Towards quantum advantage for topological data analysis

Casper Gyurik [*1], Chris Cade [†2], and Vedran Dunjko [‡1]

[1]Leiden University, Leiden, The Netherlands
[2]QuSoft and CWI, Amsterdam, The Netherlands

May 7, 2020

Abstract

A particularly promising line of quantum machine leaning (QML) algorithms with the potential to exhibit exponential speedups over their classical counterparts has recently been set back by a series of "dequantization" results, that is, quantum-inspired classical algorithms which perform equally well in essence. This raises the important question whether other QML algorithms are susceptible to such dequantization, or whether it can be formally argued that they are out of reach of classical computers. In this paper, we study the quantum algorithm for topological data analysis by Lloyd, Garnerone and Zanardi (LGZ). We provide evidence that certain crucial steps in this algorithm solve problems that are classically intractable by closely relating them to the one clean qubit model, a restricted model of quantum computation whose power is strongly believed to lie beyond that of classical computation. While our results do not imply that the topological data analysis problem solved by the LGZ algorithm (i.e., Betti number estimation) is itself DQC1-hard, our work does provide the first steps towards answering the question of whether it is out of reach of classical computers. Additionally, we discuss how to extend the applicability of this algorithm beyond its original aim of estimating Betti numbers and demonstrate this by looking into quantum algorithms for spectral entropy estimation. Finally, we briefly consider the suitability of the LGZ algorithm for near-term implementations.

1 Introduction

The field of quantum machine learning (QML) is a rapidly growing and bustling field [1]. Recently it has been popular for at least two reasons. Firstly, it offers the potential to speedup machine learning tasks in big data analysis by utilizing quantum algorithms for linear algebra, such as Harrow, Hassidim and Lloyd's sparse matrix inversion algorithm [2]. Secondly, parameterized quantum circuits could function as near-term machine learning models with remarkable expressive power [3, 4, 5]. However, regarding its potential to speedup tasks in big data analysis using quantum algorithms for linear algebra, previously speculated exponential speedups in low-rank settings (a practical assumption in many machine learning tasks) were revealed to actually be at most polynomial speedups when compared with their true classical analog [6, 7].

[*]c.f.s.gyurik@liacs.leidenuniv.nl
[†]chris.cade@cwi.nl
[‡]v.dunjko@liacs.leidenuniv.nl

In this paper we consider a less studied quantum algorithm that also relies on linear-algebraic foundations, namely the quantum algorithm for topological data analysis of Lloyd, Garnerone and Zanardi (LGZ) [8]. This algorithm aims to estimate so-called Betti numbers of datasets. These are topological invariants capturing the number of holes in a topological space induced by the underlying dataset. A nice property of the LGZ algorithm is that it is clear how to efficiently load the classical data into the quantum computer and how to conclude useful information from the quantum outputs, which poses a challenge for many other QML algorithms [9]. For instance, they often require the data to be encoded into the amplitudes of a quantum state, which is far from an easy task to perform.

For the problem of Betti number estimation that arises in topological data analysis, classical hardness has only been superficially studied: exact versions are known to be NP-hard [10], but for the approximate variant solved by the LGZ algorithm hardness remains unclear. Moreover, the applicability of its methods beyond the estimation of Betti numbers remains largely unexplored. In this paper we take the first steps in resolving these two issues.

The main contribution of this paper can be summarized as follows: we prove that certain generalizations of the topological data analysis problems solved by the LGZ algorithm are DQC1-hard, and in some cases even DQC1-complete. To be precise, we show that for sparse positive semidefinite matrices the problem of estimating the fraction of its eigenvalues that lie below a threshold that is at least inverse polynomially large is DQC1-hard, and that this problem is DQC1-complete when restricted to log-local Hamiltonians. Here DQC1 denotes the class of problems that are efficiently solvable in the one clean qubit model of computation [11], a restricted model of quantum computation that is nonetheless widely believed to not be classically simulatable [12, 13]. Therefore, showing that a problem is DQC1-hard provides evidence that it is classically intractable. We note that while our result does not directly show that the problem of estimating Betti numbers that arises in topological data analysis is DQC1-hard, it does provide evidence that the problems solved by the LGZ algorithm, or, more precisely, more general applications of the same algorithm, are likely classically intractable.

In addition, we discuss how the routines employed by the LGZ algorithm can serve not just to analyze big data by estimating Betti numbers, but could also aid to speedup a more general class of algorithms. This class of algorithms consists of higher-order generalizations of algorithms that utilize the ordinary graph Laplacian, as the combinatorial Laplacian can be viewed as a higher-order generalization of the ordinary graph Laplacian. We demonstrate this by looking into quantum algorithms that estimate the spectral entropy of combinatorial Laplacians, which serves as a numerical measure to compare complex networks by [14, 15, 16]. Finally, we briefly discuss the possibilities and challenges of near-term implementations of the LGZ algorithm.

The rest of the paper is organized as follows. In Section 2 we go over the required background and we review the LGZ algorithm by highlighting its crucial steps. Our main hardness result is presented in Section 3, with the proofs deferred to Appendices A and B. Finally, in Section 4 we discuss how to extend the applicability of the methods used by the LGZ algorithm and we discuss the potential for near-term implementations in Section 5.

2 Quantum algorithm for topological data analysis

In this section we discuss the quantum algorithm for topological data analysis by Lloyd, Garnerone and Zanardi (LGZ) [8]. We begin by briefly going over the preliminaries of topological data analysis.

Afterwards, we give a short overview of the algorithm, highlighting its crucial steps along the way.

2.1 Preliminaries and definitions

Let $G = ([n], E)$ be a graph. We will encode a subset $\{i_1, \ldots, i_k\} \subseteq [n]$ as an n-bit string with Hamming weight k, where there are ones at indices $i_1, \ldots, i_k$ and zeroes elsewhere. Using this notation, let $\mathrm{Cl}_k(G) \subset \{0,1\}^n$ denote the set of bitstrings that encode a $(k+1)$-clique in G and define the clique complex of G by $\mathrm{Cl}(G) = \bigcup_{k=1}^{n} \mathrm{Cl}_k(G)$.

To encode a dataset $\{v_i\}_{i=1}^n$ into a clique complex, we first fix a grouping scale $\varepsilon > 0$ and some efficiently computable notion of distance $d(v_i, v_j)$. From this we construct the so-called *Vietoris Rips* complex, which is the clique-complex of the graph $G = ([n], E_\varepsilon)$, with

$$E_\varepsilon = \{(i,j) \mid d(v_i, v_j) \leq \varepsilon\}.$$

Let $\mathcal{H}_k$ denote the Hilbert space spanned by the computational basis state with Hamming weight $k+1$ and let $\mathcal{H}_k^G \subset \mathcal{H}_k$ denote the subspace spanned by $\mathrm{Cl}_k(G)$. On these Hilbert spaces we define the k-th boundary map $\partial_k : \mathcal{H}_k \to \mathcal{H}_{k-1}$ by linearly extending its action on the basis states given by

$$\partial_k |j\rangle = \sum_{i=0}^{k} (-1)^i \left| \widehat{j(i)} \right\rangle,$$

where $\widehat{j(i)}$ denotes the n-bit string obtained from j by setting the i-th one to zero. We denote the restriction of ∂_k to $\mathcal{H}_k^G$ with ∂_k^G. Intuitively, these boundary maps map a $(k+1)$-clique to the alternating sum of the k-cliques that it contains.

The quotient space $H_k(G) = \ker \partial_k^G / \operatorname{Im} \partial_{k+1}^G$ is called the k-th homology group of the clique complex of G. The k-th Betti number β_k^G is defined as the dimension of this k-th homology group, i.e., $\beta_k^G = \dim H_k(G)$. Intuitively, the k-th Betti number captures the amount of k-dimensional holes in your complex. They have been employed in the context of coverage verification and hole-detection in sensor networks [17, 18].

Finally, we define the k-th combinatorial Laplacian by $\Delta_k^G = (\partial_k^G)^\dagger \partial_k^G + \partial_{k+1}^G (\partial_{k+1}^G)^\dagger$. These combinatorial Laplacians can be viewed as higher-order generalizations of the ordinary graph Laplacian in that they encode interactions between cliques in your graph as opposed to encoding the interactions between single vertices. For instance, the entries of the higher-order combinatorial Laplacians encode whether two cliques share a common subclique or whether a clique is contained in another larger clique.

These combinatorial Laplacians were first introduced by Eckmann [19], who formulated and proved the discrete version of the Hodge theorem, which states that the kernel of Δ_k^G is isomorphic to the k-th homology group. This implies that the nullity (i.e., dimension of the kernel) of the k-th combinatorial Laplacian satisfies

$$\mathrm{null}(\Delta_k^G) = \beta_k^G, \tag{1}$$

which is often used by algorithms as a way to compute Betti numbers [20].

2.2 The LGZ algorithm

Lloyd, Garnerone and Zanardi (LGZ) devised a quantum algorithm that estimates Betti numbers of clique complexes [8]. Their quantum algorithm does so by estimating the nullity of the k-th combinatorial Laplacian, which as discussed in the previous section equals the k-th Betti number.

Given a sparse Hermitian matrix one can use Hamiltonian simulation and quantum phase estimation to sample from its eigenvalues. If the eigenvector register of the quantum phase estimation algorithm is in the maximally mixed state, then measurements of the eigenvalue register will produce approximations of eigenvalues, sampled uniformly at random.

Lloyd et al. employ this routine to estimate the nullity of the combinatorial Laplacian by estimating the number of eigenvalues that lie below a certain threshold. Namely, if this threshold is smaller than the spectral gap (i.e., the smallest non-zero eigenvalue) then this number will be equal to the nullity. Counting the number of such small eigenvalues can be done either via quantum counting, or by repeatedly sampling and computing the relative frequency of sampling such small eigenvalues.

The combinatorial Laplacians Δ_k^G are in general not sparse. To get around this, the LGZ algorithm applies the above routine to the so-called Dirac operator B which collects the boundary maps in the following way:

$$B = \begin{bmatrix} 0 & \partial_1 & 0 & \dots & \dots & 0 \\ \partial_1^\dagger & 0 & \partial_2 & \dots & \dots & 0 \\ 0 & \partial_2^\dagger & 0 & \ddots & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 0 & \partial_n \\ 0 & 0 & 0 & \dots & \partial_n^\dagger & 0 \end{bmatrix} \quad \text{and satisfies} \quad B^2 = \begin{bmatrix} \Delta_1 & 0 & \dots & 0 \\ 0 & \Delta_2 & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \dots & \Delta_n \end{bmatrix}, \tag{2}$$

where $\Delta_k = (\partial_k)^\dagger \partial_k + \partial_{k+1}(\partial_{k+1})^\dagger$. Together with a method to restrict the domain of the operator, their algorithm produces approximations of eigenvalues of the combinatorial Laplacian, sampled uniformly at random. To make our presentation self-contained, we will briefly review this method. For a more complete review of the LGZ algorithm see [21].

In the first step of the LGZ algorithm, Grover's algorithm is used to prepare the so-called *simplex state* given by

$$|\psi_k\rangle = \frac{1}{\sqrt{\mathrm{Cl}_k(G)}} \sum_{j \in \mathrm{Cl}_k(G)} |j\rangle.$$

When given access to the adjacency matrix of G, one can check in $O(k^2)$ operations whether a bitstring $j \in \{0,1\}^n$ encodes a valid k-clique and mark them in the application of Grover's algorithm. Once we have prepared the state $|\psi_k\rangle$, we prepare the maximally-mixed state over $\mathcal{H}_k^G$ given by

$$\rho_k = \frac{1}{\mathrm{Cl}_k(G)} \sum_{j \in \mathrm{Cl}_k(G)} |j\rangle\langle j|,$$

by applying a CNOT gate to each qubit of $|\psi_k\rangle$ into some ancilla qubits and tracing those out. We note that ρ_k can of course also be directly prepared without the use of Grover's algorithm by using rejection sampling: we choose $j \in \{0,1\}^n$ uniformly at random and accept it if it encodes a

valid k-clique. This is quadratically less efficient, however it has advantages if one has near-term implementations in mind, as it is a completely classical subroutine.

After preparing the state ρ_k the LGZ algorithm uses Hamiltonian simulation to implement the unitary e^{iB} to which it applies quantum phase estimation with the state $\rho_k \otimes |k\rangle\langle k|$ in the eigenvector register. After the quantum phase estimation, the LGZ algorithm measures the eigenvalue register and squares the measurement results. This results in approximations of the eigenvalues of Δ_k^G, where each eigenvalue appears with uniform probability.

Recall that the goal was to estimate the k-th Betti number, which by Equation 1 is equal to the number of eigenvalues of Δ_k^G that are equal to zero. Unfortunately, we cannot do so directly using the above routine, as we are only able to sample approximations of eigenvalues and hence cannot tell if the actual eigenvalue is exactly equal to zero or not. However, we can use this routine of generating approximations of uniformly random sampled eigenvalues of Δ_k^G to estimate the number of small eigenvalues. In fact, what we can estimate is the *fraction* of eigenvalues that lie below a given threshold (allowing some mistakes if the eigenvalue is very close to the threshold). Moreover, if this threshold is less than the spectral gap (i.e., the smallest non-zero eigenvalue) then this fraction will be equal to the *normalized* Betti number (i.e., $\beta_k^G / \dim \mathcal{H}_k^G$). Unfortunately, not much is known in terms of lower bounds for the spectral gap of general combinatorial Laplacians [20].

In summary, the LGZ algorithm samples from approximations of eigenvalues of the combinatorial Laplacian, where each eigenvalue appears with uniform probability, to estimate not the actual dimension of the kernel, but rather the fraction of small eigenvalues (i.e., eigenvalues that lie below a given threshold). In the next section we study the complexity of the problems of estimating the fraction of eigenvalues that lie below a given threshold and that of generating approximations of eigenvalues that are sampled uniformly at random.

3 Problem definitions and our results

To show that the LGZ algorithm is out of reach of classical computers, one would like to show that the problem of estimating Betti numbers is classically intractable. As discussed in Section 2, this problem is equivalent to counting the number of eigenvalues of the combinatorial Laplacian that are equal to zero. In this section, we take a first step in this direction by addressing the hardness of the more general problem of estimating the fraction of eigenvalues that lie below a given threshold for arbitrary sparse positive semidefinite matrices. We do so by building on results by Brandão on hard problems for the complexity class DQC1 that are related to generating approximations of eigenvalues that are sampled uniformly at random [22].

3.1 Sparse Hermitian uniform eigenvalue sampling and spectral density

We begin by formally defining the problems whose complexity we will study. First, we define the problem of generating approximations of eigenvalues that are sampled uniformly at random. Afterwards, we define the problem of estimating the fraction of eigenvalues that lie below a given threshold, which we will call the low-lying eigenvalue density.

We shall define these problems for general sparse positive semidefinite matrices. We call a $2^n \times 2^n$ matrix sparse if at most $\mathcal{O}(\text{poly}(n))$ entries in each row are non-zero. A special class of sparse matrices that we will consider is the class of *log-local* Hamiltonians. That is, n-qubit

Hamiltonians that can be written as a sum

$$H = \sum_{j=1}^{m} H_j,$$

where each H_j acts on at most $\mathcal{O}(\log n)$ qubits and $m = \mathcal{O}(\text{poly}(n))$.

As already mentioned, we begin by defining the problem of generating approximations of eigenvalues that are sampled uniformly at random for sparse positive semidefinite matrices.

Sparse uniform eigenvalue sampling (SUES)

Input: 1) A sparse positive semidefinite matrix $H \in \mathbb{C}^{2^n \times 2^n}$ given either by sparse access or as local terms $\{H_i\}_{i=1}^m$ if it is a log-local Hamiltonian. Also, suppose the eigenvalues of H are $\{\lambda_k\}_{k=0}^{2^n-1}$ satisfying $\lambda_k < \text{poly}(n)$ for each k.
2) An estimation precision $\delta > 0$.
3) A sampling error probability $\mu > 0$.

Output: An estimate of λ_k up to additive error δ with probability at least $1 - \mu$, where k is drawn uniformly at random from the set $\{0, \ldots, 2^n - 1\}$.

Note that the above is a sampling problem rather than a standard decision or searching problem. Here we mean that an algorithm for the above problem should output an estimate up to additive precision δ of some eigenvalue $\lambda \in \{\lambda_k\}_{k=0}^{2^n-1}$, where for $0 \leq k \leq 2^n - 1$ we have that

$$\Pr(\lambda = \lambda_k) = \frac{1}{2^n},$$

and that it fails to do so (i.e., outputs a sample that does not satisfy this distribution) with probability less than μ[1].

Define the *eigenvalue density between* $a, b \in \mathbb{R}$ of an Hermitian matrix $H \in \mathbb{C}^{2^n \times 2^n}$ by

$$D_H(a, b) = \frac{1}{2^n} \sum_{k \,:\, a \leq \lambda_k \leq b} 1,$$

where $\lambda_0 \leq \cdots \leq \lambda_{2^n-1}$ denote the eigenvalues of H. We define the problem of estimating the low-lying eigenvalue density (i.e., the fraction of small eigenvalues) as follows.

Low-lying eigenvalue density estimation (LLED)

Input: 1) A sparse positive semidefinite matrix $H \in \mathbb{C}^{2^n \times 2^n}$ given either by sparse access or as local terms $\{H_i\}_{i=1}^m$ if it is a log-local Hamiltonian. Also, suppose the eigenvalues of H are $\{\lambda_k\}_{k=0}^{2^n-1}$ satisfying $\lambda_k < \text{poly}(n)$ for each k.
2) A threshold $b = \Omega(1/\text{poly}(n))$.
3) Precision parameters $\delta, \varepsilon > 0$.

Output: An estimate $\chi \in [0, 1]$ satisfying $|\chi - y| < \varepsilon$, with

$$D_H(0, b) \leq y \leq D_H(0, b + \delta).$$

To provide some intuition for this definition note that we require the non-zero precision parameter δ as no finite-time algorithm can estimate the eigenvalues to infinite precision, which would

[1] For a more formal definition and discussion of this notion of approximate sampling see [23].

be required to make sure a given eigenvalue is (or is not) exactly less than or equal to b. Also, the non-zero precision parameter ε is necessary due to the sampling error we incur by sampling from the output of SUES and counting the number of samples that lie below the given threshold.

3.2 The one clean qubit model of computation

In the following section we will relate the above problems to the so-called one clean qubit model of quantum computation [11]. In this model we are given a quantum state consisting of a single qubit in the pure state $|0\rangle$, and $n-1$ qubits in the maximally mixed state. One can then apply any polynomially-sized quantum circuit to this state, and measure the first qubit in the computational basis. Following [11], we will refer to the complexity class of problems that can be solved in polynomial time using this model of computation as DQC1 – "deterministic quantum computation with a single clean qubit".

We will refer to a problem as DQC1-hard if any problem in DQC1 can be reduced to it under polynomial time truth-table reductions. That is, a problem L is DQC1-hard if we can decide any problem in DQC1 using polynomially many non-adaptive queries to L, together with polynomial time pre-processing of the inputs and post-processing of the outcomes. Technically, instead of considering the problem of estimating a given quantity, we should consider the decision problem of deciding whether this quantity is greater than a given threshold. However, as the estimation version of these problems are straightforwardly reduced to their decision version using binary search, we will bypass this point.

The problems of estimating the normalized trace of a unitary matrix corresponding to a polynomial-depth quantum circuit and that of estimating Jones polynomials are well-known examples of complete problems for DQC1 [12]. As these quantities appear to be hard to estimate classically, it seems that the one clean qubit model of computation is more powerful than classical computation. Moreover, there is also complexity-theoretical evidence that demonstrates this [13]. Hence, showing that a problem is DQC1-hard is good evidence that it is likely classically intractable.

3.3 Our results

Our main technical results, the proofs of which can be found in the appendix, are given as follows.

Theorem 1. *LLED with $1/\delta, 1/\varepsilon = \mathcal{O}(poly(n))$ is DQC1-hard. Moreover, LLED with these precision parameters remains DQC1-hard when restricted to log-local Hamiltonians.*

We remind the reader that we are interested in the hardness of the LLED problem because it is the closest generalization of the problem of Betti number estimation, which is at the core of the LGZ algorithm for topological data analysis. We discuss this in more detail in the next section.

We prove Theorem 1 by reducing LLED to the problem of estimating a quantity similar to a normalized subtrace (i.e., the normalized sum of eigenvalues that lie below a given threshold), which is DQC1-hard as shown by Brandão [22]. We approximate this normalized subtrace by constructing a histogram approximation of the low-lying spectrum and computing the area under this histogram. To do so we use evaluations of instances of LLED to estimate the number of eigenvalues that lie in each bin of the histogram. To avoid double counting of eigenvalues due to imprecisions around the thresholds, we subtract the outcome of LLED of the lower-threshold of the bin from the outcome

of LLED of the upper-threshold of the bin, obtaining an estimate of the eigenvalues within the bin and misplacing eigenvalues by at most one bin in the process[1].

Besides being hard for the class DQC1, both LLED and SUES can be solved efficiently on a quantum computer. In fact, as the computational capacity of the DQC1 model does not change by allowing up to $\mathcal{O}(\log(n))$ clean (i.e., pure) qubits [12], it turns out that in some instances we can perform the algorithms that solve LLED and SUES using the computational resources of DQC1. These extra clean qubits are important since the quantum algorithms we need (i.e., Hamiltonian simulation and quantum phase estimation) require additional ancilla qubits, which need to be initialized in the all-zeroes state and hence be clean. This results in the following proposition, which together with Theorem 1 also implies the subsequent corollary.

Proposition 2. *LLED and SUES with* $1/\delta, 1/\varepsilon, 1/\mu = \mathcal{O}(poly(n))$ *are contained in* BQP*. Moreover, if* H *is log-local then these problems are also contained in* DQC1*.*

Corollary 3. *LLED and SUES with* $1/\delta, 1/\varepsilon, 1/\mu = \mathcal{O}(poly(n))$ *are* DQC1*-complete when restricted to log-local Hamiltonians.*

Remark. *The complexity of SUES was stated as an open problem in [23]. The fact that SUES is* DQC1*-hard when restricted to log-local Hamiltonians was already shown by Brandão [22].*

3.3.1 Quantum advantage for topological data analysis

As discussed in Section 2, the problem of Betti number estimation that arises in topological data analysis is equivalent to estimating the number of eigenvalues that are equal to zero for a special class of positive semidefinite matrices (i.e., the combinatorial Laplacians). Our result is not the hardness of exactly this problem, but we show that estimating the number of inverse polynomially small eigenvalues (i.e., not necessarily *exactly* equal to zero) of a more general class of positive semidefinite matrices is hard for the class DQC1 (and thus presumably classically intractable). Thus, the main open questions regarding the classical hardness of Betti number estimation are twofold.

Firstly, it is unclear whether LLED remains classically hard when we restrict the family of positive semidefinite matrices to the combinatorial Laplacians. Note that there already exist restricted families of sparse positive semidefinite matrices that are hard for LLED, such as the log-local Hamiltonians obtained by applying Kitaev's circuit-to-Hamiltonian construction to polynomial depth quantum circuits consisting of a small set of approximately universal gates. As a part of this work, we attempted to close this gap by investigating whether the combinatorial Laplacians already contain chosen Kitaev Hamiltonians as sub-matrices of sufficiently larger instances. While indeed sub-matrices related to various quantum gates can be found in the combinatorial Laplacians, we did not succeed in finding an explicit embedding. In our view, this remains a promising approach in showing that LLED remains DQC1-hard when restricted to the combinatorial Laplacians, although other routes (e.g., going through the discrete structures related to Tutte and Jones polynomials [24, 12]) are possible as well. Regardless, even if LLED does not remain classically hard

[1] For completeness, we believe that our approach could be modified to a Karp reduction by encoding an instance of the normalized subtrace estimation problem into a single instance of LLED. This construction entails manipulation of the Kitaev circuit-to-Hamiltonian construction (i.e., putting energy penalties on certain eigenvectors) to subtract eigenvalue densities and taking direct sums of matrices to take the convex combination of eigenvalue densities. However, as this reduction is not vital for our main claim, nor trivial, we leave this question for future work.

when restricted to combinatorial Laplacians, we can envision generalizations of spectral methods going beyond the combinatorial Laplacians that could end up being DQC1-hard using our result, yielding novel quantum algorithms for classically intractable problems related to data analysis.

Secondly, it is unclear whether the problem of estimating the nullity (i.e., the number of eigenvalues *exactly* equal to zero) is at least as hard as estimating the number of inverse polynomially small eigenvalues. Nonetheless, if we know an inverse polynomially large lower-bound on the smallest non-zero eigenvalue of the combinatorial Laplacian, then we can estimate the normalized Betti number (i.e., $\beta_k^G / \dim \mathcal{H}_k^G$) by solving LLED using this lower-bound as the threshold. However, in general not much is known regarding lower bounds on the smallest non-zero eigenvalue of the combinatorial Laplacian [20].

On the other hand, if we do not know such a lower bound on the smallest non-zero eigenvalue, then the fraction of small eigenvalues of the combinatorial Laplacian still conveys information about the underlying graph. For instance, Cheeger's inequality tells us that the size of the smallest non-zero eigenvalue of the zeroth combinatorial Laplacian is closely related to a numerical measure that captures whether or not a graph contains a "bottleneck". Hence, barring the matter regarding the restriction to the combinatorial Laplacians, our result shows that the LGZ algorithm solves a presumably classically intractable problem whose solution encodes interesting information regarding the underlying graph.

4 Beyond Betti numbers

Given that the LGZ algorithm for Betti number estimation is one of the remaining candidates for exponential speedups over classical methods, it is interesting to consider generalizations of the routines employed by this algorithm that may share the same hardness as discussed in Section 3. The combinatorial Laplacian has many interesting graph-oriented applications beyond the applications to homology discussed in Section 2. The main reason for this is that one can view the combinatorial Laplacian as a higher-order generalization of the ordinary graph Laplacian $L_G = D - A$, where D is the degree matrix and A is the adjacency matrix of your graph. Instead of encoding the connectivity of vertices, the k-th combinatorial Laplacian encodes the connectivity of k-cliques.

Because of this, the many algorithms that utilize the ordinary graph Laplacian can be generalized to employ higher-order combinatorial Laplacians. For example, in [25] the authors introduce higher-order generalizations of spectral clustering and label propagation by utilizing the spectral structure of the higher-order combinatorial Laplacians, as opposed to the ordinary graph Laplacian. Moreover, in [26] the authors describe how random walks on the vertices of a graph, which are usually related to a normalized version of the ordinary graph Laplacian, can be generalized to random walks on the edges and they discuss how this is related to a normalized version of the 1st combinatorial Laplacian.

The well-known fact that the eigenvalues of the ordinary graph Laplacian encode properties of the underlying graph also carries over to the higher-order combinatorial Laplacians. For example, in [27] the authors study the spectrum of a normalized version of the higher-order combinatorial Laplacian and they show that it conveys information about the existence of circuits of cliques (i.e., ordered lists of adjacent cliques that cover the whole graph) and about the chromatic number. Additionally, Cheeger's inequality (which relates the sparsest cut of a graph to the smallest nonzero eigenvalues of its ordinary graph Laplacian) turns out to have a higher-order generalization that utilizes the combinatorial Laplacian [28]. Lastly, Kirchhoff's matrix tree theorem (which relates

the eigenvalues of the ordinary graph Laplacian to the number of spanning trees) also turns out to have a similar higher-order generalization [29].

Motivated by all of the above, we conclude that there is the potential for quantum algorithms to compute features of graphs using the combinatorial Laplacian, while possibly maintaining some of the underlying classical hardness as discussed in Section 3. Moreover, the size of the k-th combinatorial Laplacian can be exponentially larger than that of the ordinary graph Laplacian (it scales as $\mathcal{O}\left(\binom{n}{k+1}\right)$ as opposed to $\mathcal{O}(n)$, where n is the number of vertices in your graph) making classical algorithms potentially infeasible for larger graphs. In the next section we take a first step in this direction by studying quantum algorithms for spectral entropy estimation, which when applied to combinatorial Laplacians can serve as a numerical measure to compare complex networks by.

4.1 Estimating the spectral entropy of the combinatorial Laplacian and complex network analysis

Recently, several quantum information-inspired entropic measures for complex network analysis have been proposed [30, 31]. One example of these are the spectral entropies of the combinatorial Laplacian, which measure the degree of overlapping of cliques within your complex network [14, 15, 16].

If $\lambda_0, \ldots, \lambda_{N-1}$ denote the eigenvalues of a positive semidefinite matrix H, then its spectral entropy is usually defined by

$$S(H) = -\sum_{j=0}^{N-1} p(\lambda_j) \log(p(\lambda_j)), \tag{3}$$

where we define $p(\lambda_j) = \lambda_j / (\sum_k \lambda_k)$. This spectral entropy coincides with the von Neumann entropy of H when interpreting it as a density matrix (i.e., after normalizing it with respect to its trace). Equivalently, it coincides with the Shannon entropy of the distribution over the eigenvalues of H whose probabilities are given by $p(\lambda_j)$. This spectral entropy is a special case of the more general α-Renyi spectral entropy, which is given by

$$S_\alpha(H) = \frac{1}{1-\alpha} \log \left(\sum_{j=0}^{N-1} p(\lambda_j)^\alpha \right), \tag{4}$$

where $\alpha \geq 0$ and $\alpha \neq 1$. The limit for $\alpha \to 1$ is the spectral entropy as defined in Equation 3.

In the rest of this section we shall discuss how quantum algorithms can be used to estimate spectral entropies of positive semidefinite matrices, which are given either via sparse access or as a log-local description. This then allows us to estimate spectral entropies of the combinatorial Laplacians by combining it with the methods discussed in Section 2.2.

One approach to estimating the spectral entropy of a positive semidefinite matrix H with eigenvalues $\lambda_0, \ldots, \lambda_{N-1}$ is to first prepare a quantum state that when measured outputs (an approximation of) some eigenvalue λ_j with probability $p(\lambda_j)$ and subsequently apply the methods discussed in [32]. To prepare such a quantum state consider the following sketch of an algorithm:

1. Prepare the following input state by taking a maximally entangled state and adding two ancillary registers

$$|\psi\rangle_{in} = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} |\psi_k\rangle |\phi_k\rangle \otimes \left|0^t\right\rangle \otimes |0\rangle,$$

where $\{|\psi_k\rangle\}_{k=0}^{N-1}$ are orthonormal eigenvectors of H and $\{|\phi_k\rangle\}_{k=0}^{N-1}$ is an orthonormal basis of $\mathbb{C}^N$.

2. Apply quantum phase estimation of $e^{2\pi i H}$ to the first register to prepare an approximation of the state

$$\frac{1}{\sqrt{N}}\sum_{k=0}^{N-1}|\psi_k\rangle|\phi_k\rangle\otimes\left|\widetilde{\lambda_k}\right\rangle\otimes|0\rangle,$$

where $\left|\widetilde{\lambda_k}\right\rangle$ contains a superposition over t-bit approximation of the eigenvalue λ_k that corresponds to the eigenvector $|\psi_k\rangle$.

3. Use controlled rotations to imprint the t-bit approximations of the eigenvalues into the amplitudes of the last register to prepare an approximation the state

$$\frac{1}{\sqrt{N}}\sum_{k=0}^{N-1}|\psi_k\rangle|\phi_k\rangle\otimes\left|\widetilde{\lambda}_k\right\rangle\otimes\left(\sqrt{\widetilde{\lambda_k}}|0\rangle+\sqrt{1-\widetilde{\lambda_k}}|1\rangle\right).$$

4. Use amplitude amplification to amplify states whose last qubit is in the state $|0\rangle$ to prepare an approximation of the state

$$|\psi\rangle_{out}=\frac{1}{\sqrt{\mathrm{Tr}(H)}}\sum_{k=0}^{N-1}\sqrt{\lambda_k}|\psi_k\rangle|\phi_k\rangle\otimes\left|\widetilde{\lambda_k}\right\rangle\otimes|0\rangle.$$

Clearly, when you measure the second register of $|\psi\rangle_{out}$ you obtain (an approximation) of some eigenvalue λ_j with probability $p(\lambda_j)$. Note that all approximations of the states in the above algorithm can be made exponentially precise (except for the precision of the eigenvalues, which can be made polynomially precise). Looking at the cost of the above algorithm, we note that Steps 2 and 3 can be implemented in time $\mathcal{O}(\mathrm{polylog}(n))$. Moreover, as the amplitude of the states ending with a 0 is given by $\mathrm{Tr}(H)/2^n$, the cost of step 4 is given by $\mathcal{O}\left(2^{n/2}/\sqrt{\mathrm{Tr}(H)}\right)$.

Another technique that allows one to estimate the α-Renyi spectral entropy of a positive semidefinite matrix H with eigenvalues $\lambda_0,\ldots,\lambda_{N-1}$ is to first normalize H with respect to its trace and subsequently apply Lemma 1 of [33] to estimate $\frac{1}{1-\alpha}\left[\log\left(\sum_j f(p(\lambda_j))/N\right)+\log(N)\right]=S_\alpha(H)$, where $f(x)=x^\alpha$. In order to normalize H with respect to its trace we estimate $\mathrm{Tr}(H)$ and subsequently divide H by this estimate. Using the methods discussed in [33] we can estimate $\mathrm{Tr}(H)$ to within polynomial precision, albeit at exponential cost.

The above methods can potentially outperform classical methods based on the following two arguments. First, in [32] the authors show that one can estimate the spectral entropy of a classical distribution quadratically faster in terms of the matrix size when given quantum sampling access (i.e., a quantum state whose amplitudes are the square-roots of the probabilities) as opposed to classical sampling access (i.e., a black-box that outputs samples according to the distribution). Secondly, the α-Renyi entropies are closely related to the Schatten p-norms $||H||_p=\mathrm{Tr}(|H|^p)^{1/p}$, as $S_\alpha(H)=\frac{\alpha}{1-\alpha}\log(||H||_\alpha)$ if H is positive semidefinite and $\mathrm{Tr}(H)=1$, and it turns out that the problem of estimating Schatten p-norms is DQC1-hard [33].

5 Possibilities for near-term implementations

As current quantum hardware is still very limited, it is crucial to make sure to use it to its fullest potential when trying to run the LGZ algorithm. Luckily, the steps in the LGZ algorithm are closely related to quantum simulation (i.e., quantum chemistry and many-body physics) which opens up the potential to use the many results in this field that are aimed at optimizing the use of the current resource-constrained hardware [34]. Moreover, as the LGZ algorithm has a large part of its input qubits in the maximally mixed state, initialization of these qubits is less of a challenge.

When it comes to near-term implementations of quantum algorithms the main concerns are the depth of the circuit[1], the total number of required qubits to achieve a real-world quantum advantage, and the robustness to noise in the hardware[2]. In this section we will discuss these challenges in the context of implementations of the LGZ algorithm.

Regarding the depth of the circuit, the LGZ algorithm requires Hamiltonian simulation, and phase estimation of the realized unitaries. This may yield substantial depth in "vanilla" implementations (i.e., beyond depths attainable in current devices). However, here it is likely much progress can be made as resource optimization of phase estimation and Hamiltonian simulation are both of central interest in quantum chemistry, and other algorithms for strongly correlated systems [35, 36, 37].

Regarding the total number of required qubits for a real-world quantum advantage, we can differentiate two separate costs. The first involves the actual graph sizes, i.e., the dimension of the combinatorial Laplacians whose kernel dimension we wish to estimate. Here, employing direct exact diagonalization methods hits a practical wall around 40 spins (leading to an approximately 2^{40}-dimensional matrix). We have not been able to find substantially better classical methods for kernel estimation.

The second cost involves all the ancillary qubits needed to run the algorithm. Here the main bottleneck comes from the fact that the Dirac operators (i.e., the square root of the combinatorial Laplacians) to which we apply Hamiltonian simulation are likely not log-local Hamiltonians, but just n-sparse matrices. Standard methods for Hamiltonian simulation of such objects requires the realization of sparse oracles [38]; realizing these directly will involve twice as many qubits, as the work space of the target Hamiltonian.

Taking all of this into account, a back-of-the-envelope calculation suggests that ≈ 100 qubits could be sufficient to challenge the best-known classical methods. However, much improvement here is possible as well. Given that the target Hamiltonian is the Dirac operator (which is fixed for a given graph size, leading to exponentially many graphs whose Betti numbers we are able to approximate using this Dirac operator), specialized optimization, and indeed, precompilation is possible. Note that we only need an implementation of the unitary e^{iBt}, and in principle this can be done without any ancillary systems, with a likely blow up in depth (and, pre-compilation overheads). This process may yield a trade-off between the required number of ancilla qubits and the amount pre-compilation together with depth of the precompiled circuit. A particular possibility for pre-compilation is finding decompositions in terms of linear combinations of unitaries [38] for realizing the unitary e^{iBt}, to achieve more-space efficient realizations of the simulation step. This would also be interesting from a different perspective, as it may shed light of what unitaries natively

[1] Of course, total gate numbers are vital as well, but depth correlates directly to required coherence times.

[2] Indeed, the final circuit shape, and its match to easy-to-implement operations on a given device will be very relevant in practice, but here we focus on aspects which are less specific to particular architectures.

occur as sub-matrices of the combinatorial Laplacian. This may also lead to new avenues to explore regarding the hardness of the family of combinatorial Laplacians for LLED, as it may offer a chance of identifying how to encode sufficiently rich families of matrices related to Kitaev Hamiltonians into the structure of the combinatorial Laplacians.

Regarding the resilience to noise, the LGZ algorithm will suffer all the issues quantum phase estimation methods for quantum chemistry and many-body physics suffer, and consequently, all mitigation methods can likely be applied as well. However, as we are interested in data-analysis applications, the fact that the algorithms serve the purpose of actually dealing with noise (in the data), one may hope this makes the use of noisy algorithms less detrimental compared to when solving more exact problems, but we cannot make such claims at this point[3]. At the moment, there is little to no work done analyzing the robustness of these types of algorithms experimentally or theoretically, and we plan on pursuing this in the future.

6 Summary

In this work, we addressed the classical hardness of the problems in topological data analysis, and in related varieties of graph analysis, that are solved by the quantum LGZ algorithm. Further, we examined generalizations of the methods employed by the LGZ algorithm and briefly discussed near-term implementations.

Regarding the classical hardness, while we come up short of a full resolution of the question, we take the first steps in this direction by showing that certain generalizations of the topological data analysis problem – which could still have applications in topological data analysis – are DQC1-hard, and hence presumably classically intractable.

As for generalizations of the LGZ algorithm, we discussed how algorithms that employ the spectral structure of the ordinary graph Laplacian can be generalized to higher-order combinatorial Laplacians, opening up a potential area for speedups by quantum algorithms. Moreover, we provide a concrete example by examining quantum algorithms for spectral entropy estimation of combinatorial Laplacians. With respect to near-term implementations, we highlighted the challenges and opportunities of near-term implementation of the LGZ algorithm by investigating the required resources for basic implementations, and by identifying possible directions for how to minimize these required resources.

The presented work opens a number of options for extensions and follow up, leading to full proofs of the hardness of approximate Betti number estimation, to different LGZ-related algorithms for which hardness may be easier to establish, and to more efficient implementation options better suited for near-term devices.

Acknowledgements

VD and CG acknowledge the support of the Quantum Software Consortium. CC acknowledges the support of the QuantERA project QuantAlgo 680-91-034. VD and CG thank Dorit Aharonov, Tomoyuki Morimae and Ronald de Wolf for early discussions on the topic. The authors are grateful to Ronald de Wolf for useful comments on the manuscript.

[3]More precisely, if one can formally "commute" the noise from the algorithm to the data, such statements can be made fully formal. While this is rarely formally possible, the intuition stands. In the case of topological data analysis, this is more unclear as we are dealing with discrete structures – graphs, and corresponding Laplacians – hence we have few smoothness assumptions to rely on.

References

[1] V. Dunjko and P. Wittek, "A non-review of quantum machine learning: trends and explorations," *Quantum Views*, vol. 4, p. 32, 2020.

[2] A. W. Harrow, A. Hassidim, and S. Lloyd, "Quantum algorithm for linear systems of equations," *Physical review letters*, vol. 103, no. 15, p. 150502, 2009. arXiv:0811.3171.

[3] V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, "Supervised learning with quantum-enhanced feature spaces," *Nature*, vol. 567, no. 7747, pp. 209–212, 2019. arXiv:1804.11326.

[4] M. Schuld, A. Bocharov, K. M. Svore, and N. Wiebe, "Circuit-centric quantum classifiers," *Physical Review A*, vol. 101, no. 3, p. 032308, 2020. arXiv:1804.00633.

[5] M. Benedetti, E. Lloyd, S. Sack, and M. Fiorentini, "Parameterized quantum circuits as machine learning models," *Quantum Science and Technology*, vol. 4, no. 4, p. 043001, 2019. arXiv:1906.07682.

[6] E. Tang, "A quantum-inspired classical algorithm for recommendation systems," in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pp. 217–228, 2019. arXiv:1807.04271.

[7] N.-H. Chia, A. Gilyén, T. Li, H.-H. Lin, E. Tang, and C. Wang, "Sampling-based sublinear low-rank matrix arithmetic framework for dequantizing quantum machine learning," in *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, to appear 2020. arXiv:1910.06151.

[8] S. Lloyd, S. Garnerone, and P. Zanardi, "Quantum algorithms for topological and geometric analysis of data," *Nature communications*, vol. 7, no. 1, pp. 1–7, 2016. arXiv:1408.3106.

[9] S. Aaronson, "Read the fine print," *Nature Physics*, vol. 11, no. 4, pp. 291–293, 2015.

[10] M. Adamaszek and J. Stacho, "Complexity of simplicial homology and independence complexes of chordal graphs," *Computational Geometry*, vol. 57, pp. 8–18, 2016.

[11] E. Knill and R. Laflamme, "Power of one bit of quantum information," *Physical Review Letters*, vol. 81, no. 25, p. 5672, 1998. arXiv:quant-ph/9802037.

[12] P. W. Shor and S. P. Jordan, "Estimating Jones polynomials is a complete problem for one clean qubit," *Quantum Information & Computation*, vol. 8, no. 8, pp. 681–714, 2008. arXiv:0707.2831.

[13] T. Morimae, "Hardness of classically sampling the one-clean-qubit model with constant total variation distance error," *Physical Review A*, vol. 96, no. 4, p. 040302, 2017. arXiv:1704.03640.

[14] F. Passerini and S. Severini, "Quantifying complexity in networks: the von Neumann entropy," *International Journal of Agent Technologies and Systems (IJATS)*, vol. 1, no. 4, pp. 58–67, 2009. arXiv:0812.2597.

[15] D. Simmons, J. Coon, and A. Datta, "The quantum Theil index: characterizing graph centralization using von Neumann entropy," *Journal of Complex Networks*, vol. 6, no. 6, pp. 859–876, 2018. arXiv:1707.07906.

[16] S. Maletić and M. Rajković, "Combinatorial Laplacian and entropy of simplicial complexes associated with complex networks," *The European Physical Journal Special Topics*, vol. 212, no. 1, pp. 77–97, 2012.

[17] A. Tahbaz-Salehi and A. Jadbabaie, "Distributed coverage verification in sensor networks without location information," *IEEE Transactions on Automatic Control*, vol. 55, no. 8, pp. 1837–1849, 2010.

[18] R. Ghrist and A. Muhammad, "Coverage and hole-detection in sensor networks via homology," in *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005*, pp. 254–260, IEEE, 2005.

[19] B. Eckmann, "Harmonische Funktionen und Randwertaufgaben in einem Komplex," *Commentarii Mathematici Helvetici*, vol. 17, no. 1, pp. 240–255, 1944.

[20] J. Friedman, "Computing Betti numbers via combinatorial Laplacians," *Algorithmica*, vol. 21, no. 4, pp. 331–346, 1998.

[21] S. Gunn and N. Kornerup, "Review of a quantum algorithm for Betti numbers," *arXiv preprint arXiv:1906.07673*, 2019. arXiv:1906.07673.

[22] F. G. Brandão, *Entanglement theory and the quantum simulation of many-body physics*. PhD thesis, University of London, 2008. arXiv:0810.0026.

[23] P. Wocjan and S. Zhang, "Several natural BQP-complete problems," *arXiv preprint quant-ph/0606179*, 2006.

[24] H. Ahmadi and P. Wocjan, "On the quantum complexity of evaluating the Tutte polynomial," *Journal of Knot Theory and its Ramifications*, vol. 19, no. 06, pp. 727–737, 2010.

[25] B. Osting, S. Palande, and B. Wang, "Spectral sparsification of simplicial complexes for clustering and label propagation," *arXiv preprint arXiv:1708.08436*, 2017.

[26] M. T. Schaub, A. R. Benson, P. Horn, G. Lippner, and A. Jadbabaie, "Random walks on simplicial complexes and the normalized Hodge 1-laplacian," *arXiv preprint arXiv:1807.05044*, 2018.

[27] D. Horak and J. Jost, "Spectra of combinatorial Laplace operators on simplicial complexes," *Advances in Mathematics*, vol. 244, pp. 303–336, 2013. arXiv:1105.2712.

[28] A. Gundert and M. Szedláky, "Higher dimensional discrete Cheeger inequalities," *Journal of Computational Geometry*, vol. 6, no. 2, 2015. arXiv:1401.2290.

[29] A. Duval, C. Klivans, and J. Martin, "Simplicial matrix-tree theorems," *Transactions of the American Mathematical Society*, vol. 361, no. 11, pp. 6073–6114, 2009. arXiv:0802.2576.

[30] J. Biamonte, M. Faccin, and M. De Domenico, "Complex networks from classical to quantum," *Communications Physics*, vol. 2, no. 1, pp. 1–10, 2019. arXiv:1702.08459.

[31] M. De Domenico and J. Biamonte, "Spectral entropies as information-theoretic tools for complex network comparison," *Physical Review X*, vol. 6, no. 4, p. 041062, 2016. arXiv:1609.01214.

[32] A. Gilyén and T. Li, "Distributional property testing in a quantum world," in *11th Innovations in Theoretical Computer Science Conference (ITCS 2020)*, Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020. arXiv:1902.00814.

[33] C. Cade and A. Montanaro, "The quantum complexity of computing Schatten p-norms," in *13th Conference on the Theory of Quantum Computation, Communication and Cryptography*, 2018. arXiv:1706.09279.

[34] B. Bauer, S. Bravyi, M. Motta, and G. K. Chan, "Quantum algorithms for quantum chemistry and quantum materials science," *arXiv preprint arXiv:2001.03685*, 2020.

[35] T. E. O'Brien, B. Tarasinski, and B. Terhal, "Quantum phase estimation of multiple eigenvalues for small-scale (noisy) experiments," *New Journal of Physics*, 2019. arXiv:1809.09697.

[36] L. Clinton, J. Bausch, and T. Cubitt, "Hamiltonian simulation algorithms for near-term quantum hardware," *arXiv preprint arXiv:2003.06886*, 2020.

[37] R. Babbush, N. Wiebe, J. McClean, J. McClain, H. Neven, and G. Chan, "Low depth quantum simulation of electronic structure," *Bulletin of the American Physical Society*, vol. 63, 2018. arXiv:1706.00023.

[38] D. W. Berry, A. M. Childs, and R. Kothari, "Hamiltonian simulation with nearly optimal dependence on all parameters," in *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pp. 792–809, IEEE, 2015. arXiv:1501.01715.

[39] A. Gilyén, Y. Su, G. H. Low, and N. Wiebe, "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*, pp. 193–204, 2019. arXiv:1806.01838.

[40] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 10th ed., 2011.

A LLED is DQC1-hard

Following the definition of [12], for any problem $L \in \mathsf{DQC1}$ and every $x \in L$, there exists a quantum circuit U of depth $T = \mathcal{O}(\text{poly}(|x|))$ that operates on $n = \mathcal{O}(\text{poly}(|x|))$ qubits such that

- $x \in L_{yes} \implies p_0 \geq \frac{1}{2} + \frac{1}{\text{poly}(|x|)}$,
- $x \in L_{no} \implies p_0 \leq \frac{1}{2} - \frac{1}{\text{poly}(|x|)}$,

where $p_0 = \text{Tr}\left[(|0\rangle\langle 0| \otimes I)U\rho U^\dagger\right]$ and $\rho = |0\rangle\langle 0| \otimes I/2^{n-1}$. From this it can be gathered that if we can estimate p_0 to within (arbitrary) $\pm 1/\text{poly}(|x|)$ precision, then we can solve L.

To show that LLED is hard for the class $\mathsf{DQC1}$ we will reduce it to the problem of estimating a kind of normalized subtrace for positive semidefinite log-local Hamiltonians, which is hard for the class $\mathsf{DQC1}$. That is, for a positive semidefinite matrix $H \in \mathbb{C}^{N\times N}$ and a threshold $b \in \mathbb{R}_{\geq 0}$ define

$$\overline{\text{Tr}_b}(H) = \frac{1}{N} \sum_{0\leq\lambda_k\leq b} \lambda_k,$$

where $\lambda_0 \leq \cdots \leq \lambda_{N-1}$ denote the eigenvalues of H, and consider the following proposition.

Proposition 4 (Brandão [22])**.** *Given as input a description of an n-qubit quantum circuit U of depth $T = \mathcal{O}(poly(n))$ together with some polynomial $r(n)$, one can efficiently construct a log-local Hamiltonian $H \in \mathbb{C}^{T2^n\times T2^n}$ and a threshold $b = \mathcal{O}(poly(n))$ such that*

$$\overline{\text{Tr}_b}(H) = p_0 \pm \frac{1}{r(n)}, \tag{5}$$

where $p_0 = \text{Tr}\left[(|0\rangle\langle 0| \otimes I)U\rho U^\dagger\right]$ and $\rho = |0\rangle\langle 0| \otimes I/2^{n-1}$. Moreover, H also satisfies:

(i) H is positive semidefinite.

(ii) There exists a $\delta = 1/poly(n)$ such that H has no eigenvalues in the interval $[b, b+\delta]$.

The Hamiltonian in the above proposition is obtained by applying Kitaev's circuit-to-Hamiltonian construction directly to the circuit U, but only constraining the input and output of the clean qubit leaving the other qubits unconstrained (thus imitating the maximally mixed state).

In what follows we will show that LLED with $1/\delta, 1/\varepsilon = \mathcal{O}(\text{poly}(n))$ allows us to estimate the quantity $\overline{\text{Tr}_b}(H)$ in Equation 5 to within arbitrary polynomial precision using a polynomial amount of non-adaptive queries to LLED with these precision parameters, plus polynomial-time classical pre-processing of the inputs and post-processing of the outcomes. In other words, we show a poly-time truth-table reduction from the problem of estimating $\overline{\text{Tr}_b}(H)$ up to arbitrary polynomial accuracy to LLED with $1/\delta, 1/\varepsilon = \mathcal{O}(\text{poly}(n))$. By Proposition 4, this shows that LLED with these precision parameters is indeed $\mathsf{DQC1}$-hard under poly-time truth-table reductions.

Our construction for estimating $\overline{\text{Tr}_b}(H)$ goes as follows. First, we divide the interval $[0, b]$ into a polynomial number of equally sized intervals. Next, we estimate the fraction of eigenvalues that lie in each of these intervals using queries to LLED. Finally, we multiply these fractions with the upper threshold of the respective intervals and compute the sum of all these quantities. Intuitively, by doing so we approximate $\overline{\text{Tr}_b}(H)$ by first constructing a histogram approximation of the low-lying spectrum of H and afterwards computing the area under this histogram.

Lemma 5. *Given as input $H \in \mathbb{C}^{T2^n \times T2^n}$ and $b = \mathcal{O}(poly(n))$ as in Proposition 4, together with some polynomial $q(n)$, one can compute a quantity Λ that, with high probability, satisfies*

$$|\Lambda - \overline{\mathrm{Tr}_b}(H)| < \frac{1}{q(n)},$$

using a polynomial number of queries to LLED with $1/\delta, 1/\varepsilon = \mathcal{O}(poly(n))$ and polynomial-time classical pre-processing of the inputs and post-processing of the outcomes.

Proof. Define $\Delta = 1/3q(n)$, $M = b/\Delta$, $\varepsilon = 1/(6Mbq(n))$ and let $\delta < \Delta/3$ be such that H has no eigenvalues in the interval $[b, b+\delta]$. Also, define the thresholds $x_j = (j+1)\Delta$, for $j = 0, \ldots, M-1$. Next, denote by $\hat{\chi}_j$ the outcome of LLED with threshold $b = x_j$ and precision parameters δ, ε as defined above. That is, $\hat{\chi}_j$ is an estimate of $\hat{y}_j$ to within additive accuracy ε, where

$$\hat{y}_j = D_H(0, x_j) + \hat{\gamma}_j, \text{ with } 0 \leq \hat{\gamma}_j \leq D_H(x_j, x_j + \delta).$$

Subsequently, define $\chi_0 = \hat{\chi}_0$, $y_0 = \hat{y}_0$ and

$$y_j = \hat{y}_j - \hat{y}_{j-1}, \tag{6}$$
$$\chi_j = \hat{\chi}_j - \hat{\chi}_{j-1}, \tag{7}$$

for $1 \leq j \leq M-1$. Finally, define the estimate

$$\Lambda = \sum_{j=0}^{M-1} \chi_j x_j. \tag{8}$$

We will now show that Λ is indeed an estimate of $\overline{\mathrm{Tr}_b}(H)$ to within additive precision $\pm 1/q(n)$. To do so, we define $\gamma_0 = \hat{\gamma}_0$ and $\gamma_j = \hat{\gamma}_j - \hat{\gamma}_{j-1}$ for $1 \leq j \leq M-1$, and we define and expand

$$\Gamma = \sum_{j=0}^{M-1} y_j x_j = \sum_{j=0}^{M-1} (D_H(x_{j-1}, x_j) + \gamma_j) x_j = \underbrace{\sum_{j=0}^{M-1} D_H(x_{j-1}, x_j) x_j}_{\mathcal{B} :=} + \underbrace{\sum_{j=0}^{M-1} \gamma_j x_j}_{\mathcal{E}_{bin} :=}.$$

We start by upper-bounding the magnitude of the $\mathcal{E}_{bin}$ term. To do so, we rewrite

$$\begin{aligned}
\mathcal{E}_{bin} &= \sum_{j=0}^{M-1} \gamma_j x_j = \hat{\gamma}_0 x_0 + \sum_{j=1}^{M-1} (\hat{\gamma}_j - \hat{\gamma}_{j-1}) x_j \\
&= \sum_{j=0}^{M-1} \hat{\gamma}_j x_j - \sum_{j=1}^{M-1} \hat{\gamma}_{j-1} x_j \\
&= \sum_{j=0}^{M-1} \hat{\gamma}_j x_j - \sum_{j=1}^{M-1} \hat{\gamma}_{j-1} (x_{j-1} + \Delta) \\
&= \sum_{j=0}^{M-1} \hat{\gamma}_j x_j - \sum_{j=1}^{M-1} \hat{\gamma}_{j-1} x_{j-1} - \Delta \sum_{j=1}^{M-1} x_{j-1} \\
&= \underbrace{\hat{\gamma}_{M-1}}_{=0} x_{M-1} - \Delta \underbrace{\sum_{j=1}^{M-1} \hat{\gamma}_{j-1}}_{\leq 1},
\end{aligned}$$

and we conclude that $|\mathcal{E}_{bin}| \leq \Delta$. Next, we upper-bound the absolute difference of $\mathcal{B}$ and $\overline{\mathrm{Tr}_b}(H)$.

$$\left|\mathcal{B} - \overline{\mathrm{Tr}_b}(H)\right| = \left|\sum_{j=0}^{M-1} D_H(x_{j-1}, x_j)x_j - \overline{\mathrm{Tr}_b}(H)\right| \leq \sum_{j=0}^{M-1} \Delta \cdot D_H(x_{j-1}, x_j) \leq \Delta.$$

Finally, we upper-bound the absolute difference between Λ and Γ.

$$|\Lambda - \Gamma| = \left|\sum_{j=0}^{M-1} (\chi_j - y_j)x_j\right| \leq \left|\sum_{j=0}^{M-1} 2\varepsilon x_j\right| \leq M \cdot 2\varepsilon \cdot b = \frac{1}{3q(n)}$$

Combining all of the above we find that

$$|\Lambda - \overline{\mathrm{Tr}_b}(H)| \leq |\Lambda - \Gamma| + |\Gamma - \overline{\mathrm{Tr}_b}(H)| \leq |\Lambda - \Gamma| + |\mathcal{B} - \overline{\mathrm{Tr}_b}(H)| + |\mathcal{E}| \leq \frac{1}{3q(n)} + \Delta + \Delta = \frac{1}{q(n)}.$$

□

B Quantum algorithms for SUES and LLED

Without loss of generality we assume that $||H|| \leq 1$. We can do so because we are promised that $\lambda_{max} := \max_k \lambda_k(H) \leq \mathrm{poly}(n)$. That is, we can rescale $H' = \lambda_{max}^{-1} H$, solve SUES and LLED for H' and rescale the obtained samples or thresholds (while still maintaining the additive inverse polynomial precision).

We will use that the DQC1 model does not change by allowing up to $\mathcal{O}(\log(n))$ clean (i.e., pure) qubits [12]. This result is important for us since the quantum algorithms we discuss will require additional ancilla qubits, which need to be initialized in the all-zeros state and hence be clean.

B.1 Quantum algorithm for SUES

The quantum algorithm for SUES uses Hamiltonian simulation to implement an approximation of the unitary e^{iH} and applies quantum phase estimation (QPE) to it with the maximally mixed state $\rho = I/2^n$ in the eigenvector register. As ρ is in a given eigenstate of H with uniform probabilities over all eigenstates of H, this will indeed output an approximation of a uniformly random sampled eigenvalue.

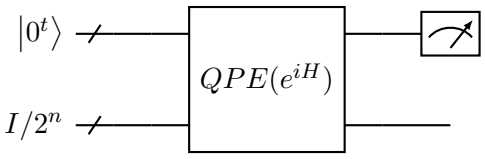

Figure 1: The circuit solving SUES for the matrix H, where $t = \mathcal{O}(\log(1/\delta) + \log(2 + 1/2\mu))$.

Let us consider the cost of this quantum algorithm. If we are given sparse access to H, then we can implement a unitary V such that

$$||V - e^{iH}|| < \gamma, \tag{9}$$

in time $\mathcal{O}(\text{poly}(n, \log(1/\gamma)))$ [38, 39]. These algorithms unfortunately require more pure qubits than are allowed in DQC1. That is, these algorithms use more than $\mathcal{O}(\log n)$ ancilla qubits, which need to be initialized in the all-zeroes state (i.e., they must be 'clean').

On the other hand, if H is log-local, then we can implement a unitary V that satisfies Equation 9 in time $\mathcal{O}(\text{poly}(n, 1/\gamma))$, while only using a constant number of ancilla qubits [33]. This makes it possible to implement this unitary V in DQC1, as we are allowed to use up to $\mathcal{O}(\log(n))$ clean qubits.

Next, denote by λ_j and ζ_j the output of the phase estimation routine when it is run using e^{iH} and V, respectively. Then we have,

$$|e^{i\lambda_j} - e^{i\zeta_j}| \leq \gamma$$

by the bound in Equation 9. With some algebra we can show that this implies that

$$|\lambda_j - \zeta_j| \leq \pi\gamma/2.$$

Therefore, if we wish to solve SUES with precision parameters $1/\delta, 1/\mu = \mathcal{O}(\text{poly}(n))$, then we let $\gamma = \delta/\pi$ and take $t = \log(2/\delta) + \lceil \log(2 + 1/2\mu) \rceil$ bits of precision in the eigenvalue register [40]. This requires a total of $\widetilde{\mathcal{O}}(2^t) = \mathcal{O}(\text{poly}(n))$ applications of the unitary V, each of which can be implemented in polynomial time as discussed above. Note that this requires only $t = \mathcal{O}(\log n)$ clean ancilla qubits in the eigenvalue register, making it possible to be done in DQC1.

B.2 Quantum algorithm for LLED

We shall describe a quantum algorithm that given a sparse positive semidefinite matrix H, two thresholds $a, b \in \mathbb{R}_{\geq 0}$ and precision parameters $\delta, \varepsilon > 0$, will output a χ that, with high probability, satisfies $|\chi - y| < \varepsilon$ with $D_H(a, b) \leq y \leq D_H(a - \delta, b + \delta)$. Clearly, this quantum algorithm will also allow us to solve LLED as defined in Section 3.1

Let $m = \mathcal{O}(\varepsilon^{-2})$ and draw for $j = 1, \ldots, m$ a sample $\overline{\lambda}_{k_j}$ from the output of SUES with $\delta/2$ as the precision parameter. Next, compute

$$\chi_j = \begin{cases} 1 \text{ if } \overline{\lambda}_{k_j} \in (a - \delta/2, b + \delta/2), \\ 0 \text{ otherwise.} \end{cases}$$

For now let us assume that all samples $\overline{\lambda}_{k_j}$ were sampled correctly, i.e., each k_j is drawn uniformly random from $\{0, \ldots, 2^n - 1\}$ and $|\lambda_{k_j} - \overline{\lambda}_{k_j}| \leq \delta/2$, where λ_{k_j} denotes the eigenvalue of which $\overline{\lambda}_{k_j}$ is an estimate. We will show that under this assumption the quantity

$$\chi := \frac{1}{m} \sum_{j=1}^{m} \chi_j$$

will be, with high probability, a correct solution to LLED.

By the Chernoff-Hoeffding inequality χ will, with probability exponentially close to 1, be ε-close to

$$y := \Pr\left[\overline{\lambda} \in (a - \delta/2, b + \delta/2)\right],$$

where the probability is taken over the $\overline{\lambda}$ being *correctly* sampled from the output of SUES.

Since a correctly sampled $\overline{\lambda}$ satisfies $|\lambda - \overline{\lambda}| \leq \delta/2$, where λ denotes the eigenvalue of which $\overline{\lambda}$ is an estimate, we find that

(i) $\Pr\left[\overline{\lambda} \in (a-\delta/2, b+\delta/2)\right] \leq \Pr\left[\lambda \in (a-\delta, b+\delta)\right] = D_H(a-\delta, b+\delta)$,

(ii) $\Pr\left[\overline{\lambda} \in (a-\delta/2, b+\delta/2)\right] \geq \Pr\left[\lambda \in (a, b)\right] = D_H(a, b)$,

where the probabilities on the right hand side of the equations are taken over the λ being uniformly sampled from the eigenvalues of H.

Combining this with the Chernoff-Hoeffding inequality we find that, with probability exponentially close to 1, we have $|\chi - y| < \varepsilon$, where y satisfies

$$D_H(a, b) \leq y \leq D_H(a-\delta, b+\delta).$$

Now let us consider the probability that all our samples $\overline{\lambda}_{k_j}$ were indeed sampled correctly. By the union bound this probability is at least $1 - m\mu$. Since $m = \mathcal{O}\big(\varepsilon^{-2}\big)$, we can choose $\mu = 1/\mathrm{poly}(\varepsilon^2, n)$ such that all our samples are correct with probability polynomially close 1.

As discussed in Section B.1, SUES with $1/\delta, 1/\mu = \mathcal{O}(\mathrm{poly}(n))$ is in DQC1 when restricted to log-local Hamiltonians. As the above quantum algorithm for LLED uses $\mathcal{O}\big(\varepsilon^2\big)$ samples from SUES and some classical post-processing, we conclude LLED with $1/\delta, 1/\varepsilon = \mathcal{O}(\mathrm{poly}(n))$ is also in DQC1 when restricted to log-local Hamiltonians.