and controlled-$U^\dagger$ gate. Subsequently, quantum signal processing transforms the spectrum of the unitary V, resulting in a new circuit U', which is a projected unitary encoding of a transformed operator $P(H)$, see Figure 1.

Our results generalize the above by working with arbitrary, not necessarily Hermitian, projected matrices A without requiring symmetric encodings (i.e., $\widetilde{\Pi} \neq \Pi$ is allowed). Moreover, the unitary U does not need to be Hermitian, and if P is an even or odd polynomial, then we do not even require access to controlled versions of U or $U^\dagger$. 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

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

We also prove a similar result in the even case, replacing $\widetilde{\Pi}$ by Π, and defining $P^{(SV)}(A) := VP(\Sigma)V^\dagger$ for even polynomials.

[1] This justifies the name qubitization: each eigenvector of H splits into two eigenvectors of V, and so the resulting two-dimensional subspace is isomorphic to a qubit.

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 complete or representative, the table only intends to give some intuition and to illustrate the different sources of speed-ups.

Speed-up	Source of speed-up	Examples of algorithms
Exponential	Dimensionality of the Hilbert space	Hamiltonian simulation [39]
	Precise polynomial approximations	Improved HHL algorithm [17]
Quadratic	Singular value = square root of probability	Grover search [27]
	Distinguishability of singular values	Amplitude estimation [15]
	Singular values close to 1 are more useful	Quantum walks [53]

1.2 Direct Algorithmic Applications

We show that many prominent quantum algorithms can be viewed as instantiations of our quantum singular value transformation meta-algorithm with an appropriately chosen polynomial. In order to illustrate the power of this technique we briefly derive some corollaries, and show natural examples of projected unitary encodings.

For example, suppose that U is a quantum algorithm that, starting from the initial state $|0\rangle^{\otimes n}$, prepares a desired state with success probability at least p and indicates success by setting the first qubit to $|1\rangle$. Thus we have a unitary

$$U : |0\rangle^{\otimes n} \mapsto \sqrt{p}|1\rangle|\psi_G\rangle + \sqrt{1-p}|0\rangle|\psi_B\rangle,$$

and the goal is to prepare the state $|\psi_G\rangle$. Since ordinary amplitude amplification may "overamplify" we need multiple repetitions in order to succeed, and if we want to perform amplification coherently it creates a large garbage state attached to $|\psi_G\rangle$. *Fixed-point* amplitude amplification [28, 54] provides a way to directly perform the $|0\rangle^{\otimes n} \mapsto |\psi_G\rangle$ mapping without using many ancilla qubits. Let us take $\widetilde{\Pi} := |1\rangle\langle 1| \otimes I_{n-1}$ and $\Pi := \left(|0\rangle\langle 0|\right)^{\otimes n}$. Observe that $A = \widetilde{\Pi}U\Pi$ is a rank-1 matrix having a single non-trivial singular value which is the square root of the success probability. If P is an odd polynomial bounded by 1 in absolute value which 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 $O\left(\log(1/\varepsilon)/\sqrt{p}\right)$, providing an efficient and conceptually simple implementation of fixed-point amplitude amplification, preparing the desired state in one go.

We develop some *new* algorithms as well, including a *singular vector transformation* algorithm, which maps right singular vectors to left singular vectors in a single-shot manner. This is a common generalization and extension of fixed-point [54] and oblivious [10] amplitude amplification.

Theorem 1 (Singular vector transformation). *Call an application of the unitary U, its inverse and the controlled reflection operators $(2\Pi - I)$, $(2\widetilde{\Pi} - I)$ a "query". If $\widetilde{\Pi}U\Pi = \sum_{i=1}^{k} \varsigma_i |\phi_i\rangle\langle\psi_i|$ is an SVD, then we can transform an arbitrary input state*

$$|\psi\rangle = \textstyle\sum_{i=i}^{k} \alpha_i |\psi_i\rangle \quad \text{to} \quad |\phi\rangle = \sum_{i=i}^{k} \alpha_i |\phi_i\rangle,$$

with precision ε, in query and gate complexity $O\left(\frac{1}{\delta}\log\left(\frac{1}{\varepsilon}\right)\right)$, under the assumption that $\varsigma_i \geq \delta$ for all $\alpha_i \neq 0$.

This algorithm also gives an efficient solution to a form of "non-commutative measurement" problem used for efficient ground-state preparation of certain local Hamiltonians [25], and quadratically improves [24] the gap dependence of earlier approaches. It can also be used for devising a new method for singular value estimation [16, 26, 37].

As discussed in Section 4, other algorithms can also be easily cast in the singular value transformation framework, including robust oblivious amplitude amplification, fast quantum OR lemma, fast QMA amplification, quantum matrix inversion, and related problems. The various types of quantum speed-ups that are incorporated in our singular value transformation framework are summarized in Table 1.

In our framework, designing quantum algorithms mostly boils down to finding low-degree polynomial approximations of various functions. Therefore we also develop some general tools for finding such approximations. These results indicate that for smooth functions the asymptotic precision dependence is typically logarithmic.

1.3 Block-encoding & Hermitian Matrices

In order to use quantum singular value transformation, one needs to construct projected unitary encodings. A special case of projected unitary encoding is called *block-encoding*, where $\widetilde{\Pi} = \Pi = \left(|0\rangle\langle 0|\right)^{\otimes a} \otimes I$. In this case A is literally[2] the top-left block of the unitary U:

$$A = \left(\langle 0|^{\otimes a} \otimes I\right) U \left(|0\rangle^{\otimes a} \otimes I\right) \Longleftrightarrow U = \begin{bmatrix} A & \cdot \\ \cdot & \cdot \end{bmatrix}. \tag{1}$$

We call such a unitary U an a-qubit block-encoding of A. One can think of U as a probabilistic implementation of A: given an input state $|\psi\rangle$, applying the unitary U to the state $|0\rangle^{\otimes a}|\psi\rangle$, measuring the first a-qubit register, and post-selecting on the $|0\rangle^{\otimes a}$ outcome, we get a state $\propto A|\psi\rangle$ in the second register.

In Section 5, we provide a versatile toolbox for efficiently constructing block-encodings, and summarize recent developments in the field. In particular we describe how to construct block-encodings of unitary matrices, density operators, POVM operators, sparse-access matrices, and matrices stored in a QROM[3]. Furthermore, we show how to form linear combinations and products of block-encodings, enabling the efficient implementation of quantum matrix arithmetic. Quantum matrix arithmetics carries out all calculations in an operational way, meaning that the matrices are represented

[2] In block-encodings we omit the dimensions projected out by $\left(|0\rangle\langle 0|\right)^{\otimes a}$ for convenience, i.e., A and U have different sizes, unlike in projected encodings.

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

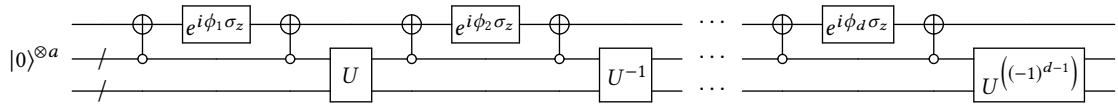


Figure 2: Circuits used for quantum singular value transformation. The empty dots denote control by the $|0\rangle^{\otimes a}$ state, so that the corresponding gate is an "inverted" multi-qubit Toffoli gate, where each qubit is conjugated by an X gate compared to the usual Toffoli. The other gates are single qubit rotations or applications of $U^{\pm 1}$. NB the crossed wires $\not+$ denote multiple qubits.

by block-encodings, in principle enabling exponential speed-ups in terms of the dimension of the matrices.

For block-encodings, the circuits in quantum singular value transformation become especially nice. Our main theorem applied to block-encodings gives the following result:

Theorem 2 (Special case of Corollary 11)**.** *Let $P\colon [-1,1]\mapsto[-1,1]$ be a degree-d even/odd polynomial map. Suppose that U is a block-encoding of A that has SVD: $A=\sum_i \varsigma_i|w_i\rangle\langle v_i|$. There is a $\Phi\in\mathbb{R}^d$ so that U_Φ, the circuit depicted on Figure 2, is such that $(H\otimes I)U_\Phi(H\otimes I)$ is a block-encoding of*

$$\sum_i P(\varsigma_i)|w_i\rangle\langle v_i| \quad \text{if } d \text{ is odd, and}$$
$$\sum_i P(\varsigma_i)|v_i\rangle\langle v_i| \quad \text{if } d \text{ is even,}$$

where H stands for the single qubit Hadamard gate. Moreover, given the coefficients of P, one can (classically) efficiently compute approximate versions of Φ.

It turns out that several earlier quantum algorithms also implicitly use block-encodings. For example, one can show that the update operator of a quantum walk corresponding to a reversible Markov chain M [45] essentially implements a block-encoding of the "discriminant matrix" $D(M)$ of the Markov chain. Also, k steps of the quantum walk operator applies the Chebyshev polynomial T_{2k} to the discriminant matrix. In fact our quantum singular value transformation circuit corresponds to a Szegedy [53] quantum walk in the special case when all phases are set to $\frac{\pi}{2}$.

Using the above theorem we can quickly reprove a recent result about quantum fast-forwarding [7] of Markov chains, while substantially improving its complexity. As we discussed above, quantum walks provide block-encodings of the discriminant matrix $D(M)$. On the other hand we know [49] that the monomial x^t can be ε-approximated by a polynomial P of degree $\sqrt{2t\log(2/\varepsilon)}$ on the interval $[-1,1]$. Since the discriminant matrix is symmetric, using this polynomial P in Theorem 2 gives an ε-approximate block-encoding of $D(M)^t$. This solves quantum fast-forwarding, and in some sense, it emulates t steps of the walk using only $\propto\sqrt{t}$ quantum operations. Due to Theorem 2 the (query) complexity of this implementation is $O\left(\sqrt{t\log(1/\varepsilon)}\right)$, which is a significant improvement in terms of precision over the complexity $O\left(\sqrt{t/\varepsilon}\log\left(1/\varepsilon\right)\right)$ of the original[4] result of Apers and Sarlette [7].

In the special case when the block-encoded matrix is Hermitian, we can remove the parity constraint from Theorem 2 by combining the even and odd parts of the polynomials. This enables a simple proof of recent optimal block-Hamiltonian simulation results [43], and readily shows how to exponentially improve the complexity of implementing fractional queries to unitaries with a gapped spectrum.

[4] In the second arXiv version of their [7] paper they also recovered this improved result using an alternative method, but also cite our approach [26].

1.4 A Lower Bound

We also prove a bound on the efficiency of singular value transformation. Our lower bound suggests that the spectrum of a Hermitian block-encoded matrix H lying close to ±1 is more "flexible" than the spectrum lying below, say, $\frac{1}{2}$ in absolute value. It also gives a lower bound on singular value transformation, as Hermitian eigenvalue transformation is a special case of singular value transformation.

Theorem 3 (Proven in the full version [26])**.** *Let $I\subseteq[-1,1]$ and suppose a unitary U block-encodes 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 block-encodes $f(H)$ with accuracy ε using T applications of U or $U^\dagger$. 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, T is at least*

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

This lower bound shows that many circuits described in this paper have essentially optimal complexity. For example our pseudoinverse implementation and our singular / eigenvalue projector implementations are essentially optimal.

2 PRELIMINARIES AND NOTATION

In this paper we will work with polynomial approximations and therefore introduce some related notation. For a function $f:\mathbb{R}\to\mathbb{C}$ and a subset $I\subseteq\mathbb{R}$ 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. For an integer number $z\in\mathbb{Z}$ we say that P has parity-$(z \bmod 2)$ 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 $T_d(x):=\cos(d\arccos(x))$.

When we present a matrix and put a $\cdot$ in some place we mean a matrix with arbitrary values of the elements in the unspecified block. For example $[\cdot]$ just denotes a matrix with arbitrary elements.

We denote the a-qubit identity operator by I_a, and use the notation $[d] := \{1, 2, \ldots, d\}$.

3 QUANTUM SINGULAR VALUE TRANSFORMATION (QSVT)

3.1 Quantum Signal Processing (QSP)

Our results are based on quantum signal processing, introduced by Low, Yoder, and Chuang [44]. They asked the following question: suppose we can implement single-qubit gate sequences of the form

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

where $\sigma_z = |0\rangle\langle 0| - |1\rangle\langle 1|$ is a Pauli matrix, and

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

is a "signal unitary". Which single-qubit unitary operators can we build this way if we can choose the angles $\phi_0, \phi_1, \ldots, \phi_k$ arbitrarily, but $x \in [-1, 1]$ is unknown to us? They characterized the set of unitary operators that can be constructed this way, and showed that the set of achievable unitary operators is quite rich. We provide a self-contained treatment of quantum signal processing in the full version of our paper [26], and give a simple induction based proof of the following characterization, analogous to the results of [44].

Theorem 4. *Let $k \in \mathbb{N}$ and $\Phi = (\phi_0, \phi_1, \ldots, \phi_k) \in \mathbb{R}^{k+1}$. Then there exist $P, Q \in \mathbb{C}[x]$ such that for all $x \in [-1, 1]$*

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

and

(i) $\deg(P) \le k$, $\deg(Q) \le k-1$, *and*
(ii) P *has parity-*$(k \bmod 2)$, Q *has parity-*$(k-1 \bmod 2)$, *and*
(iii) $\forall x \in [-1, 1]$: $|P(x)|^2 + (1-x^2)|Q(x)|^2 = 1$.

Moreover, if $P, Q \in \mathbb{C}[x]$ satisfy (i)-(iii), then we can find a $\Phi \in \mathbb{R}^{k+1}$ that satisfies the above equation.

Working with both the polynomials P and Q is often unnecessary, and Low, Yoder, and Chuang [44] showed that if one is only interested in the real part of P, the formulation becomes much simpler. Instead of working with $W(x)$, for us it is beneficial to work with single-qubit reflection operators, which are defined for all $x \in [-1, 1]$ as

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

In the full version [26] we prove the following result:

Corollary 5 (Real quantum signal processing)**.** *Let $P_{\Re}(x) \in \mathbb{R}[x]$ be a degree-d polynomial for some $d \ge 1$, such that*

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

Then there is a $P \in \mathbb{C}[x]$ of degree d and a $\Phi \in \mathbb{R}^d$ such that

$$\prod_{j=1}^{d}\left(R(x)e^{i\phi_j\sigma_z}\right) = \begin{bmatrix} P(x) & \cdot \\ \cdot & \cdot \end{bmatrix}. \tag{2}$$

From an algorithmic perspective, it is also important to know how to find the angles given the real polynomial $P_{\Re}(x)$. Since our proofs are constructive, they also show how to compute the angles. A rigorous analysis of numerical errors and an optimized algorithm for finding an angle sequence corresponding to a given polynomial was recently developed by Haah [29]. His algorithm runs in time $\widetilde{O}\left(d^3\,\mathrm{polylog}(1/\varepsilon)\right)$ for a degree-d polynomial and returns the respective angles for a uniform ε-approximating polynomial over the interval $[-1, 1]$.

A special case is when $P_{\Re}(x)$ is a Chebyshev polynomial of the first kind. In this case, we exactly know what is the corresponding angle sequence:

Lemma 6 (Chebyshev polynomials in QSP)**.** *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 (2) we get that $P = T_d$.*

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

This result shows that quantum singular value transformation is a direct generalization of Szegedy quantum walks [53].

3.2 Higher Dimensional Extension

Now we leverage the quantum signal processing results to perform quantum singular value transformation of projected unitary matrices, with ideas coming from qubitization [41].

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 our approach is to lift the quantum signal processing results presented in the previous subsection by studying two-dimensional invariant subspaces coming from Camille Jordan's Lemma [34]. Then by understanding how the two-dimensional subspaces behave, one can infer the higher-dimensional behavior.

Qubitization can be understood along the lines of Jordan's Lemma about the common invariant subspaces of two reflections. Jordan's result [34] is most often presented stating that the product of two reflections decomposes into 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 [53] as well as Marriott and Watrous's QMA amplification scheme [46].

Motivated by a series of prior works on quantum search algorithms [28, 33, 54], Low and Chuang [41] replaced one of the reflections in Jordan's Lemma by a phase-gate, such as in Figure 3b. 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[5] by an arbitrary unitary operator U, and analyze the procedure with carefully chosen one- and two-dimensional subspaces coming from SVD, similar to Jordan's Lemma.

[5] One could in principle merge U and $U^\dagger$ into one of the projectors, leading to a product of reflections as in Jordan's Lemma, however we take a slightly different perspective.

Definition 7 (SVD of a projected unitary). *Let $\mathcal{H}_U$ be a finite-dimensional Hilbert space and let $U, \Pi, \widetilde{\Pi}$ be $\mathcal{H}_U \to \mathcal{H}_U$ linear operators, where U is a unitary, Π and $\widetilde{\Pi}$ are orthogonal projectors, and let $A := \widetilde{\Pi}U\Pi$. Let $d := \mathrm{rank}(\Pi)$, $\tilde{d} := \mathrm{rank}(\widetilde{\Pi})$, and $d_{\min} := \min(d, \tilde{d})$. By the SVD of A we know that there exist orthonormal bases $(|\psi_i\rangle : i \in [d])$, $(|\tilde{\psi}_i\rangle : i \in [\tilde{d}])$ of the subspaces $\mathrm{img}(\Pi)$ and $\mathrm{img}(\widetilde{\Pi})$ respectively, such that*

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

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

In the full version of this paper we treat the problem in full generality, but for the sake of simplicity here we assume that $d = \tilde{d} = \mathrm{rank}(A)$, and $\varsigma_1 < 1$. Then we can decompose the Hilbert space $\mathcal{H}_U$ to invariant orthogonal subspaces associated to the above SVD in the following way. For all $i \in [d]$ we define

$$|\psi_i^\perp\rangle := \frac{(I-\Pi)U^\dagger|\tilde{\psi}_i\rangle}{\left\|(I-\Pi)U^\dagger|\tilde{\psi}_i\rangle\right\|} = \frac{(I-\Pi)U^\dagger|\tilde{\psi}_i\rangle}{\sqrt{1-\varsigma_i^2}},$$
$$|\tilde{\psi}_i^\perp\rangle := \frac{(I-\widetilde{\Pi})U|\psi_i\rangle}{\left\|(I-\widetilde{\Pi})U|\psi_i\rangle\right\|} = \frac{(I-\widetilde{\Pi})U|\psi_i\rangle}{\sqrt{1-\varsigma_i^2}},$$
$$\mathcal{H}_i := \mathrm{Span}\big(|\psi_i\rangle, |\psi_i^\perp\rangle\big) \quad \text{and} \quad \tilde{\mathcal{H}}_i := \mathrm{Span}\big(|\tilde{\psi}_i\rangle, |\tilde{\psi}_i^\perp\rangle\big). \tag{3}$$

Finally let $\mathcal{H}_\perp := \left(\bigoplus_{i\in[d]} \mathcal{H}_i\right)^\perp$, and $\tilde{\mathcal{H}}_\perp := \left(\bigoplus_{i\in[d]} \tilde{\mathcal{H}}_i\right)^\perp$.

Now we introduce some notation for matrices that represent linear maps acting between different subspaces. For two vector (sub)spaces $\mathcal{H}, \mathcal{H}'$ let us denote by $[\,\cdot\,]_{\mathcal{H}'}^{\mathcal{H}}$ the matrix of a linear map $\mathcal{H} \to \mathcal{H}'$. Moreover, if the subspaces are as in (3) 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 (3). This will enable us to conveniently express matrices in a block-diagonal form. Using the SVD of Definition 7 we get that

$$U = \bigoplus_{i\in[d]} \begin{bmatrix} \varsigma_i & \sqrt{1-\varsigma_i^2} \\ \sqrt{1-\varsigma_i^2} & -\varsigma_i \end{bmatrix}_{\tilde{\mathcal{H}}_i}^{\mathcal{H}_i} \oplus [\,\cdot\,]_{\tilde{\mathcal{H}}_\perp}^{\mathcal{H}_\perp}. \tag{4}$$

Moreover, $2\Pi - I$ and $e^{i\phi(2\Pi-I)}$ respectively can be written as

$$2\Pi - I = \bigoplus_{i\in[d]} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}_{\mathcal{H}_i}^{\mathcal{H}_i} \oplus [\,\cdot\,]_{\mathcal{H}_\perp}^{\mathcal{H}_\perp}, \tag{5}$$

$$e^{i\phi(2\Pi-I)} = \bigoplus_{i\in[d]} \begin{bmatrix} e^{i\phi} & 0 \\ 0 & e^{-i\phi} \end{bmatrix}_{\mathcal{H}_i}^{\mathcal{H}_i} \oplus [\,\cdot\,]_{\mathcal{H}_\perp}^{\mathcal{H}_\perp}. \tag{6}$$

Similarly, $2\widetilde{\Pi} - I$ and $e^{i\phi(2\widetilde{\Pi}-I)}$ respectively can be written as

$$2\widetilde{\Pi} - I = \bigoplus_{i\in[d]} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}_{\tilde{\mathcal{H}}_i}^{\tilde{\mathcal{H}}_i} \oplus [\,\cdot\,]_{\tilde{\mathcal{H}}_\perp}^{\tilde{\mathcal{H}}_\perp}, \tag{7}$$

$$e^{i\phi(2\widetilde{\Pi}-I)} = \bigoplus_{i\in[d]} \begin{bmatrix} e^{i\phi} & 0 \\ 0 & e^{-i\phi} \end{bmatrix}_{\tilde{\mathcal{H}}_i}^{\tilde{\mathcal{H}}_i} \oplus [\,\cdot\,]_{\tilde{\mathcal{H}}_\perp}^{\tilde{\mathcal{H}}_\perp}. \tag{8}$$

Definition 8 (Alternating phase modulation sequences). *Assume the notation of Definition 7. Let $\Phi \in \mathbb{R}^n$; we define the* alternating phase modulation sequence $U_\Phi^{(APM)}$ *as follows*

$$U_\Phi^{(APM)} := \begin{cases} U\ e^{i\phi_n(2\Pi-I)} \cdots U^\dagger e^{i\phi_2(2\widetilde{\Pi}-I)} U e^{i\phi_1(2\Pi-I)} \\ U^\dagger e^{i\phi_n(2\widetilde{\Pi}-I)} \cdots U^\dagger e^{i\phi_2(2\widetilde{\Pi}-I)} U e^{i\phi_1(2\Pi-I)} \end{cases}$$

for odd and even n respectively.

Definition 9 (Singular value transformation by 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 an SVD of A. We define* singular value transformation *corresponding to f as*

$$f^{(SV)}(A) := \begin{cases} \sum_{i=1}^{d_{\min}} f(\varsigma_i)|\tilde{\psi}_i\rangle\langle\psi_i| & \text{for odd } n, \text{ and} \\ \sum_{i=1}^{d} f(\varsigma_i)|\psi_i\rangle\langle\psi_i| & \text{for even } n, \end{cases}$$

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 [42, Theorem 4]. Our result is more general because it works for arbitrary matrices as opposed to only Hermitian (or normal) matrices. Another improvement is that we remove the constraint $P_{\mathfrak{R}}(0) = 0$ for even d. Also we note that the following theorem can be viewed as a generalization of the quantum walk techniques introduced by Szegedy [53].

Theorem 10 (QSVT by alternating phase modulation). *Assume the notation of Definition 7. Let $P \in \mathbb{C}[x]$ and $\Phi \in \mathbb{R}^n$ be as in Corollary 5. Then*

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

Proof. In the special case when $d = \tilde{d} = \mathrm{rank}(A)$, and $\varsigma_1 < 1$, this follows from the structure of the matrices in (4)-(8). The proof of the general case is similar. □

Corollary 11 (Singular value transformation by real polynomials). *Assume the notation of Definition 7. Suppose that $P_{\mathfrak{R}} \in \mathbb{R}[x]$ is a degree-n polynomial satisfying both*

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

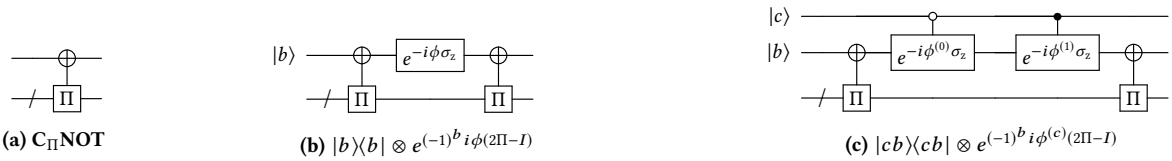
Then there exist $\Phi \in \mathbb{R}^n$ such that $P_{\mathfrak{R}}^{(SV)}\Big(\widetilde{\Pi}U\Pi\Big)$ equals

$$\Big(\langle+| \otimes \widetilde{\Pi}\Big)\Big(|0\rangle\langle 0| \otimes U_\Phi^{(APM)} + |1\rangle\langle 1| \otimes U_{-\Phi}^{(APM)}\Big)\Big(|+\rangle \otimes \Pi\Big) \tag{9}$$

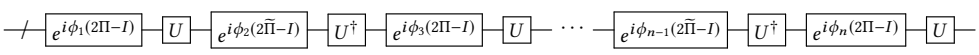
if n is odd. If n is even, it equals

$$\Big(\langle+| \otimes \Pi\Big)\Big(|0\rangle\langle 0| \otimes U_\Phi^{(APM)} + |1\rangle\langle 1| \otimes U_{-\Phi}^{(APM)}\Big)\Big(|+\rangle \otimes \Pi\Big). \tag{10}$$

Proof. By Corollary 5 we can find $P \in \mathbb{C}[x]$ of degree d and a $\Phi \in \mathbb{R}^d$ such that $\mathfrak{R}[P] = P_{\mathfrak{R}}$. Observe that $-\Phi$ gives rise to P^* in (2). Let $\Pi' = \widetilde{\Pi}$ for n odd and let $\Pi' = \Pi$ for n even. Then by Theorem 10 we get that $P^{(SV)}\Big(\widetilde{\Pi}U\Pi\Big) = \Pi' U_\Phi^{(APM)}\Pi$, and



(a) $C_\Pi NOT$ **(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) $U_\Phi^{(APM)} = U e^{i\phi_n(2\Pi-I)} U^\dagger e^{i\phi_{n-1}(2\widetilde{\Pi}-I)} \cdots U e^{i\phi_3(2\Pi-I)} U^\dagger e^{i\phi_2(2\widetilde{\Pi}-I)} U e^{i\phi_1(2\Pi-I)}$ **(for odd n)**

Figure 3: Gates and gate sequences used for singular value transformation in Theorem 10. Figure 3a shows how to implement a $C_\Pi NOT$ gate, and Figure 3b shows how to implement $e^{i\phi(2\Pi-I)}$ using a single ancilla qubit, two $C_\Pi NOT$ gates, and an $e^{-i\phi\sigma_z}$ gate. Figure 3c 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 3d summarizes the complete circuit used in Theorem 10.

$P^{*(SV)}\left(\widetilde{\Pi}U\Pi\right) = \Pi' U_{-\Phi}^{(APM)}\Pi$. Therefore

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

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

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

Regarding the parity constraint, note that if P is an even polynomial, then $P^{(SV)}(A) = S(A^\dagger A)$, where $S(x)$ is the polynomial $P(\sqrt{x})$. Similarly if P' is an odd polynomial, then we have $P'^{(SV)}(A) = A \cdot S'(A^\dagger A)$ for the analogous choice of S'. This shows that these transformations can be performed without explicitly calculating the SVD, which seems to be necessary for efficient quantum singular value transformation.

What remains is to discuss how to efficiently implement alternating phase modulation sequences. For this we introduce *projector-controlled NOT gates*. For an orthogonal projector Π we will frequently use the Π-controlled NOT gate, denoted by $C_\Pi NOT$, which flips the value of the first (single-qubit) register whenever the state of the second register is in the image of Π. For example if $\Pi = |1\rangle\langle 1|$ is the projector to the single-qubit basis state $|1\rangle$, then $C_\Pi NOT$ is just the CNOT gate controlled by the second qubit.

Definition 12 ($C_\Pi NOT$ gate)**.** *For an orthogonal projector Π let us define the Π-controlled NOT gate as the unitary operator*

$$C_\Pi NOT := X \otimes \Pi + I \otimes (I - \Pi).$$

This operator can be used for instance for implementing a coherent (unitary) analogue of a projective measurement. Note that up to a conjugation by a Hadamard gate on the first qubit, this gate is the same as the controlled reflection $(I - 2\Pi)$.

The operator $e^{i\phi(2\Pi-I)} = C_\Pi NOT\left(I\otimes e^{-i\phi\sigma_z}\right)C_\Pi NOT$ can be implemented using a single ancilla qubit, two uses of $C_\Pi NOT$, and a single-qubit phase gate $e^{-i\phi\sigma_z}$. This leads to an efficient implementation of $U_\Phi := |0\rangle\langle 0| \otimes U_\Phi^{(APM)} + |1\rangle\langle 1| \otimes U_{-\Phi}^{(APM)}$, and in turn of Corollary 11, as illustrated by Figure 3.

3.3 Robustness of Singular Value Transformation

We prove results about the robustness of singular value transformation. More precisely we prove bounds on $\left\|P^{(SV)}(A) - P^{(SV)}(\tilde{A})\right\|$ in terms of the magnitude of the initial "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 this is not true in general. For a recent survey on the topic, see the work of Aleksandrov and Peller [2].

Although the Lipschitz property cannot be saved directly, one need not lose more than some logarithmic factors in the modulus of continuity. We use a result form the theory of operator functions due to Farforovskaya and Nikolskaya [21, Theorem 10], leading to the following corollary.

Corollary 13 (Robustness of singular value transformation)**.** *Suppose that $f\colon [-1,1] \to \mathbb{C}$ is an even or odd function such that $\omega\colon [0,2] \to [0,\infty]$ is a modulus of continuity, i.e., for all $x, x' \in [-1,1]$ we have that $|f(x) - f(x')| \leq \omega(|x - x'|)$. Then for all $A, \tilde{A} \in \mathbb{C}^{\tilde{d}\times d}$ matrices of operator norm at most 1*

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

We prove the above corollary in the full version of this paper [26], alongside with two other versions, which in some cases enable us to remove the log factors form the above claim.

4 DIRECT APPLICATIONS OF QSVT

In this section we show some direct applications of QSVT and sketch their proofs. For detailed proofs and more discussions about the connection to Szegedy quantum walks [53], we refer to the full version [26].

Most applications presented here will rely on a few basic polynomial approximation results. Similarly to the work of Low and Chuang [42, Corollary 6] we need to construct polynomial approximations of the sign function.

Lemma 14 (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 $n = O\left(\frac{\log(1/\varepsilon)}{\delta}\right)$, such that*

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

As a first application, observe that Theorem 1 immediately follows from Lemma 14 and Corollary 11.

Another easy-to-derive corollary of our machinery is robust oblivious amplitude amplification [11], which is a technique originally developed [10] for Hamiltonian simulation.[6] This algorithm solves the following problem: given one copy of a quantum state $|\psi\rangle$, apply the unitary U to this state, having access only to a subnormalized implementation of U in the form of a block-encoding, say $(\langle 0| \otimes I)V(|0\rangle \otimes I) = U/100$. For more discussion about this problem see the full version [26].

Theorem 15 (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*

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

for all $|\psi\rangle \in \mathrm{img}(\Pi)$. Then we can construct a unitary $\tilde{U}$ that for all $|\psi\rangle \in \mathrm{img}(\Pi)$ satisfies $\left\|W|\psi\rangle - \widetilde{\Pi}\tilde{U}|\psi\rangle\right\| \leq 2n\varepsilon$, which uses a single ancilla qubit, with n uses of U and $U^\dagger$, n uses of C_ΠNOT, and n uses of $C_{\widetilde{\Pi}}$NOT gates, and n single-qubit gates.

Proof. Use the Chebyshev polynomial T_n in Theorem 10, together with one of our robustness resutls. □

For our next application it will be useful to derive a corollary of Lemma 14 about polynomial approximations of the rectangle function.

Corollary 16 (Polynomial approximations of the rectangle function). *Let $\delta', \varepsilon' \in (0, \frac{1}{2})$ and $t \in [-1, 1]$. There exists an even polynomial $P' \in \mathbb{R}[x]$ of degree $O\left(\log(\frac{1}{\varepsilon'})/\delta'\right)$, such that $|P'(x)| \leq 1$ for all $x \in [-1, 1]$, $P'(x) \in [1 - \varepsilon', 1]$ for all $x \in [-t + \delta', t - \delta']$, and $P'(x) \in [0, \varepsilon']$ for all $x \in [-1, -t - \delta'] \cup [t + \delta', 1]$.*

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

Theorem 17 (Uniform singular value amplification). *Assume the notation of Definition 7. Let $\gamma > 1$ and let $\delta, \varepsilon \in (0, \frac{1}{2})$. Then there is an $m = O\left(\frac{\gamma}{\delta}\log\left(\frac{\gamma}{\varepsilon}\right)\right)$ and a $\Phi \in \mathbb{R}^m$ such that $(\langle +| \otimes \widetilde{\Pi})U_\Phi(|+\rangle \otimes \Pi) = \sum_i \tilde{\varsigma}_i |\tilde{\psi}_i\rangle\langle\psi_i|$, where $\left\|\frac{\tilde{\varsigma}_i}{\gamma\varsigma_i} - 1\right\| \leq \varepsilon$ whenever $\varsigma_i \leq (1 - \delta)/\gamma$.*

Proof. In Corollary 16 set $t := \frac{1-\delta/2}{\gamma}$, $\delta' := \frac{\delta}{2\gamma}$ and $\varepsilon' := \frac{\varepsilon}{\gamma}$ in order to get an even polynomial R of degree $O\left(\frac{\gamma}{\delta}\log\left(\frac{\gamma}{\varepsilon}\right)\right)$ that is an $\frac{\varepsilon}{\gamma}$-approximation of the rectangle function. Let us define $P_{\mathfrak{R}}(x) := \gamma \cdot x \cdot R(x)$, and use Corollary 11. □

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

4.1 Fast OR Lemma, Matrix Inversion (HHL) and Quantum Machine Learning Aspects

As another application we develop *singular value threshold projectors*, which project out singular vectors with singular values below a certain threshold. These threshold projectors play a major role in quantum algorithms recently proposed by Kerenidis et al. [35, 37], 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 [37].

Definition 18 (Singular value threshold projectors). *Let $A = \widetilde{\Pi}U\Pi = W\Sigma V^\dagger$ be an SVD of a projected unitary. For $S \subseteq \mathbb{R}$ let Σ_S be the matrix that we get from Σ by replacing all diagonal entries $\Sigma_{ii} \in S$ by 1 and replacing all diagonal entries $\Sigma_{jj} \notin S$ by 0. 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 $\theta \in \mathbb{R}$ we define $\Pi_{\geq\theta} := \Pi_{[\theta,\infty)}$, and also define $\Pi_{>\theta}, \Pi_{\leq\theta}, \Pi_{<\theta}$, and $\widetilde{\Pi}_{>\theta}, \widetilde{\Pi}_{\leq\theta}, \widetilde{\Pi}_{<\theta}$ analogously.*

By combing Corollaries 11 and 16, we give an efficient approximate singular value threshold projector implementation.

Theorem 19 (Implementing singular value threshold projectors). *Assume the notation of Definition 7. Let $\theta, \delta, \varepsilon \in (0, 1)$. Then there is an $m = O\left(\frac{\log(1/\varepsilon)}{\delta}\right)$ and a $\Phi \in \mathbb{R}^m$ such that $(\langle +| \otimes \Pi)U_\Phi(|+\rangle \otimes \Pi) = \sum_i \tilde{\varsigma}_i |\psi_i\rangle\langle\psi_i|$, where $\tilde{\varsigma}_i \in [1-\varepsilon, 1]$ whenever $\varsigma_i \geq \theta + \delta$ and $\tilde{\varsigma}_i \in [0, \varepsilon]$ whenever $\varsigma_i \leq \theta - \delta$.*

A useful application of singular value threshold projectors is *singular value discrimination*, which decides whether a given quantum state has singular value below or above a certain threshold.

Theorem 20 (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 by a polynomial of degree $O\left(\log(1/\varepsilon)/\max\left[b-a, \sqrt{1-a^2} - \sqrt{1-b^2}\right]\right)$. Moreover, if $a = 0$ or $b = 1$, then we can make the error one sided.*

Proof. Apply a singular vector projector corresponding to either $\widetilde{\Pi}U\Pi$ or $(I - \widetilde{\Pi})U\Pi$, depending on the values a, b. □

Now we turn to the quantum OR lemma. The problem is the following: we are given a collection of m projective measurements, and a mixed quantum state ρ. We would like to compute the OR function of the measurement outcomes, under the promise that for each projective measurement, the measurement probability is either very close to 1 or very close to 0. The quantum OR lemma of Harrow et al. [31] shows that we can solve this problem using only a constant number of copies of ρ. The fast quantum OR lemma of Brandão et al. [13] improves the procedure's computational complexity, which we quickly rederive with slightly improved error dependence.

Theorem 21 (Fast quantum OR lemma). *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*

(i) either there exists some $i \in [m]$ such $\mathrm{Tr}[\rho\Pi_i] \geq 1-\eta$,
(ii) or $\frac{1}{m}\sum_{j=1}^{m}\mathrm{Tr}[\rho\Pi_j] \leq \nu$.

Let V be a unitary such that $(\langle i|\otimes I)V(|i\rangle\otimes I) = C_{\Pi_i}\mathrm{NOT}$ 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 $O\left(\sqrt{m}\log\left(\frac{1}{\varepsilon}\right)\right)$ times and uses $O\left(\sqrt{m}\log(m)\log\left(\frac{1}{\varepsilon}\right)\right)$ other gates and $O(\log(m))$ ancilla qubits.

Proof. Follow the analysis of Harrow et al. [31] and apply Theorem 20 to $A := \frac{1}{m}\sum_{i=1}^{m}(I-\Pi_i)$, observing that $I - \Pi_i = (\langle 0|\otimes I)\mathrm{C}_{\Pi_i}\mathrm{NOT}(|0\rangle\otimes I)$. □

The improved version of the Marriott-Watrous [46] QMA amplification procedure due Nagaj et al. [47] can also be seen as a direct corollary of our singular value discrimination result. For more details see the full version [26].

4.1.1 Principal component regression and the HHL algorithm. The famous quantum algorithm of Harrow, Hassidim, and Lloyd [30] solves the system of linear equations $Ax = b$ in a very quantum sense. Given suitable access to the matrix A, and the ability to prepare a quantum state $|b\rangle$, it prepares a quantum state proportional to $|x\rangle$. The most common assumption for the matrix A is that it is s-sparse and that the entries of the matrix are efficiently computable, as well as the locations of the non-zero entries. Then the HHL algorithm runs in time that depends polynomially on the sparsity and the condition number of A, and polylogarithmically on the dimension and the desired precision. When A is not invertible, then one can still prepare a state proportional to the least-square solution $A^+|b\rangle$, by appropriately defining a substitute for the condition number.

In machine learning applications, the matrix A is not always well-conditioned. However it is sometimes useful to discard small singular values. This is the problem of principal component regression, which is a generalization of the matrix (pseudo)inversion problem. We formally state the problem as follows [23]: given a matrix $A \in \mathbb{R}^{n\times d}$, a vector $b \in \mathbb{R}^n$, and a threshold value $0 < \theta$, find $x \in \mathbb{R}^d$ of minimal norm such that

$$x = \mathrm{argmin}_{x\in\mathbb{R}^d}\left\|\widetilde{\Pi}_{\geq\theta}A\Pi_{\geq\theta}x - b\right\|, \tag{11}$$

where $\widetilde{\Pi}_{\geq\theta}, \Pi_{\geq\theta}$ denote left and right singular value threshold projectors. A closed-form expression for the optimal solution of (11) is given by $x = \Pi_{\geq\theta}A^+\widetilde{\Pi}_{\geq\theta}b = A^+\widetilde{\Pi}_{\geq\theta}b$.

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

Corollary 22 (Implementing the threshold pseudoinverse)**.** *Let U, Π, $\widetilde{\Pi}$ form a projected unitary encoding of the matrix A, and let $\varepsilon, \theta \in (0, \frac{1}{2}]$ and $\delta \in (0, \theta)$. Then there is an $m = O\left(\frac{1}{\delta}\log(\frac{1}{\varepsilon})\right)$ and a $\Phi \in \mathbb{R}^m$ such that corresponding matrix $\left(\langle +|\otimes\left(\Pi - \Pi_{[\theta-\delta,\theta+\delta]}\right)\right)U_\Phi\left(|+\rangle\otimes\left(\widetilde{\Pi} - \widetilde{\Pi}_{[\theta-\delta,\theta+\delta]}\right)\right)$ is ε-close to $\Pi_{\geq\theta}\left(\frac{\theta}{2}A^+\right)\widetilde{\Pi}_{\geq\theta}$ in spectral norm.*

Proof. Our proof is based on the following: let $f(x) := 0$ for $x \in (-\theta, \theta)$, and let $f(x) := 1/x$ otherwise. Then we have $\Pi_{\geq\theta}A^+\widetilde{\Pi}_{\geq\theta} = f^{(SV)}(A^\dagger)$. Now construct an odd polynomial that is bounded by 1 on $[-1, 1]$, ε-approximates the function $\frac{\theta}{2x}$ on the interval $[\theta+\delta, 1]$, and is ε-close to 0 for all $x \in [0, \theta-\delta]$. This can be done, e.g., using Corollary 23. □

Given a unitary preparing a quantum state $|b\rangle$, we can approximately perform principal component regression by applying the above approximation of $\Pi_{\geq\theta}\left(\frac{\theta}{2}A^+\right)\widetilde{\Pi}_{\geq\theta}$ to $|b\rangle$, and then applying amplitude amplification to get $|x\rangle$.

4.2 Bounded Polynomial Approximations of Piecewise Smooth Functions

Corollary 23 (Taylor series based bounded polynomial approximations)**.** *Let $x_0 \in [-1, 1]$, $r \in (0, 2]$, $\delta \in (0, r]$, and let $f : [-x_0 - r - \delta, x_0 + r + \delta] \to \mathbb{C}$ 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 $O\left(\frac{1}{\delta}\log\left(\frac{B}{\varepsilon}\right)\right)$ s.t.*

$$\|f(x) - P(x)\|_{[x_0-r, x_0+r]} \leq \varepsilon \tag{12}$$

$$\|P(x)\|_{[-1,1]} \leq \varepsilon + \|f(x)\|_{[x_0-r-\delta/2, x_0+r+\delta/2]} \leq \varepsilon + B \tag{13}$$

$$\|P(x)\|_{[-1,1]\setminus[x_0-r-\delta/2, x_0+r+\delta/2]} \leq \varepsilon. \tag{14}$$

In the full version of this paper [26] we also devise a version of the above for combining multiple local Taylor series.

5 MATRIX ARITHMETICS USING BLOCKS OF UNITARIES

In this section we describe a generic toolbox for implementing matrix calculations on a quantum computer, representing matrices by unitary circuits and vectors as quantum states. The methodology we describe is a distilled version of the results of a series of works on quantum algorithms [6, 11, 16, 17, 30, 41]. Our results in some cases slightly improve or generalize earlier approaches; for the proofs we refer to the full version [26].

5.1 Block-encoding

Definition 24 (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.

As block-encodings defined above are special cases of the projected unitary encodings in Definition 7, our QSVT results are all applicable to block-encoded matrices. 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 3 is just a Toffoli gate on $a+1$ qubits,

which can be implemented by $O(a+1)$ two-qubit gates and using a single additional ancilla qubit [32].

5.2 Creating & Combining Block-encodings

A unitary matrix is a $(1,0,0)$-block-encoding of itself, which we call a *trivial block-encoding*. 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.

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 registers.

Low and Chuang [41] showed, how to construct a block-encoding of a purified density operator. This technique can be used in combination with the optimal block-Hamiltonian simulation result Corollary 32, in order to get much better simulation performance, compared to density matrix exponentiation techniques [38, 40] which does not use purification. This result can be generalized for subnormalized density operators too, for more details see [4].

Lemma 25 (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 $|0\rangle^{\otimes a}|0\rangle^{\otimes s}$ input state prepares a purification $|0\rangle^{\otimes a}|0\rangle^{\otimes s} \mapsto |\rho\rangle$, 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.*

Van Apeldoorn and Gilyén [4] recently also showed that an implementation scheme for a binary POVM measurement $\{M, I-M\}$ can also be easily transformed to block-encodings of the POVM operators. By an implementation scheme we mean a quantum circuit U that given an arbitrary input state ρ, sets a flag qubit to 0 with probability $\mathrm{Tr}[\rho M]$. (The CNOT gate in the following lemma is controlled on the second qubit.)

Lemma 26 (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 operators ρ*

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

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.

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. Now we introduce the notation $[n]-1$ to denote the set $\{0, 1, \ldots, n-1\}$.

Lemma 27 (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 $\{|\psi_i\rangle : i \in [2^s]-1\}$, $\{|\phi_j\rangle : j \in [2^s]-1\}$, s.t.*

$$U_L : |0\rangle|i\rangle \to |\psi_i\rangle \quad \text{and} \quad U_R : |0\rangle|j\rangle \to |\phi_j\rangle.$$

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 this one can efficiently implement block-encodings of sparse matrices. Indeed, if a matrix A is s_r-row-sparse and s_c-column-sparse, and each element of A has absolute value at most 1, then we can efficiently construct a block-encoding of $A/\sqrt{s_r s_c}$ with $O(1)$ queries to the standard sparse-access oracles. In some cases one can get further improvements using preamplification. For more details see the full version [26].

Finally, we note that for matrices that are stored in a clever quantum data structure in QROM, it is also possible to implement the corresponding block-encodings efficiently [16, 36].

5.2.1 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. [11] for exponentially improving the precision of Hamiltonian simulation, and was later also used by Childs et al. [17] for exponentially improving the precision of quantum linear equation solving.

Definition 28 (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 29 (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 $P_L^\dagger$.*

5.2.2 Product of block-encoded matrices. In general if we are 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 30 (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[7] $(I_b \otimes U)(I_a \otimes V)$ is an $(\alpha\beta, a+b, \alpha\varepsilon + \beta\delta)$-block-encoding of AB.*

6 IMPLEMENTING SMOOTH FUNCTIONS OF HERMITIAN MATRICES

In the previous section, we developed an efficient methodology to perform basic matrix arithmetic, 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 a more efficient way to transform Hermitian matrices according to smooth functions using singular

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

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 are the same.

The following theorem is our improvement of Corollary 11 removing the counter-intuitive parity constraint at the expense of a subnormalization factor 1/2, which is not a problem in most applications.

Theorem 31 (Polynomial eigenvalue transformation of arbitrary parity)**.** *Suppose that U is an (α, a, ε)-block-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 which consists of d applications of U and $U^\dagger$ gates, a single application of controlled-U, and $O((a+1)d)$ other one- and two-qubit gates. Moreover we can compute a description of such a circuit with a classical computer in time $O(\mathrm{poly}(d, \log(1/\delta)))$.*

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 from the fact that the implementation is a sum of 4 different terms (even/odd component of the real/imaginary part).

Now we describe some applications of the above theorem. For the proofs we refer to the full version [26].

6.1 Optimal Hamiltonian Simulation

We can quickly rederive the optimal Hamiltonian simulation results of Low and Chuang [41]. Using Theorems 15 and 31, and polynomial approximations of e^{itx} (e.g, obtained by Taylor truncation), we get the following upper bound, and in the full version [26] we also derive a robust version. (The lower bound follows from the argument laid out in [43].)

Corollary 32 (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*

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

6.2 Fractional Queries

Scott Aaronson listed as one of "The ten most annoying questions in quantum computing" [1] the following problem: given a unitary U, can we implement $\sqrt{U}$? This was positively answered by Sheridan et al. [50]. We show how to improve their complexity, in some cases exponentially.

Corollary 33 (Taking the logarithm of unitaries, see also [42])**.** *Suppose that $U = e^{iH}$, where H is a Hamiltonian of spectral norm at most $\frac{\pi}{2} - \delta$ for some $\delta \in (0,1)$. Let $\varepsilon \in (0, \frac{1}{2}]$, then we can implement a $(\frac{\pi}{2}, 2, \varepsilon)$-block-encoding of H with $\left\lceil \frac{8}{\delta^2} \ln\left(\frac{1}{\varepsilon}\right)\right\rceil$ uses of controlled-U and its inverse, using $O\left(\frac{1}{\delta^2}\log\left(\frac{1}{\varepsilon}\right)\right)$ two-qubit gates and using a single ancilla qubit.*

Corollary 34 (Implementing fractional queries)**.** *Let $U = e^{iH}$, where H is a Hamiltonian of norm at most 1. If $\varepsilon \in (0, \frac{1}{2}]$ and $t \in [-1,1]$, then we can implement an ε-approximation of $U^t = e^{itH}$ with $O\left(\log\left(\frac{1}{\varepsilon}\right)\right)$ uses of controlled-U and its inverse, using $O\left(\log\left(\frac{1}{\varepsilon}\right)\right)$ two-qubit gates and $O(1)$ ancilla qubits.*

If we would be promised that, say $\|H\| \leq \pi - \delta$, the above can be modified giving complexity $O(\log(1/\varepsilon)/\delta)$, which exponentially improves the complexity $O\left(\max[\frac{1}{\delta}, \frac{1}{\varepsilon}]\log\left(\frac{1}{\varepsilon}\right)\right)$ of Sheridan et al. [50] in the case of $\delta = \Theta(1)$.

6.2.1 Gibbs-sampling. For a Hermitian matrix H and inverse temperature $\beta > 0$, the task of *Gibbs-sampling* [20, 48] is to prepare a mixed quantum state $\rho = e^{-\beta H}/\mathrm{Tr}[e^{-\beta H}]$. This is a primitive related to statistical physics that turns out to be also useful for (quantum) convex optimization [4–6, 8, 13, 14].

Now we describe a result regarding Gibbs-sampling. By preparing a maximally entangled state on two registers, applying the map $e^{-\frac{\beta}{2}(H+I)}$ to the first register, and tracing out the second register, one gets a subnormalized Gibbs state $e^{-\beta(H+I)}$ in the first register. Using (fixed-point) amplitude amplification, one can get a purification of the Gibbs state. Each of these steps can be compactly performed using singular value transformation techniques, providing an efficient implementation.

Theorem 35. *Let $\varepsilon \in (0, 1/2)$, $\beta = \Omega(1)$, and suppose that U is an a-qubit block-encoding of the Hamiltonian $H \in \mathbb{C}^{n\times n}$. Then we can prepare a pure state on two registers, so that tracing out the first register yields a density operator ε-close to the Gibbs state $e^{-\beta H}/\mathrm{Tr}\left[e^{-\beta H}\right]$ in trace distance, with $O\left(\beta\sqrt{n}\log^3\left(\frac{n}{\varepsilon}\right)\right)$ uses of (controlled) U, $U^\dagger$, and $O(a)$ times more two-qubit gates.*

7 CONCLUSION

Our main contribution is to provide a new quantum algorithmic paradigm based on efficient transformations of block-encoded matrices. This unifies a host of quantum algorithms ranging from linear equation solving to quantum simulation and quantum walks. Prior to this contribution, each of these algorithms would have to be understood separately, which makes mastering all of them a challenge. By presenting them all within the framework of quantum singular value transformation, many of the popular quantum algorithmic primitives become direct corollaries. This greatly simplifies entering the field while also reveals hitherto unknown algorithms.

ACKNOWLEDGMENTS

A.G. thanks Ronald de Wolf, Robin Kothari, Joran van Apeldoorn, Shantanav Chakraborty, Stacey Jeffery, Vedran Dunjko, Yimin Ge, Ashwin Nayak and Arjan Cornelissen for inspiring and insightful 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

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

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

[3] Andris Ambainis. 2004. Quantum Walk Algorithm for Element Distinctness. In *Proceedings of the 45th IEEE Symposium on Foundations of Computer Science (FOCS)*. 22–31. arXiv: quant-ph/0311001

[4] Joran van Apeldoorn and András Gilyén. 2019. Improvements in Quantum SDP-Solving with Applications. In *Proceedings of the 46th International Colloquium on Automata, Languages, and Programming (ICALP)*. (to appear) arXiv: 1804.05058

[5] Joran van Apeldoorn and András Gilyén. 2019. Quantum algorithms for zero-sum games. (2019). arXiv: 1904.03180

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

[7] Simon Apers and Alain Sarlette. 2018. Quantum Fast-Forwarding Markov Chains. (2018). arXiv: 1804.02321

[8] Sanjeev Arora and Satyen Kale. 2016. A Combinatorial, Primal-Dual Approach to Semidefinite Programs. *Journal of the ACM* 63, 2 (2016), 12:1–12:35. Earlier version in STOC'07.

[9] Dominic W. Berry, Graeme Ahokas, Richard Cleve, and Barry C. Sanders. 2007. Efficient Quantum Algorithms for Simulating Sparse Hamiltonians. *Communications in Mathematical Physics* 270, 2 (2007), 359–371. arXiv: quant-ph/0508139

[10] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma. 2014. Exponential improvement in precision for simulating sparse Hamiltonians. In *Proceedings of the 46th ACM Symposium on the Theory of Computing (STOC)*. 283–292. arXiv: 1312.1414

[11] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma. 2015. Simulating Hamiltonian Dynamics with a Truncated Taylor Series. *Physical Review Letters* 114, 9 (2015), 090502. arXiv: 1412.4687

[12] Dominic W. Berry, Andrew M. Childs, and Robin Kothari. 2015. Hamiltonian Simulation with Nearly Optimal Dependence on all Parameters. In *Proceedings of the 56th IEEE Symposium on Foundations of Computer Science (FOCS)*. 792–809. arXiv: 1501.01715

[13] Fernando G. S. L. Brandão, Amir Kalev, Tongyang Li, Cedric Yen-Yu Lin, Krysta M. Svore, and Xiaodi Wu. 2019. Quantum SDP Solvers: Large Speed-ups, Optimality, and Applications to Quantum Learning. In *Proceedings of the 46th International Colloquium on Automata, Languages, and Programming (ICALP)*. (to appear) arXiv: 1710.02581

[14] Fernando G. S. L. Brandão and Krysta M. Svore. 2017. Quantum Speed-ups for Solving Semidefinite Programs. In *Proceedings of the 58th IEEE Symposium on Foundations of Computer Science (FOCS)*. 415–426. arXiv: 1609.05537

[15] Gilles Brassard, Peter Høyer, Michele Mosca, and Alain Tapp. 2002. Quantum Amplitude Amplification and Estimation. In *Quantum Computation and Quantum Information: A Millennium Volume*. Contemporary Mathematics Series, Vol. 305. AMS, 53–74. arXiv: quant-ph/0005055

[16] Shantanav Chakraborty, András Gilyén, and Stacey Jeffery. 2019. The power of block-encoded matrix powers: improved regression techniques via faster Hamiltonian simulation. In *Proceedings of the 46th International Colloquium on Automata, Languages, and Programming (ICALP)*. (to appear) arXiv: 1804.01973

[17] Andrew M. Childs, Robin Kothari, and Rolando D. Somma. 2017. Quantum Algorithm for Systems of Linear Equations with Exponentially Improved Dependence on Precision. *SIAM Journal on Computing* 46, 6 (2017), 1920–1950. arXiv: 1511.02306

[18] Andrew M. Childs, Dmitri Maslov, Yunseong Nam, Neil J. Ross, and Yuan Su. 2017. Toward the first quantum simulation with quantum speedup. (2017). arXiv: 1711.10980

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

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

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

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

[23] Roy Frostig, Cameron Musco, Christopher Musco, and Aaron Sidford. 2016. Principal Component Projection Without Principal Component Analysis. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*. 2349–2357. arXiv: 1602.06872

[24] András Gilyén. 2019. *Quantum Singular Value Transformation & Its Algorithmic Applications*. Ph.D. Dissertation. University of Amsterdam. Advisor(s) Ronald de Wolf.

[25] András Gilyén and Or Sattath. 2017. 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)*. 439–450. arXiv: 1611.08571

[26] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. 2018. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. [Full version] arXiv: 1806.01838

[27] Lov K. Grover. 1996. A Fast Quantum Mechanical Algorithm for Database Search. In *Proceedings of the 28th ACM Symposium on the Theory of Computing (STOC)*. 212–219. arXiv: quant-ph/9605043

[28] Lov K. Grover. 2005. Fixed-Point Quantum Search. *Physical Review Letters* 95, 15 (2005), 150501. arXiv: quant-ph/0503205

[29] Jeongwan Haah. 2018. Product Decomposition of Periodic Functions in Quantum Signal Processing. (2018). arXiv: 1806.10236

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

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

[32] Yong He, Ming-Xing Luo, E. Zhang, Hong-Ke Wang, and Xiao-Feng Wang. 2017. Decompositions of n-qubit Toffoli Gates with Linear Circuit Complexity. *International Journal of Theoretical Physics* 56, 7 (2017), 2350–2361.

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

[34] Camille Jordan. 1875. Essai sur la géométrie à n dimensions. *Bulletin de la Société Mathématique de France* 3 (1875), 103–174. http://eudml.org/doc/85325

[35] Iordanis Kerenidis and Alessandro Luongo. 2018. Quantum classification of the MNIST dataset via Slow Feature Analysis. (2018). arXiv: 1805.08837

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

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

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

[39] Seth Lloyd. 1996. Universal Quantum Simulators. *Science* 273, 5278 (1996), 1073–1078.

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

[41] Guang Hao Low and Isaac L. Chuang. 2016. Hamiltonian Simulation by Qubitization. (2016). arXiv: 1610.06546

[42] Guang Hao Low and Isaac L. Chuang. 2017. Hamiltonian Simulation by Uniform Spectral Amplification. (2017). arXiv: 1707.05391

[43] Guang Hao Low and Isaac L. Chuang. 2017. Optimal Hamiltonian Simulation by Quantum Signal Processing. *Physical Review Letters* 118, 1 (2017), 010501. arXiv: 1606.02685

[44] Guang Hao Low, Theodore J. Yoder, and Isaac L. Chuang. 2016. Methodology of Resonant Equiangular Composite Quantum Gates. *Physical Review X* 6, 4 (2016), 041067. arXiv: 1603.03996

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

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

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

[48] David Poulin and Pawel Wocjan. 2009. Sampling from the Thermal Quantum Gibbs State and Evaluating Partition Functions with a Quantum Computer. *Physical Review Letters* 103, 22 (2009), 220502. arXiv: 0905.2199

[49] Sushant Sachdeva and Nisheeth K. Vishnoi. 2014. Faster Algorithms via Approximation Theory. *Found. Trends Theor. Comput. Sci.* 9, 2 (2014), 125–210.

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

[51] Peter W. Shor. 1997. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM Journal on Computing* 26, 5 (1997), 1484–1509. Earlier version in FOCS'94. arXiv: quant-ph/9508027

[52] Peter W. Shor. 2003. Why Haven't More Quantum Algorithms Been Found? *Journal of the ACM* 50, 1 (2003), 87–90.

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

[54] Theodore J. Yoder, Guang Hao Low, and Isaac L. Chuang. 2014. Fixed-Point Quantum Search with an Optimal Number of Queries. *Physical Review Letters* 113, 21 (2014), 210501. arXiv: 1409.3305