

Quantum Algorithms for Spectral Sums

Alessandro Luongo^{1,2}, Changpeng Shao³

¹Centre for Quantum Technologies (CQT), National University of Singapore, Singapore.

²Inveriant Pte. Ltd., Singapore.

³State Key Laboratory of Management and Systems (SKLMS), Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China
ale@nus.edu.sg,

Abstract

We propose new quantum algorithms for estimating spectral sums of positive semi-definite (PSD) matrices. The spectral sum of an PSD matrix A , for a function f , is defined as $\text{Tr}[f(A)] = \sum_j f(\lambda_j)$, where λ_j are the eigenvalues of A . Typical examples of spectral sums are the von Neumann entropy, the trace of A^{-1} , the log-determinant, and the Schatten p -norm, where the latter does not require the matrix to be PSD. The current best classical randomized algorithms estimating these quantities have a runtime that is at least linearly in the number of nonzero entries of the matrix and quadratic in the estimation error. Assuming access to a block-encoding of a matrix, our algorithms are sub-linear in the matrix size, and depend at most quadratically on other parameters, like the condition number and the approximation error, and thus can compete with most of the randomized and distributed classical algorithms proposed in the literature, and polynomially improve the runtime of other quantum algorithms proposed for the same problems. We show how the algorithms and techniques used in this work can be applied to three problems in spectral graph theory: approximating the number of triangles, the effective resistance, and the number of spanning trees in a graph.

1 Introduction

Spectral sums are matrix quantities that are central to many problems in computational sciences. They can be found in machine learning, computational chemistry, biology, statistics, finance, and many other disciplines (Rue and Held 2005; Rasmussen and Nickisch 2010; Cichocki, Cruces, and Amari 2015; Gutman 2001; Hardt, Ligett, and McSherry 2012; Nie, Huang, and Ding 2012; Bengtsson and Życzkowski 2017; Alter, Brown, and Botstein 2000; Golub and Von Matt 1997; Dashti and Stuart 2011). Given the wealth of applications for spectral sums, there is a significant interest in developing efficient algorithms to estimate these quantities (Han et al. 2017; Kontopoulou et al. 2018; Ubaru, Chen, and Saad 2017; Musco et al. 2017; Boutsidis et al. 2017; Wu et al. 2016; Han, Malioutov, and Shin 2015). A spectral sum is defined as follows.

Definition 1 (Spectral sum). *Let $A \in \mathbb{R}^{n \times n}$ be a symmetric matrix with $A = UDU^\dagger$ its eigenvalue decomposition,*

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

where $D = \text{diag}(\lambda_1, \dots, \lambda_n)$ is the diagonal matrix of the eigenvalues. Let $f : \mathbb{R} \rightarrow \mathbb{R}$ be a function. The spectral sum of A for the function f is defined as $\mathcal{S}_f(A) := \text{Tr}[f(A)] = \sum_{j=1}^n f(\lambda_j)$.

We can have an analogous definition using the singular values. As an example, consider the *logarithm* of the determinant, which is perhaps the most famous example of a spectral sum. The conventional definition of the determinant through the Laplace expansion lead to an algorithm of $O(n!)$ operations, lacking computational efficiency (Strang 2023). Fortunately, many practical applications often require the logarithm of the determinant, and algorithms tailored for the former have proven to be more advantageous in such scenarios, even compared to algorithms based on fast matrix multiplication, whose runtime is $O(n^{2.373})$ (Aho and Hopcroft 1974).

There are two key ingredients in most classical algorithms for spectral sums: a way to compute the approximation of a matrix function and a stochastic trace estimation technique. It is possible to compute a matrix function bypassing the costly diagonalization of the matrix, by using techniques from approximation theory and linear algebra. This idea is better explained by an example. Imagine you want to compute $\text{Tr}[\log(A)]$, for some PSD matrix A . Set $B = I - A$, and observe that for a fixed $m_1 \in \mathbb{N}$, $\text{Tr}[\log(I - B)] \approx \text{Tr}\left[-\sum_{k=1}^{m_1} \frac{B^k}{k}\right] = -\sum_{k=1}^{m_1} \frac{\text{Tr}[B^k]}{k}$. This trick bypasses the need for diagonalizing A , applying the function to the eigenvalues, and computing the sum, by reducing the problem to (repeated) matrix multiplications and stochastic trace estimation algorithms, for which we have efficient randomized algorithm. Most of these algorithms work in the *matrix-vector product model*, formalized in (Rashtchian, Woodruff, and Zhu 2020; Sun et al. 2021). In this model we are given access to an oracle, which returns a vector Ax , for a PSD matrix A and a vector x . These subroutines are a kind of Monte Carlo algorithm, and the estimate of the trace can be seen as random variables whose expected value is the trace of A . For example, consider a random vector z whose entries are Rademacher random variables (i.e. $\Pr[+1] = 1/2 = \Pr[-1]$). It is known (Avron and Toledo 2011, Lemma 1) that $\mathbb{E}[z^T A z] = \text{Tr}[A]$. This is called the Hutchinson estimator. Several variants of this estimator exist, like the unit vector estimator, the normalized

Rayleigh-quotient, and others (Avron and Toledo 2011). In the Hutchinson estimator it is possible to compute confidence regions for the estimators, (using either ideas from Monte Carlo techniques, or the Hoeffding inequality (Bai and Golub 1996; Avron and Toledo 2011)). Other recent developments in trace estimation techniques include algorithms based on Krylov subspaces. These methods can be used to approximate the trace of a matrix function for a low-rank approximation of the input matrix (Chen and Hallman 2023). The algorithm XTrace (Epperly, Tropp, and Webber 2024) achieves an efficient estimator by carefully selecting the samples used to estimate the trace, following an “exchangeability principle”. An implication of this idea, which is leveraged in their work, is that an estimator should be a symmetric function of the samples. A recent work (Meyer et al. 2021) shows how to estimate the trace of a PSD matrix with relative error in only $O(1/\epsilon)$ access to an oracle that gives matrix-vector products (see also (Persson, Cortinovis, and Kressner 2022)). Merging ideas from Hutch++ and techniques such as the Nyström approximation (Li, Kwok, and Lü 2010; Nakatsukasa 2020), it is possible (Saibaba, Alexanderian, and Ipsen 2017; Persson and Kressner 2023) to approximate matrix functions under low-rank assumptions. For lower bounds in this model, the interested reader is referred to (Jiang et al. 2021; Roosta-Khorasani and Ascher 2015), while a recent review of classical algorithms for spectral sums using Krylov subspaces can be found in Ref. (Chen, Trogdon, and Ubaru 2022).

1.1 Main results

We present quantum algorithms to estimate the log-determinant, the von Neumann entropy of a graph, the trace of inverse, and the Schatten p -norm. As applications in spectral graph theory, we show how to estimate the number of triangles in a graph, estimate the effective resistance between nodes of an electric network, and count the spanning trees in a graph: all problems relevant for the practitioner in machine learning and artificial intelligence.

Our algorithms work in the block-encoding model. An (α, ϵ) -block-encoding of a matrix A is a unitary that has a matrix A'/α in its top-left corner satisfying a condition on the distance induced by the spectral norm: $\|A - \alpha A'\| \leq \epsilon$ (see definition 6). As in classical algorithms, we leverage the idea that the trace of a square matrix is equal to the sum of its eigenvalues. Thus, we perform trace estimation of a suitably created matrix which we manipulate with quantum singular value transformation techniques ((Chakraborty, Gilyén, and Jeffery 2019; Gilyén et al. 2019; Low and Chuang 2019; Chakraborty, Morolia, and Peduri 2022; Tang and Tian 2024; Motlagh and Wiebe 2023)). Quantum algorithms for computing the (normalized) trace are well-studied (e.g. in the *one clean qubit* model of quantum computation — see (Shor and Jordan 2008; Cade and Montanaro 2018) for instance), and we generalize them the block-encoding setting, which we do by formalizing previous algorithms (Chowdhury, Somma, and Subaşı 2021; Subramanian and Hsieh 2021). In particular we use subroutines to estimate the trace with absolute and relative error, and subroutines to estimate the trace of $A^T A$ with relative er-

ror (Van Apeldoorn et al. 2020). We add to the list of useful quantum subroutines to manipulate block-encodings a way to compute the product of k preamplified block-encodings (see full version of the manuscript).

We denote with κ the condition number of the matrix A , and with α a quantity upper bounded by the Frobenius norm of the matrix (both are precisely defined in Section 1.3). The notation $\tilde{O}(\cdot)$ hides polylogarithmic factors from the asymptotic complexity (reported in the main manuscript and the appendix). For all the algorithms in Section 2, we show exact bounds on the maximum tolerable error of the block-encoding that we receive as input for our algorithms. Specifically, for an (α, ϵ) -block-encoding and a chosen ϵ , we determine the maximum ϵ allowable to ensure that our algorithm produces an estimate of the desired quantity with an error bounded by ϵ . While it is not necessary to use a quantum arithmetic model (full version of the manuscript), doing so simplifies the analysis of certain steps in the algorithm returning relative error estimates. Importantly, we recall that it is possible to build block-encodings with a *quantum memory device* or using the so called *query access to a matrix*, which works well with sparse matrices. While the cost of our algorithm is expressed in terms of use of block-encodings, size and depth of the circuit can be obtained by multiplying the cost of the queries to the block-encodings with the size and depth of the circuit implementing the block-encoding.

Quantum algorithm for log-determinants. The importance of algorithms for the (logarithm of the) determinant of an PSD matrix A cannot be stressed enough. The log-determinant is defined as $\log\det(A) := \sum_{j=1}^n \log \lambda_j$. If the matrix is positive semi-definite, the definition works by considering only non-zero eigenvalues. The log-determinant is used in machine learning and other computational sciences (Rue and Held 2005; Rasmussen and Nickisch 2010; Cichocki, Cruces, and Amari 2015; Kerenidis, Luongo, and Prakash 2020; Reynolds 2009). There are many classical algorithms for this problem (Barry and Pace 1999; Pace and LeSage 2004; Boutsidis et al. 2017; Avron and Toledo 2011; Saibaba, Alexanderian, and Ipsen 2017; Aune, Simpson, and Eidsvik 2014; Han, Malioutov, and Shin 2015; Zhang and Leithead 2007).

Theorem 2. *Let U_A be an α -block-encoding of an PSD matrix $A \in \mathbb{R}^{n \times n}$. There is a quantum algorithm that returns an estimate $\overline{\log\det(A)}$ which is with ϵ -relative error of $\log\det(A)$ using $\tilde{O}\left(\frac{\alpha\kappa}{\epsilon}\right)$ queries to U_A .*

While we discuss comparison with other works in the next section, we briefly mention that we give a quadratic speedup in the condition number with respect to the previous quantum algorithms for the log-determinant (Zhao et al. 2019, 2021). Interestingly, our algorithm also achieves faster performance than a recent approach (Giovannetti, Lloyd, and Maccone 2025), which attracted considerable attention in the quantum algorithms community. Their work has a runtime of $\tilde{O}\left(\frac{\kappa s^2}{\epsilon^3}\right)$, where s is the row sparsity of the (Hermitian) matrix. As mentioned, our algorithm works in the *more general* block-encoding model, subsuming the access model used in (Giovannetti, Lloyd, and Maccone 2025).

In our framework, the block-encoding normalization factor α reduces to s using query access to the matrix, which is common on matrices derived from physical problems (the sparse-matrix assumptions). Therefore, specializing our result yields a runtime of $\tilde{O}\left(\frac{s\kappa}{\epsilon}\right)$, delivering a quadratic improvement in the condition-number and cubic improvement in the approximation error. Recent work shows that the non-normalized estimation of the log-determinant with inverse polynomial accuracy is BQP-hard, and becomes PP-complete with inverse exponential accuracy (Edenhofer, Hasegawa, and Gall 2025).

Quantum algorithms for Schatten norm. Let $A \in \mathbb{R}^{m \times n}$ be a matrix with singular values $\{\sigma_i\}_{i=1}^{\min(m,n)}$. The Schatten p -norm of A , for $p \in \mathbb{N}^+$ is defined as $\|A\|_p := \left(\sum_{i=1}^{\min(m,n)} \sigma_i^p\right)^{1/p}$. There are many applications for the algorithms computing the Schatten p -norms, for example in completion problems, theoretical chemistry, image processing, and many others (Netrapalli et al. 2014; Gutman 2001; Lu and Negahban 2015; Nie, Huang, and Ding 2012; Majumdar and Ward 2011; Xie et al. 2016). We propose two different results for estimating the Schatten p -norm of a matrix. We also report in the full manuscript the results of numerical experiments showing the asymptotic scaling of certain parameters on real-world datasets. These numerical experiments show that in some instances, and certain regimes of parameters, of real-world datasets, the parameter that governs the runtime of the quantum algorithm scales more favorably than the worst-case analysis.

Theorem 3. *Let U_A be an α -block-encoding of a matrix $A \in \mathbb{R}^{m \times n}$ with $n \leq m$. There is a quantum algorithm that returns an estimate $\|A\|_p$ which is with ϵ -relative error of $\|A\|_p$ using U_A for $\tilde{O}\left(\frac{p\sqrt{n}(\sqrt{2}\|A\|)^{p/2}}{\epsilon\|A\|_p^{p/2}}\left(\frac{\alpha}{\|A\|}\right)\right)$ times if p is even, or $\tilde{O}\left(\frac{\sqrt{n}\alpha^{1.5}(\sqrt{2})^{p/2}(\|A\|)^{p/2-1}}{\epsilon\|A\|_p^{p/2}}(p+\kappa)\right)$ times if p is odd.*

Our results improve polynomially over previous quantum algorithms (Montanaro 2015). For large values of p , we have another algorithm that uses an efficient polynomial approximation of monomials in $[-1, 1]$ that allows to have a quadratic improvement in p . For this, we need to assume a square matrix and a bound on the spectral norm of A . The exponential dependence in p follows from the subroutines we use for block-encoding amplification, and thus can potentially be removed by improving said subroutines.

Quantum algorithm for von Neumann entropy of graphs. We study an algorithm to estimate the von Neumann entropy of a density matrix. Contrary to many other quantum algorithms for this problem, we do not require access to a purification that allows the creation of a block-encoding of the density matrix. This is especially relevant for problem where the density matrix is obtained from graphs, like the Laplacian of a graph. For a graph $G = (V, E)$ a graph Laplacian \mathcal{L} is a PSD matrix defined as $\mathcal{L} = \Delta(G) - A(G)$. We can associate to G a density matrix $\rho_G = \mathcal{L}/\text{Tr}[\mathcal{L}]$. Then, the von Neumann entropy of the graph G is defined as

$H(G) := H(\rho_G) = -\text{Tr}[\rho_G \log \rho_G]$. In the following $s(G)$ is the number of edges in the graph G , i.e. $|E|$. The von Neumann entropy is used in financial data analysis, genomics, complex network analysis, and pattern recognition (Caraiani 2014; Banerjee and Pal 2014; Alter, Brown, and Botstein 2000; Minello, Rossi, and Torsello 2018; Han et al. 2012; Passerini and Severini 2008). There is a vast literature of quantum algorithms for von Neumann entropies of density matrices ρ , but they work in a model where one has access to a purification of the density matrix, which results in a 1-block encoding of ρ (Subramanian and Hsieh 2021; Li and Wu 2018; Gilyén and Li 2020; Wang et al. 2022; Gur, Hsieh, and Subramanian 2021).

Theorem 4. *Let U_A be an α -block-encoding of an PSD matrix $A \in \mathbb{R}^{n \times n}$. There is a quantum algorithm that returns an estimate $\overline{H}(A)$ which is an estimate of $H(A)$ with ϵ -absolute error using $\tilde{O}\left(\frac{n\alpha\kappa}{\epsilon s(G)}\right)$ queries to U_A .*

Quantum algorithms for trace of the inverse. For a square matrix A , the trace of the inverse is $I(A) := \text{Tr}[A^{-1}] = \sum_{i=1}^n \lambda_i^{-1}$. The trace of the inverse finds application lattice quantum chromodynamics, generalized cross validation, and uncertainty quantification (Dashti and Stuart 2011; Golub and Von Matt 1997; Stathopoulos, Laeuchli, and Orginos 2013). To our knowledge, there are only previous works for the trace of the inverse for classical algorithms (Han et al. 2017; Ubaru, Chen, and Saad 2017; Wu et al. 2016), while there are no previous quantum algorithms.

Theorem 5. *Let U_A be an α -block-encoding of an PSD matrix $A \in \mathbb{R}^{n \times n}$. There is a quantum algorithm that returns an estimate $\overline{I}(A)$ which is an estimate of $I(A)$ with ϵ -relative error using $O\left(\frac{\alpha^2\kappa^2}{\epsilon}\right)$ queries to U_A .*

1.2 Applications in spectral graph theory

Our algorithms have extensive applications, and we illustrate their versatility by discussing multiple applications across three distinct topics in spectral graph theory, whose proof and discussion can be found in appendix 2. First, the techniques used for computing the Schatten p -norms can be used for counting the number of triangles in a graph. We propose two algorithms. Denoting $\Delta(G)$ the number of triangles in G , the runtime of the first algorithm is $O\left(\frac{\alpha n}{\epsilon \Delta(G)}\right)$, and the runtime of the second algorithm is $O\left(\frac{\alpha \sqrt{n\kappa}}{\epsilon \sqrt{\Delta(G)}}\right)$. Algorithms for counting triangles find many applications in network analysis (Suri and Vassilvitskii 2011; Easley, Kleinberg et al. 2012). Second, we show how to give a relative error estimate of the number of spanning trees in a graph G , assuming block-encoding access to the graph Laplacian. Our quantum algorithm makes $\tilde{O}(n\alpha\kappa/\epsilon)$ calls to the block-encoding of the Laplacian matrix of A . There are many applications in machine learning (Meila and Jordan 2000), genomics, and network theory (Wang, Zhang, and Zhuang 2014; Kirby et al. 2016). Third, we show how to compute the effective resistance between two nodes in a graph using the block-encoding of (a modified) graph Laplacian \mathcal{L} for $\tilde{O}(n\alpha\kappa/\epsilon)$

times. Applications include graph clustering, graph sparsification and analysis, graph neural networks, and many others (Alev et al. 2017; Fortunato 2010; Ahmad et al. 2021; Katz 1953; Spielman and Srivastava 2008).

1.3 Preliminaries and notation

We assume a basic understanding of quantum computing, and we recommend (Nielsen and Chuang 2010) for a comprehensive introduction to the subject. For a matrix M we denote its transpose-conjugate as M^\dagger . For a matrix $A = (a_{ij})_{n \times n} \in \mathbb{R}^{n \times n}$ we write $A = U\Sigma V^\dagger = \sum_{i=1}^n \sigma_i u_i v_i^\dagger$ as its singular value decomposition, where σ_i are the singular values and u_i, v_i its left and right singular vectors respectively. We use λ_i to denote the eigenvalues of a matrix. We assume that the singular values (and eigenvalues) are sorted such that σ_1 is the biggest and σ_n is the smallest. A Hermitian square matrix $A \in \mathbb{C}^{n \times n}$ is said to be positive semi-definite (PSD) i.e. $A \geq 0$ if $x^\dagger A x \geq 0$ for all $x \in \mathbb{C}^n$. With $\|A\|_0$ we denote the number of non-zero elements of the matrix A , with $\|A\| = \sigma_1$ the biggest singular value of A , and with $\|A\|_F = \sqrt{\sum_{i,j=1}^n |a_{ij}|^2}$ its Frobenius norm. With s_r (s_c) we denote the row(column)-sparsity, that is, the maximum number of non-zero entries of the rows (columns). The sparsity s of a matrix A is defined as the number of non-zero entries (i.e. $s := \|A\|_0$). If the matrix is the adjacency matrix of a graph G , we write $s(G)$. For a matrix $L \in \mathbb{R}^{n \times n}$ and $A \subset [n]$ we denote with $L(A)$ the matrix obtained from L by removing the columns and rows indexed by A . With $\kappa(A)$ we denote the condition number of A , that is, the ratio between the biggest and the smallest non-zero singular values. When it is clear from the context, we will simply denote $\kappa(A)$ as κ . In this work we use the quantum singular value transformation, a technique comprehensively studied in (Chakraborty, Gilyén, and Jeffery 2019; Gilyén et al. 2019; Low and Chuang 2019; Chakraborty, Morolia, and Peduri 2022; Tang and Tian 2024; Motlagh and Wiebe 2023). In the full manuscript we report polynomial approximations of useful functions. The theorem that we are using for singular value transformation below requires classical computation for obtaining a description of a quantum circuit, which depends on the degree of the polynomial and the precision used to compute certain angles. As this procedure is very fast on a classical desktop computer, resulting in negligible error, we will consider it only in the proof of theorem 2. The interested reader is referred to (Chao et al. 2020; Dong et al. 2021) for more information.

2 Quantum Algorithms for Spectral Sums

In the quantum setting, we compute a spectral sum of a matrix A using a methodology analogous to the classical case, but with different components. We work in a *quantum oracle model*, where our quantum computer has access to a unitary that encodes the input of the problem. In our case, this unitary is a block-encoding of the matrix A .

Definition 6 (Block-encoding (Gilyén et al. 2019; Low and Chuang 2019)). *Let $A \in \mathbb{R}^{2^m \times 2^m}$, $\alpha, \epsilon \in \mathbb{R}^+$ and $q \in \mathbb{N}$.*

The $(m+q)$ -qubit unitary U_A is an (α, q, ϵ) -encoding of A , if $\|A - \alpha(|0\rangle^{\otimes q} \otimes I)U_A(|0\rangle^{\otimes q} \otimes I)\| \leq \epsilon$.

In this setting, we measure the asymptotic complexity of our algorithms in calls to the unitaries of the block-encoding. Creating block-encodings is quite straightforward in the *QRAM model* (i.e. when we have access to a quantum random access memory via a QMD: a quantum memory device (Allcock et al. 2023)) or in the *sparse access model*. We discuss both cases in the full manuscript, where we show how to work with non-square matrices. In those setting, it is possible to build an $(\alpha, a, 0)$ -encoding of a matrix, for small a , and where α can be a matrix function (folklore in literature). To build a block-encoding in both models, we have to perform a classical preprocessing that requires linear time (with some polylogarithmic overhead) in the size of the matrices. This time is needed to build a circuit for sparse access, or a data structure that will be stored in the quantum memory. Build a block-encoding for $A \in \mathbb{R}^{2^m \times 2^m}$ requires $O(m)$ queries to a quantum memory device. The depth of those circuits is usually of $O(m)$ (Jaques and Rattew 2023; Chakraborty, Gilyén, and Jeffery 2019). Using a quantum computer with a QMD, the runtime of the computation is proportional to the depth of the circuits, which for the algorithms presented in this work, is proportional to the number of calls to the QMD. Hence, we use the query complexity of the block-encoded matrix (i.e. number of usages of U_A) as a proxy for the runtime of the computation. A precise definition of a quantum computer with access to quantum memory can be found in (Allcock et al. 2023). We assume quantum access to sub-normalized matrices, i.e. where $\|A\| < 1$, say $1/e$. This assumption can be easily satisfied by dividing the matrix by a multiple of its biggest singular value, before creating quantum access to the matrix. For instance, one can use (Kerenidis and Prakash 2020, Algorithm 4.3, Proposition 4.8) to estimate $\|A\|$ with relative error, or $\kappa(A)$ with additive error in time $O(\frac{\|A\|_F \log(1/\epsilon)}{\|A\|_\epsilon})$. We assume that this procedure is done immediately after having built quantum access so the cost of this operation can be subsumed into the cost of creating quantum access to A . In the quantum algorithm we compute a polynomial approximation of the function of choice, and perform quantum singular value transformation on the block-encoding, (for example, using (Gilyén et al. 2019, Theorem 56)). In particular, we will build a block-encoding of $P(A)$, where $P(x)$ is a polynomial approximation a function of choice f by using singular value transformation techniques (Chakraborty, Gilyén, and Jeffery 2019; Gilyén et al. 2019; Low and Chuang 2019; Chakraborty, Morolia, and Peduri 2022; Tang and Tian 2024; Motlagh and Wiebe 2023). In the full manuscript we report some polynomial approximation for the functions we need, along with other algorithms for manipulating the spectrum of a matrix. Lastly, we use a quantum algorithm to estimate the trace of a block-encoded matrix.

Lemma 7 (Quantum trace estimation). *Let $\epsilon \in (0, 1)$. Let U be an (α, q, δ) block-encoding of $A \in \mathbb{C}^{n \times n}$. There is a quantum algorithm that returns $\overline{\text{Tr}[A]}$ with probability at least $2/3$ such that $|\text{Tr}[A] - \overline{\text{Tr}[A]}| \leq n\epsilon$ using $O(\frac{\alpha}{\epsilon})$ calls*

of U and U^\dagger , if $\delta \leq \epsilon/2$.

The proof of this statement can be found in the full manuscript, along with an algorithm for estimating a trace with relative error and small failure probability, which we formalize in the language of block-encodings from (Chowdhury, Somma, and Subaşı 2021; Subramanian and Hsieh 2021). There are other quantum algorithms related to the problem of trace estimation. For example, in (Quek, Kaur, and Wilde 2024) they estimate the trace of the product of m different density matrices (i.e. $\text{Tr}[\rho_1 \dots \rho_m]$), in a computational model where all the ρ_i matrices are available at the beginning of the computation. The quantum circuit has constant depth, but at the expense of running the circuit for $O(\frac{1}{\epsilon^2})$ times. A recent algorithm for multivariate trace estimation (Yosef et al. 2024) propose high-level set of sub-routines to write new quantum algorithms for linear algebra (quantum Matrix State Linear Algebra). In (Subramanian and Hsieh 2021), they revisited and simplified the previous algorithm by demonstrating how to keep the failure probability of every iteration constant, (but as a function of a lower bound of the trace). The approach of maintaining a constant failure probability for every iteration works well with the quantum arithmetic model, as there exists an inherent “smallest trace” that can be estimated on a quantum computer when the numbers are represented in an explicit arithmetic model. We discuss this in the appendix.

2.1 Log-determinant

Definition 8 (Log-determinant of an PSD matrix A). *Let $A \in \mathbb{R}^{n \times n}$ be a PSD matrix, and $b \in \mathbb{N}$. Let $\lambda_1, \dots, \lambda_n$ be the eigenvalues of A . Then the log-determinant of A is defined by $\text{logdet}(A) := \log_2 \det(A) = \sum_{j=1}^n \log \lambda_j$.*

An algorithm for estimating the log-determinant can be used to estimate the determinant (and vice versa). Observe that, while the determinant of a PSD matrix is always positive (because the eigenvalues are all positive), the log-determinant of a non-PSD matrix can be either positive or negative. Under the assumption that the singular values of the matrix lie in the interval $(0, 1]$, the log-determinant is always a negative quantity. In case they are not, we can always consider a rescaled matrix $A' = A/\beta$ where $\beta \geq \|A\|$. Then, we can recover the log-determinant of A as $\text{logdet}(A) = n \log(\beta) + \text{logdet}(A')$. The log-determinant is used in many different research areas. The log-determinant is used for the calculation of the marginal log-likelihood of non-parametric kernel-based methods (Dong et al. 2017). For example, computing a log-determinant is necessary when training Gaussian processes and Gaussian graphical models (Rue and Held 2005; Rasmussen and Nickisch 2010), and in tasks such as model selection, and model inference. The log-determinant also appears in other machine learning problems, as the computation of Bregman divergences (Cichocki, Cruces, and Amari 2015), and quantum and classical algorithms for fitting Gaussian mixtures with Expectation-Maximization algorithms (Kerenidis, Luongo, and Prakash 2020; Reynolds 2009). There are many popular algorithms for the estimation of the log-determinant. For many years, most of the software used algorithms based on

the Cholesky decomposition of a matrix. One of the first stochastic algorithms proposed the idea of using a Taylor expansion of the logarithm, along with a stochastic trace estimator subroutine (Barry and Pace 1999). Few years later, a new algorithm used the Chebyshev approximation of the logarithm function and an exact trace calculation algorithm (Pace and LeSage 2004). More recently (Boutsidis et al. 2017), improved the estimation of the log-determinant using again a Taylor expansion of the logarithm function, but using some improved results (Avron and Toledo 2011) for stochastic trace estimation. Under low-rank assumptions, it is possible to study algorithms based on subspace-iteration and access the matrix only through matrix-vector products (Saibaba, Alexanderian, and Ipsen 2017). Using iterative methods over Krylov subspaces and dynamic choice of the probing vectors it is possible to improve experimentally the performances of the estimator (Aune, Simpson, and Eidsvik 2014). To our knowledge, the best asymptotic complexity for classical algorithms (Han, Malioutov, and Shin 2015) is of $O(\frac{\sqrt{\kappa} \|A\|_0}{\epsilon^2} \log(\frac{\kappa}{\delta}))$, where ϵ is the approximation error and δ is the failure probability. This work combines stochastic trace estimators and Chebyshev approximation techniques. Later on, early results featured some error compensation schemes for improving the accuracy (Zhang and Leithead 2007) and thus reducing the constant factors in the runtime. A quantum algorithm for estimating the determinant exists (Zhao et al. 2021), whose complexity can be measured in the number of samples to a quantum algorithm, and has a quadratic dependence on the precision of the results. There is a quantum algorithm for estimating the log-determinant (Zhao et al. 2019), with a complexity of $\tilde{O}(\frac{\kappa^2 \|A\|_0}{\epsilon})$. As mentioned before, the work of (Giovannetti, Lloyd, and Maccone 2025) achieves a runtime of $\tilde{O}(\frac{\kappa s^2}{\epsilon^3})$, where s is the row sparsity of the (Hermitian) matrix. Interestingly, a surge of interest in this topic Recent work shows that the non-normalized estimation of the log-determinant with inverse polynomial accuracy is BQP-hard, and becomes PP-complete with inverse exponential accuracy (Edenhofer, Hasegawa, and Gall 2025).

Theorem 9 (Log-determinant). *Let $\epsilon \in (0, 1/6)$, $\delta \in (0, 1)$, and let U_A be an (α, q, ϵ_1) -encoding of a PSD matrix $A \in \mathbb{R}^{n \times n}$ with $\|A\| < 1/e$ and $\epsilon_1 \leq \frac{\epsilon^2}{\alpha [64\kappa \log(\alpha\kappa) \log(\frac{\log(\alpha\kappa)}{\epsilon})]^2}$. There is an algorithm that returns an estimate $\overline{\text{logdet}(A)}$ such that $|\overline{\text{logdet}(A)} - \log(\det(A))| \leq \epsilon |\ln(\det(A))|$ with probability at least $1 - \delta$, using U_A and U_A^\dagger for $O(\frac{\alpha\kappa}{\epsilon} \log(\frac{\log(\kappa\alpha)}{\epsilon}) \log(1/\delta))$ times.*

In the case where $\|A\| > 1$ we can scale the matrix as described before. With the algorithm we can approximate the log-determinant up to precision $n\epsilon$ (by skipping the observation that $n \leq |\text{logdet}(A)|$). In this case, it is not easy to transform it into a relative error because of the mixed-sign of the logarithm of the singular values.

2.2 Schatten norm

Definition 10 (Schatten p -norm (Kittaneh 1985)). *Let $A \in \mathbb{R}^{m \times n}$ be a matrix with singular values $\{\sigma_i\}_{i=1}^{\min(m,n)}$. Let $p \in \mathbb{N}^+$, then the Schatten p -norm $\|A\|_p$ is defined as:*

$$\|A\|_p := \left(\sum_{i=1}^{\min(m,n)} \sigma_i^p \right)^{1/p}.$$

This norm can be defined also for non-square matrices. When $p = 1$, it is also known as the nuclear or trace norm, and for $p = 2$ the Schatten 2-norm is the Frobenius norm. Schatten p -norms are often used in matrix completion algorithms (Netrapalli et al. 2014), in theoretical chemistry (Gutman 2001), rank aggregation and collaborative ranking (Lu and Negahban 2015), convex relaxations for rank-constrained optimization (Nie, Huang, and Ding 2012), and in image processing (Majumdar and Ward 2011; Xie et al. 2016). There are quantum algorithms for estimating the Schatten p -norms. A recent paper (Cade and Montanaro 2018) focuses on the one clean qubit model, and they assume that both p and $1/\epsilon$ are of order $O(\text{poly log } n)$. Another quantum algorithm for estimating Schatten p -norms (Zhao et al. 2021) is similar in spirit to the quantum algorithm for the log-determinant of (Zhao et al. 2019), i.e., they sample from the uniform distribution of singular values and perform the rest of the computation classically. Because of the sampling technique employed there, the algorithm has a dependence on the precision of $O(\epsilon^{-3})$, and further depends quadratically on the Lipschitz constant K_p of the function x^p in the domain $[0, \sigma_{\max}]$. Furthermore, their algorithms assume that the spectrum of the matrix follows a uniform distribution, i.e. that the singular values are all $\Theta(1)$, which is not the case for low-rank matrices that are obtained from real-world datasets (Udell and Townsend 2019). The present work offers a polynomial speedup with respect to those results. To our knowledge, the fastest classical algorithm for estimating the Schatten- p norm (Han et al. 2017) has complexity $\tilde{O}(\|A\|_0 p \kappa / \epsilon^2)$. Note that a fast quantum algorithm for estimating the Schatten 2-norm is a trace estimation of product of matrices, that can be found in standard literature. In theorem 11 we consider the case for non-square, non-normalized matrices. In the full manuscript we propose another quantum algorithm with a further quadratic speedup in p , where further assume that $\|A\| \leq 1$. In the full manuscript we numerically verify the scaling of the factor $\rho = \frac{(\sqrt{2}\|A\|)^{p/2}}{\|A\|_p^{p/2}}$, appearing in the runtimes. To prove the following, we extensively use results on composition on block-encodings. The full manuscript has a new theorem for obtaining the product of k different amplified block-encodings.

Theorem 11 (Schatten p -norms). *Let $\epsilon, \delta \geq 0$, $p \in \mathbb{N}^+$. Assume U_A is an (α, a, ϵ) -encoding of a matrix $A \in \mathbb{R}^{m \times n}$ with $n \leq m$. There is a quantum algorithm that computes an ϵ -relative approximation of the Schatten p -norm of A using U_A and U_A^\dagger for $\tilde{O}\left(\frac{p\sqrt{n}(\sqrt{2}\|A\|)^{p/2}}{\epsilon\|A\|_p^{p/2}} \left(\frac{\alpha}{\|A\|}\right)\right)$ times if p is even and $\epsilon \leq \frac{\epsilon\|A\|_p^{p/2}}{4p(\sqrt{2})^{p/2-1}\sqrt{n}\|A\|^{p/2-1}}$, or $\tilde{O}\left(\frac{\sqrt{n}\alpha^{1.5}(\sqrt{2})^{p/2}(\|A\|)^{p/2-1}}{\epsilon\|A\|_p^{p/2}} \left(\frac{p}{2} + \kappa\right)\right)$ times if p is odd and*

$$\epsilon \leq \frac{\epsilon\|A\|_p^{\frac{p-1}{2}}}{8\sqrt{n} \min(2\alpha^{0.5}, (\sqrt{2}\|A\|)^{(p-1)/2})}. \text{ The } \tilde{O} \text{ notation hides factor polylogarithmic in } p, \frac{2^{p/2}\|A\|_p^p}{\|A\|_p^{p/2}}, \sqrt{n}, \alpha, \text{ and } 1/\epsilon.$$

This algorithm is linear in p , as the classical one. In the following, thanks a polynomial approximation of monomials, we can obtain a further quadratic advantage for computing a Schatten p -norms. We formulate the algorithm for working with square matrices with $\|A\| \leq 1$, which allows using a theorem computing a block-encoding of positive powers of Hermitian matrices. This will also save a further factor of $\alpha^{0.5}$ in the final runtime. We believe the algorithm can be generalized to non-square matrices (for example using (Gilyén et al. 2019)). For simplicity, we assume a 0-error block-encoding as input. In the a new algorithm, we will use $\tilde{O}(\sqrt{p})$ calls to U_A , where the asymptotic notation hides constant and polylogarithmic factors coming from a polynomial approximation of monomials, and the computation of the description of the circuits given by quantum singular value transformations. For small and even values of p it might be convenient to use the previous algorithm, which is using exactly $p/2$ calls to U_A . The statement and proof are reported in the full manuscript.

2.3 von Neumann entropy of graphs

The spectral sum for the function $f(x) = -x \log x$ (where we take $0 \log 0 = 0$ by convention) can be linked to the von Neumann entropy of a quantum state: the quantum generalization of the classical Shannon entropy. It is possible to study the notion of quantum entropy of a network and other combinatorial structures, see for example (Braunstein, Ghosh, and Severini 2006). Recall that a density matrix is defined as a PSD matrix with unit trace.

Definition 12 (von Neumann entropy of a density matrix). *For a density matrix ρ with eigenvalue decomposition $\rho = \sum_{i=1}^n \lambda_i |\psi_i\rangle \langle \psi_i| \in \mathbb{R}^{n \times n}$, the von Neumann entropy of ρ is defined as: $H(\rho) := -\text{Tr}[\rho \log \rho] = -\sum_{i=1}^n \lambda_i \log \lambda_i$.*

We can associate the notion of entropy of a simple, undirected, and connected graph $G = (V, E)$ with n vertices. Let $A(G)$ be the adjacency matrix of the graph (i.e. $A(G)_{ij} = 1$ if $(i, j) \in E$), and $\Delta(G)$ be the so-called degree matrix of G : a diagonal matrix where $\Delta(G)_{ii} = d(i)$, where $d(i)$ is the number of neighbors of the node i . The *combinatorial Laplacian matrix* of G is defined as $\mathcal{L}(G) := \Delta(G) - A(G)$. It is possible to see that the matrix is PSD (see the Gershgorin disk theorem (Horn and Johnson 2012), or observe that $\mathcal{L} = M^T M$ where M is the incidence matrix of G) and that the rank of the matrix is $n - k$, where k is the number of connected components of the graph (Bapat 1996). To obtain a density matrix from a graph, we define $\rho_G := \frac{\mathcal{L}}{\text{Tr}[\mathcal{L}]} = \frac{\mathcal{L}}{s(G)}$ where we noted that $s(G) = \text{Tr}[\mathcal{L}]$. The entropy $H(G)$ of a graph G is then defined as the entropy of $H(\rho_G)$.

Definition 13 (von Neumann entropy of a graph). *Let $G = (V, E)$ be a graph with Laplacian $\mathcal{L} = \Delta(G) - A(G) = \sum_{i=0}^{n-1} \nu_i u_i u_i^T$ and associated density matrix $\rho_G = \mathcal{L}/\text{Tr}[\mathcal{L}] = \sum_{i=0}^{n-1} \lambda_i u_i u_i^T$. The von Neumann en-*

entropy of the graph G is defined as $H(G) := H(\rho_G) = -\text{Tr}[\rho_G \log \rho_G] = -\sum_{i=1}^n \lambda_i \log \lambda_i$.

Observe that the entropy of G can be rewritten as $H(G) = -s(G)^{-1} \sum_i \nu_i \log \nu_i + \log(s(G))$. The von Neumann entropy of a quantum state quantifies the amount of entanglement contained in a bipartite quantum system (Preskill 1998; Bengtsson and Życzkowski 2017). In applications, the von Neumann entropy is also important in feature selection (Banerjee and Pal 2014), financial data analysis (Caraiani 2014), and genomic data (Alter, Brown, and Botstein 2000). In graph theory, the von Neumann entropy is a spectral measure that has applications in complex network analysis and pattern recognition (Minello, Rossi, and Torsello 2018; Han et al. 2012; Passerini and Severini 2008). Three randomized classical algorithms were given in (Kontopoulou et al. 2018), where the best classical algorithm is (Kontopoulou et al. 2018, theorem 2) which shows an asymptotic complexity of $\tilde{O}(\|A\|_0 \kappa / \epsilon^2)$ operations. There is a vast literature on quantum algorithms for estimating von Neumann entropies, and the more general Rényi entropies (Subramanian and Hsieh 2021; Li and Wu 2018; Gilyén and Li 2020). In (Gilyén and Li 2020), they show how to estimate with additive error using $\tilde{O}(\frac{n}{\epsilon^{1.5}})$ queries to the oracle giving the purification. In (Wang et al. 2022) the algorithms assume quantum access to a purification of a density matrix that encodes a sub-normalized operator. In this setting, they can obtain algorithms that do not depend on the condition number of the operator or the size of the system, but only on the rank of the operator, achieving an asymptotic of $\tilde{O}(r^2 / \epsilon^2)$. Always in the purified query model, it is also possible to estimate the entropy with a relative error using $\tilde{O}(n^{\frac{1}{2} + \frac{1+\eta}{2\gamma s}})$ queries, where η is a lower bound on the entropy, and γ is the relative precision (Gur, Hsieh, and Subramanian 2021). Previous works in quantum algorithms for von Neumann entropies assume to have access to a unitary matrix that produces a state that is the purification of a density matrix, as is described in the full manuscript. From a purification is possible to obtain a 1-block encoding of ρ . We can relax this requirement, by assuming a generic α -block-encoding of a (normalized) combinatorial Laplacian $L := \mathcal{L} / \|\mathcal{L}\|$. When working using with graphs in the sparse access model we have quantum access to a matrix whose spectral norm is usually greater than 1. We can always estimate it, and re-create quantum access to the scaled matrix during the preprocessing.

Theorem 14 (von Neumann entropy). *Let G be a graph with graph Laplacian \mathcal{L} , for a $s(G)$ known. Let $\epsilon \in (0, 1/2)$, $\delta \in (0, 1)$, and let U_L be a an (α, q, ϵ) -encoding of $L = \mathcal{L} / \|\mathcal{L}\|$ and $\epsilon \leq \frac{s(G)^2 \epsilon^2 \alpha}{(196\sqrt{2}n \|\mathcal{L}\| \kappa \alpha \log(2\kappa\alpha) \log(n \log(2\kappa\alpha/\epsilon)))^2}$. There is an algorithm that returns an ϵ -absolute approximation of $H(G)$ with probability $\geq 1 - \delta$, calling U_L and U_L^\dagger for $O\left(\frac{n \|\mathcal{L}\| \alpha \kappa}{\epsilon s(G)} \log^2\left(\frac{n \|\mathcal{L}\| \log(2\kappa\alpha)}{\epsilon}\right) \log\left(\frac{1}{\delta}\right)\right)$ times.*

2.4 Matrix inverse

Definition 15 (Trace of inverse). *For a matrix $A \in \mathbb{R}^{n \times n}$ with non-zero eigenvalues $\{\lambda_i\}_{i=1}^n$, the trace of the inverse of A is defined as: $I(A) := \text{Tr}[A^{-1}] = \sum_{i=1}^n \lambda_i^{-1}$.*

Note that for any nonzero α , we have $I(\alpha A) = I(A)/\alpha$. Thus, if we can approximate the trace of a matrix inverse of a scaled matrix up to relative error ϵ , we can also obtain an ϵ -relative approximation of the original quantity $I(A)$. In most applications, the matrix is usually PSD, but we are not using this assumption in our algorithm. Computing this quantity has wide applications, especially in the study of lattice quantum chromodynamics (Stathopoulos, Laeuchli, and Orginos 2013), generalized cross validation (Golub and Von Matt 1997), and uncertainty quantification (Dashti and Stuart 2011). To our knowledge, the best known classical algorithm (Han et al. 2017) to solve this problem has complexity $\tilde{O}(\|A\|_0 \sqrt{\kappa} / \epsilon^2)$, and is using Chebychev approximation and stochastic trace estimation algorithms. Another classical algorithm (Ubaru, Chen, and Saad 2017), which uses stochastic Lanczos quadrature, has a similar runtime. Recent work (Wu et al. 2016) leverages preconditioners and ideas to approximate the diagonal of the inverse of the matrix using the diagonal of some approximate inverse that can be computed inexpensively. Furthermore, it uses dynamical techniques to control the error of the estimation. However, it lacks a theoretical analysis of the runtime. To our knowledge, there are no previous quantum algorithms for this problem.

Theorem 16 (Trace of the inverse). *Let $\epsilon \in (0, 1/2)$, $\delta \in (0, 1)$, and let U_A be a an (α, q, ϵ_1) -encoding of a PSD matrix $A \in \mathbb{R}^{n \times n}$ with $\|A\| \leq 1$ and $\epsilon_1 \leq \frac{1}{\alpha^3} \left(\frac{3\epsilon}{128\kappa^2 \log(\frac{\alpha\kappa}{\epsilon})}\right)^2$. There is an algorithm that returns an ϵ -relative approximation of $I(A)$ with probability at $\geq 1 - \delta$, using U_A and U_A^\dagger for $O\left(\frac{\alpha^2 \kappa^2 \log(\mu\kappa/\epsilon)}{\epsilon} \log(1/\delta)\right)$ times.*

3 Conclusions and Future Works

Some polynomial approximations that are used in classical algorithms (like Gaussian quadrature, the Lanczos method, and the Cauchy integral (Dong et al. 2017; Dorn and Enßlin 2015; Hale, Higham, and Trefethen 2008)) lead to small polynomial dependence on condition number. These polynomial can potentially be used in the quantum setting. Interestingly, some recent classical algorithms that use new stochastic trace estimators (Meyer et al. 2021) have a linear dependence on error, similar to the quantum case. This leaves the question open whether quantum algorithms can beat classical algorithms and be sub-linear in the approximation error. Many classical algorithms also report the variance associated their estimate, which in the quantum setting can potentially be improved using (Cornelissen and Hamoudi 2023). A possible generalization of the quantum algorithm for von Neumann entropy is a quantum algorithm for estimating the relative entropy between two quantum states, which can be used to estimate the number of spanning trees of a graph (Giovannetti and Severini 2013). We leave for future work the study of algorithms for estimating the distance between PSD matrices (using e.g. (Bhatia, Jain, and Lim 2019)), computing the log-determinant of non-square matrices, studying more applications of algorithms for spectral sums, and other algorithms for counting the number of triangles (for example using (Gall and Ng 2022)).

Acknowledgments

We would like to thank, in sparse order, Chris Cade, András Gilyén, Sander Gribling, Yassine Hamoudi, Iordanis Kerenidis, Pablo Rotondo, Ashley Montanaro, Armando Bellante, Varun Narasimhachar, Stephen Piddock, Sathyawageeswar Subramanian, Liron Mor Yosef, Miklos Santha, and Haim Avron for useful discussions and feedback on the manuscript. AL and CS have been supported by QuantERA ERA-NET Cofund in Quantum Technologies implemented within the European Union’s Horizon 2020 Programme (QuantAlgo project). AL has been also supported by ANTR and CS is supported by EPSRC grants EP/L021005/1 and EP/R043957/1. This work was supported by the National Research Foundation, Singapore, and A*STAR under its CQT Bridging Grant. We also acknowledge funding from the Quantum Engineering Programme (QEP 2.0) under grants NRF2021-QEP2-02-P05 and NRF2021-QEP2-02-P01.

References

- Ahmad, T.; Jin, L.; Lin, L.; and Tang, G. 2021. Skeleton-based action recognition using sparse spatio-temporal GCN with edge effective resistance. *Neurocomputing*, 423: 389–398.
- Aho, A. V.; and Hopcroft, J. E. 1974. *The design and analysis of computer algorithms*. Pearson Education India.
- Alev, V. L.; Anari, N.; Lau, L. C.; and Gharan, S. O. 2017. Graph clustering using effective resistance. *arXiv preprint arXiv:1711.06530*.
- Allcock, J.; Bao, J.; Doriguello, J. F.; Luongo, A.; and Santha, M. 2023. Constant-depth circuits for Uniformly Controlled Gates and Boolean functions with application to quantum memory circuits. *arXiv preprint arXiv:2308.08539*.
- Alter, O.; Brown, P. O.; and Botstein, D. 2000. Singular value decomposition for genome-wide expression data processing and modeling. *Proceedings of the National Academy of Sciences*, 97(18): 10101–10106.
- Aune, E.; Simpson, D. P.; and Eidsvik, J. 2014. Parameter estimation in high dimensional Gaussian distributions. *Statistics and Computing*, 24(2): 247–263.
- Avron, H.; and Toledo, S. 2011. Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix. *Journal of the ACM (JACM)*, 58(2): 1–34.
- Bai, Z.; and Golub, G. H. 1996. Bounds for the trace of the inverse and the determinant of symmetric positive definite matrices. *Annals of Numerical Mathematics*, 4: 29–38.
- Banerjee, M.; and Pal, N. R. 2014. Feature selection with SVD entropy: Some modification and extension. *Information Sciences*, 264: 118–134.
- Bapat, R. B. 1996. The Laplacian matrix of a graph. *Mathematics Student-India*, 65(1): 214–223.
- Barry, R. P.; and Pace, R. K. 1999. Monte Carlo estimates of the log determinant of large sparse matrices. *Linear Algebra and its applications*, 289(1-3): 41–54.
- Bengtsson, I.; and Życzkowski, K. 2017. *Geometry of quantum states: an introduction to quantum entanglement*. Cambridge: Cambridge university press.
- Bhatia, R.; Jain, T.; and Lim, Y. 2019. On the Bures–Wasserstein distance between positive definite matrices. *Expositiones Mathematicae*, 37(2): 165–191.
- Boutsidis, C.; Drineas, P.; Kambadur, P.; Kontopoulou, E.-M.; and Zouzias, A. 2017. A randomized algorithm for approximating the log determinant of a symmetric positive definite matrix. *Linear Algebra and its Applications*, 533: 95–117.
- Braunstein, S. L.; Ghosh, S.; and Severini, S. 2006. The Laplacian of a graph as a density matrix: a basic combinatorial approach to separability of mixed states. *Annals of Combinatorics*, 10: 291–317.
- Cade, C.; and Montanaro, A. 2018. The Quantum Complexity of Computing Schatten p-norms. In *13th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.
- Caraiani, P. 2014. The predictive power of singular value decomposition entropy for stock market dynamics. *Physica A: Statistical Mechanics and its Applications*, 393: 571–578.
- Chakraborty, S.; Gilyén, A.; and Jeffery, S. 2019. The power of block-encoded matrix powers: improved regression techniques via faster Hamiltonian simulation. In *Proc. 46th International Colloquium on Automata, Languages, and Programming*, 33:1–33:14.
- Chakraborty, S.; Morolia, A.; and Peduri, A. 2022. Quantum Regularized Least Squares. *arXiv preprint arXiv:2206.13143*.
- Chao, R.; Ding, D.; Gilyen, A.; Huang, C.; and Szegedy, M. 2020. Finding angles for quantum signal processing with machine precision. *arXiv preprint arXiv:2003.02831*.
- Chen, T.; and Hallman, E. 2023. Krylov-aware stochastic trace estimation. *SIAM Journal on Matrix Analysis and Applications*, 44(3): 1218–1244.
- Chen, T.; Trogdon, T.; and Ubaru, S. 2022. Randomized matrix-free quadrature for spectrum and spectral sum approximation. *arXiv preprint arXiv:2204.01941*.
- Chowdhury, A. N.; Somma, R. D.; and Subaşı, Y. 2021. Computing partition functions in the one-clean-qubit model. *Physical Review A*, 103(3): 032422.
- Cichocki, A.; Cruces, S.; and Amari, S.-i. 2015. Log-determinant divergences revisited: Alpha-beta and gamma log-det divergences. *Entropy*, 17(5): 2988–3034.
- Cornelissen, A.; and Hamoudi, Y. 2023. A sublinear-time quantum algorithm for approximating partition functions. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 1245–1264. SIAM.
- Dashti, M.; and Stuart, A. M. 2011. Uncertainty quantification and weak approximation of an elliptic inverse problem. *SIAM Journal on Numerical Analysis*, 49(6): 2524–2542.
- Dong, K.; Eriksson, D.; Nickisch, H.; Bindel, D.; and Wilson, A. G. 2017. Scalable log determinants for Gaussian

- process kernel learning. In *Advances in Neural Information Processing Systems*, 6327–6337.
- Dong, Y.; Meng, X.; Whaley, K. B.; and Lin, L. 2021. Efficient phase-factor evaluation in quantum signal processing. *Physical Review A*, 103(4): 042419.
- Dorn, S.; and Enßlin, T. A. 2015. Stochastic determination of matrix determinants. *Physical Review E*, 92(1): 013302.
- Easley, D.; Kleinberg, J.; et al. 2012. Networks, crowds, and markets: Reasoning about a highly connected world. *Significance*, 9: 43–44.
- Edenhofer, R.; Hasegawa, A.; and Gall, F. L. 2025. Dequantization and Hardness of Spectral Sum Estimation. *arXiv preprint arXiv:2509.20183*.
- Epperly, E. N.; Tropp, J. A.; and Webber, R. J. 2024. Xtrace: Making the most of every sample in stochastic trace estimation. *SIAM Journal on Matrix Analysis and Applications*, 45(1): 1–23.
- Fortunato, S. 2010. Community detection in graphs. *Physics reports*, 486(3-5): 75–174.
- Gall, F. L.; and Ng, I.-I. 2022. Quantum Approximate Counting for Markov Chains and Application to Collision Counting. *arXiv preprint arXiv:2204.02552*.
- Gilyén, A.; and Li, T. 2020. Distributional Property Testing in a Quantum World. In *11th Innovations in Theoretical Computer Science Conference (ITCS 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.
- Gilyén, A.; Su, Y.; Low, G. H.; and Wiebe, N. 2019. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, 193–204.
- Giovannetti, V.; Lloyd, S.; and Maccone, L. 2025. A quantum algorithm for estimating the determinant. *arXiv preprint arXiv:2504.11049*.
- Giovannetti, V.; and Severini, S. 2013. Kirchhoff’s Matrix-Tree Theorem Revisited: Counting Spanning Trees with the Quantum Relative Entropy. *Advances in Network Complexity*, 177–190.
- Golub, G. H.; and Von Matt, U. 1997. Generalized cross-validation for large-scale problems. *Journal of Computational and Graphical Statistics*, 6(1): 1–34.
- Gur, T.; Hsieh, M.-H.; and Subramanian, S. 2021. Sublinear quantum algorithms for estimating von Neumann entropy. *arXiv preprint arXiv:2111.11139*.
- Gutman, I. 2001. The energy of a graph: old and new results, Algebraic Combinatorics and Applications. *Betten, A., Kohner, A., Laue, R., Wassermann, A.(eds.)*, 196–211.
- Hale, N.; Higham, N. J.; and Trefethen, L. N. 2008. Computing A^α , $\log(A)$, and related matrix functions by contour integrals. *SIAM Journal on Numerical Analysis*, 46(5): 2505–2523.
- Han, I.; Malioutov, D.; Avron, H.; and Shin, J. 2017. Approximating spectral sums of large-scale matrices using stochastic chebyshev approximations. *SIAM Journal on Scientific Computing*, 39(4): A1558–A1585.
- Han, I.; Malioutov, D.; and Shin, J. 2015. Large-scale log-determinant computation through stochastic Chebyshev expansions. In *International Conference on Machine Learning*, 908–917.
- Han, L.; Escolano, F.; Hancock, E. R.; and Wilson, R. C. 2012. Graph characterizations from von Neumann entropy. *Pattern Recognition Letters*, 33(15): 1958–1967.
- Hardt, M.; Ligett, K.; and McSherry, F. 2012. A simple and practical algorithm for differentially private data release. In *Advances in Neural Information Processing Systems*, 2339–2347.
- Horn, R. A.; and Johnson, C. R. 2012. *Matrix analysis*. Cambridge: Cambridge university press.
- Jaques, S.; and Rattew, A. G. 2023. Qram: A survey and critique. *arXiv preprint arXiv:2305.10310*.
- Jiang, S.; Pham, H.; Woodruff, D.; and Zhang, R. 2021. Optimal sketching for trace estimation. *Advances in Neural Information Processing Systems*, 34: 23741–23753.
- Katz, L. 1953. A new status index derived from sociometric analysis. *Psychometrika*, 18(1): 39–43.
- Kerenidis, I.; Luongo, A.; and Prakash, A. 2020. Quantum expectation-maximization for Gaussian mixture models. In *International Conference on Machine Learning*, 5187–5197. PMLR.
- Kerenidis, I.; and Prakash, A. 2020. Quantum gradient descent for linear systems and least squares. *Physical Review A*, 101(2): 022316.
- Kirby, E. C.; Mallion, R. B.; Pollak, P.; and Skrzyński, P. J. 2016. What Kirchhoff actually did concerning spanning trees in electrical networks and its relationship to modern graph-theoretical work. *Croatica Chemica Acta*, 89(4): 403–417.
- Kittaneh, F. 1985. Inequalities for the Schatten p-norm. *Glasgow Mathematical Journal*, 26(2): 141–143.
- Kontopoulou, E.-M.; Dexter, G.-P.; Szpankowski, W.; Grama, A.; and Drineas, P. 2018. Randomized Linear Algebra Approaches to Estimate the Von Neumann Entropy of Density Matrices. *arXiv preprint arXiv:1801.01072v3*.
- Li, M.; Kwok, J. T.-Y.; and Lü, B. 2010. Making large-scale Nyström approximation possible. In *Proceedings of the 27th International Conference on Machine Learning, ICML 2010*, 631.
- Li, T.; and Wu, X. 2018. Quantum query complexity of entropy estimation. *IEEE Transactions on Information Theory*, 65(5): 2899–2921.
- Low, G. H.; and Chuang, I. L. 2019. Hamiltonian simulation by qubitization. *Quantum*, 3: 163.
- Lu, Y.; and Negahban, S. N. 2015. Individualized rank aggregation using nuclear norm regularization. In *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 1473–1479. IEEE.
- Majumdar, A.; and Ward, R. K. 2011. An algorithm for sparse MRI reconstruction by Schatten p -norm minimization. *Magnetic resonance imaging*, 29(3): 408–417.

- Meila, M.; and Jordan, M. I. 2000. Learning with mixtures of trees. *Journal of Machine Learning Research*, 1(Oct): 1–48.
- Meyer, R. A.; Musco, C.; Musco, C.; and Woodruff, D. P. 2021. Hutch++: Optimal stochastic trace estimation. In *Symposium on Simplicity in Algorithms (SOSA)*, 142–155. SIAM.
- Minello, G.; Rossi, L.; and Torsello, A. 2018. On the von Neumann entropy of graphs. *Journal of Complex Networks*, 7(4): 491–514.
- Montanaro, A. 2015. Quantum speedup of Monte Carlo methods. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 471(2181): 20150301.
- Motlagh, D.; and Wiebe, N. 2023. Generalized quantum signal processing. *arXiv preprint arXiv:2308.01501*.
- Musco, C.; Netrapalli, P.; Sidford, A.; Ubaru, S.; and Woodruff, D. P. 2017. Spectrum approximation beyond fast matrix multiplication: Algorithms and hardness. *arXiv preprint arXiv:1704.04163*.
- Nakatsukasa, Y. 2020. Fast and stable randomized low-rank matrix approximation. *arXiv preprint arXiv:2009.11392*.
- Netrapalli, P.; Niranjan, U.; Sanghavi, S.; Anandkumar, A.; and Jain, P. 2014. Non-convex robust PCA. In *Advances in Neural Information Processing Systems*, 1107–1115.
- Nie, F.; Huang, H.; and Ding, C. 2012. Low-rank matrix recovery via efficient Schatten p-norm minimization. In *Twenty-sixth AAAI conference on artificial intelligence*.
- Nielsen, M. A.; and Chuang, I. 2010. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge: Cambridge University Press.
- Pace, R. K.; and LeSage, J. P. 2004. Chebyshev approximation of log-determinants of spatial weight matrices. *Computational Statistics & Data Analysis*, 45(2): 179–196.
- Passerini, F.; and Severini, S. 2008. The von Neumann Entropy of Networks. Available at SSRN 1382662.
- Persson, D.; Cortinovis, A.; and Kressner, D. 2022. Improved variants of the Hutch++ algorithm for trace estimation. *SIAM Journal on Matrix Analysis and Applications*, 43(3): 1162–1185.
- Persson, D.; and Kressner, D. 2023. Randomized low-rank approximation of monotone matrix functions. *SIAM Journal on Matrix Analysis and Applications*, 44(2): 894–918.
- Preskill, J. 1998. Lecture notes for physics 229: Quantum information and computation. *California Institute of Technology*, 16(1): 1–8.
- Quek, Y.; Kaur, E.; and Wilde, M. M. 2024. Multivariate trace estimation in constant quantum depth. *Quantum*, 8: 1220.
- Rashtchian, C.; Woodruff, D. P.; and Zhu, H. 2020. Vector-matrix-vector queries for solving linear algebra, statistics, and graph problems. *arXiv preprint arXiv:2006.14015*.
- Rasmussen, C. E.; and Nickisch, H. 2010. Gaussian processes for machine learning (GPML) toolbox. *Journal of machine learning research*, 11(Nov): 3011–3015.
- Reynolds, D. A. 2009. Gaussian mixture models. *Encyclopedia of biometrics*, 741(659-663).
- Roosta-Khorasani, F.; and Ascher, U. 2015. Improved bounds on sample size for implicit matrix trace estimators. *Foundations of Computational Mathematics*, 15(5): 1187–1212.
- Rue, H.; and Held, L. 2005. *Gaussian Markov random fields: theory and applications*. Cambridge: CRC press.
- Saibaba, A. K.; Alexanderian, A.; and Ipsen, I. C. 2017. Randomized matrix-free trace and log-determinant estimators. *Numerische Mathematik*, 137(2): 353–395.
- Shor, P. W.; and Jordan, S. P. 2008. Estimating Jones polynomials is a complete problem for one clean qubit. *Quantum Information & Computation*, 8(8): 681–714.
- Spielman, D. A.; and Srivastava, N. 2008. Graph sparsification by effective resistances. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, 563–568.
- Stathopoulos, A.; Laeuchli, J.; and Orginos, K. 2013. Hierarchical probing for estimating the trace of the matrix inverse on toroidal lattices. *SIAM Journal on Scientific Computing*, 35(5): S299–S322.
- Strang, G. 2023. *Introduction to linear algebra*. Wellesley MA: Wellesley-Cambridge Press.
- Subramanian, S.; and Hsieh, M.-H. 2021. Quantum algorithm for estimating α -Renyi entropies of quantum states. *Physical Review A*, 104(2): 022428.
- Sun, X.; Woodruff, D. P.; Yang, G.; and Zhang, J. 2021. Querying a matrix through matrix-vector products. *ACM Transactions on Algorithms (TALG)*, 17(4): 1–19.
- Suri, S.; and Vassilvitskii, S. 2011. Counting triangles and the curse of the last reducer. In *Proceedings of the 20th international conference on World wide web*, 607–614.
- Tang, E.; and Tian, K. 2024. A CS guide to the quantum singular value transformation. In *2024 Symposium on Simplicity in Algorithms (SOSA)*, 121–143. SIAM.
- Ubaru, S.; Chen, J.; and Saad, Y. 2017. Fast Estimation of $\text{tr}(f(A))$ via Stochastic Lanczos Quadrature. *SIAM Journal on Matrix Analysis and Applications*, 38(4): 1075–1099.
- Udell, M.; and Townsend, A. 2019. Why are big data matrices approximately low rank? *SIAM Journal on Mathematics of Data Science*, 1(1): 144–160.
- Van Apeldoorn, J.; Gilyén, A.; Gribling, S.; and de Wolf, R. 2020. Quantum SDP-solvers: Better upper and lower bounds. *Quantum*, 4: 230.
- Wang, G.-W.; Zhang, C.-X.; and Zhuang, J. 2014. Clustering with Prim’s sequential representation of minimum spanning tree. *Applied Mathematics and Computation*, 247: 521–534.
- Wang, Q.; Guan, J.; Liu, J.; Zhang, Z.; and Ying, M. 2022. New quantum algorithms for computing quantum entropies and distances. *arXiv preprint arXiv:2203.13522*.
- Wu, L.; Laeuchli, J.; Kalantzis, V.; Stathopoulos, A.; and Gallopoulos, E. 2016. Estimating the trace of the matrix inverse by interpolating from the diagonal of an approximate inverse. *Journal of Computational Physics*, 326: 828–844.

Xie, Y.; Gu, S.; Liu, Y.; Zuo, W.; Zhang, W.; and Zhang, L. 2016. Weighted Schatten p -norm minimization for image denoising and background subtraction. *IEEE transactions on image processing*, 25(10): 4842–4857.

Yosef, L. M.; Ubaru, S.; Horesh, L.; and Avron, H. 2024. Multivariate trace estimation using quantum state space linear algebra. *arXiv preprint arXiv:2405.01098*.

Zhang, Y.; and Leithead, W. E. 2007. Approximate implementation of the logarithm of the matrix determinant in Gaussian process regression. *Journal of Statistical Computation and Simulation*, 77(4): 329–348.

Zhao, L.; Zhao, Z.; Rebentrost, P.; and Fitzsimons, J. 2021. Compiling basic linear algebra subroutines for quantum computers. *Quantum Machine Intelligence*, 3(2): 1–10.

Zhao, Z.; Fitzsimons, J. K.; Osborne, M. A.; Roberts, S. J.; and Fitzsimons, J. F. 2019. Quantum algorithms for training Gaussian processes. *Physical Review A*, 100(1): 012304.