

# Quantum algorithm for persistent Betti numbers and topological data analysis

パーシステント・ベッチ数およびトポロジカルデータ解析に関する量子アルゴリズム

早川 龍

# Quantum algorithm for persistent Betti numbers and topological data analysis

Ryu Hayakawa

Yukawa Institute for Theoretical Physics, Kyoto University, Kitashirakawa Oiwakecho, Sakyo, Kyoto 606-8502, Japan

Topological data analysis (TDA) is an emergent field of data analysis. The critical step of TDA is computing the persistent Betti numbers. Existing classical algorithms for TDA are limited if we want to learn from high-dimensional topological features because the number of high-dimensional simplices grows exponentially in the size of the data. In the context of quantum computation, it has been previously shown that there exists an efficient quantum algorithm for estimating the Betti numbers even in high dimensions. However, the Betti numbers are less general than the persistent Betti numbers, and there have been no quantum algorithms that can estimate the persistent Betti numbers of arbitrary dimensions.

This paper shows the first quantum algorithm that can estimate the (*normalized*) persistent Betti numbers of arbitrary dimensions. Our algorithm is efficient for simplicial complexes such as the Vietoris-Rips complex and demonstrates exponential speedup over the known classical algorithms.

## 1 Introduction

Processing large data has become a highly demanding task in modern society. Topological data analysis (TDA) is an emergent field of information processing based on topological and geometric methods [13, 40]. In particular, TDA based on persistent homology [6, 14, 15, 32, 42] has been gathering a lot of attention recently. TDA can reveal the “shape of data” with multi-scale analysis of topological features. A crucial property of persistent homology, which is favorable in practical situations, is its stability against the noise concerning how the data is sampled [8]. There have been numerous applications of TDA, including the protein structure [41], the quantum many-body dynamics [37], the cosmic web [33], and the string theory [9], to name a few. Recently, combinations of persistent homology with supervised or unsupervised machine learning methods have also been actively studied [34].

In the procedure of TDA, data is first converted into a nested sequence of topological objects. Such a topological object is called a *simplicial complex*, and a nested sequence of simplicial complexes is called *filtration*. We can then study how topological invariants, such as the number of connected components, holes, voids, and their high-dimensional counterparts, change within the filtration using persistent homology. Typically, topological invariants that persist longer are considered to be more “significant” than those that persist shorter. The behavior of topological invariants is summarized as diagrams or functions in the form such as the persistent Barcodes [17] and the persistent Landscapes [5]. Therefore,

Ryu Hayakawa: [ryu.hayakawa@yukawa.kyoto-u.ac.jp](mailto:ryu.hayakawa@yukawa.kyoto-u.ac.jp)

a typical procedure of TDA consists of first constructing a family of topological objects from the given data and then computing the persistence of topological invariants, i.e. computing the persistent Betti numbers. (For the formal definition of persistent Betti numbers, see Section 2).

There are challenges in classical TDA that arise from the combinatorial nature of the topological treatment of data. Assume that the input data is composed of  $n$  points. Then, the number of possible  $q$ -dimensional elements (i.e.  $q$ -simplices) that can contribute to topology is  $\binom{n}{q+1}$ . (We say a simplex is  $q$ -dimensional if it is composed of  $q + 1$  vertices.) The known classical algorithms for persistent homology work in polynomial time *in the number of simplices* [30, 42]. Classical algorithms work in polynomial time in  $n$  for constant dimensional cases because the number of constant-dimensional simplices is at most polynomial in  $n$ . On the other hand, there can be exponentially many simplices for super constant dimensions. As a result, performing a high-dimensional analysis of TDA is a difficult task for classical computers. There are many studies to overcome this issue such as the way of reducing the size of simplicial complexes while keeping their homology unchanged [2]. However, such techniques are not enough to overcome the exponential growth of simplices in high dimensions.

**Existing quantum approaches for TDA.** Data learning tasks are also actively studied in the field of quantum computation. In order to investigate the quantum computational approaches for TDA, Lloyd, Garnerone, and Zanardi [27] proposed a quantum algorithm (LGZ algorithm) for estimating the (normalized)<sup>1</sup> *Betti numbers*. The  $q$ -th Betti number represents the number of “ $q$ -dimensional holes” of a simplicial complex. By estimating the Betti numbers for simplicial complexes that arise from each step of the filtration, the LGZ algorithm can be used to learn the topological structures of data. A function that represents how the Betti numbers vary within the filtration is called a *Betti function* or a *Betti curve* [6]. Therefore, the LGZ algorithm can approximately recover the Betti curve of arbitrary dimensions. The LGZ algorithm demonstrates an exponential speedup over the known best classical algorithm for estimating the high-dimensional Betti numbers. A proof of principle experiment of the LGZ algorithm has also been done [24].

Estimating the Betti numbers for each element of the filtration is not enough to fully perform TDA based on persistent homology. To perform such a full TDA, it is crucial to estimate the *persistent Betti numbers*. The  $q$ -th  $(t, s)$ -persistent Betti number is the number of  $q$ -dimensional holes that exist at scale parameter  $t$  that are still alive at  $s$ . (For a formal definition, see Section 2.) Because the  $(t, t)$ -persistent Betti number is just the usual Betti number, estimating the persistent Betti numbers is a more general task than estimating the Betti numbers. It is shown in [29, 31] that the LGZ algorithm can be used to estimate the persistent Betti numbers of the zeroth dimension but fail in higher dimensions in general. The TDA of the zeroth dimension is not the region where we can exploit the benefit of quantum computation because the classical algorithms run in polynomial time for constant dimensions. Therefore, constructing a quantum algorithm for persistent Betti numbers for arbitrary dimensions has been a significant open problem for the quantum TDA.

**Our result.** We present a first quantum algorithm for estimating the persistent Betti numbers of arbitrary dimensions. We provide error and complexity analysis and the condition for the algorithm to succeed. The formal statement can be seen in Theorem 7 in

---

<sup>1</sup>“Normalized” means that Betti number is divided by the total number of  $q$ -dimensional elements of the simplicial complex.

Section 4. We resolve the open problem of estimating the persistent Betti numbers by estimating the spectral density of the ground states of a certain positive-semidefinite hermitian operator using quantum computation. The positive-semidefinite operator is called *persistent combinatorial Laplacian* [30, 39]. (The persistent combinatorial Laplacian is different from the combinatorial Laplacian, which is used in the LGZ algorithm.) Estimating the persistent Betti numbers allows one to recover the topological summary of data such as the persistent Barcodes.

**Comparison with the classical TDA.** The best known classical algorithm for computing the persistent Betti numbers of a simplicial pair  $K \hookrightarrow L$  [30] runs in time  $\mathcal{O}((n_q^L)^\omega)$  for  $\omega < 2.373$ , where  $n_q^L$  is the number of  $q$ -simplices of  $L$  (assuming  $n_q^L = \mathcal{O}(n_{q+1}^L)$ , which holds in many practical cases such as for Vietoris-Rips or Čech complexes). For an approximation task, the best-known classical algorithms for estimating the spectral densities of matrices [7, 11, 26] runs time linearly with the number of non-zero elements of the matrices. The known classical algorithm for computing the matrix representation of the persistent Laplacian [30] takes  $\mathcal{O}((n_q^L)^\omega + n_{q+1}^L)$  time. Therefore, there are no known classical algorithms for computing or approximating the persistent Betti numbers efficiently when  $n_q^L$  is exponential in the number of vertices  $n$ .

We present our quantum algorithm using the membership oracle for simplicial complexes. Therefore, it is necessary that we can efficiently construct the membership function so that we can efficiently run the quantum algorithm. Indeed, we can efficiently implement the membership functions for the construction of simplicial complexes such as the Vietoris-Rips complex and the lazy witness complex, as we see in Appendix B. Therefore, our quantum algorithm demonstrates exponential speedup over the best-known classical algorithm for instances of problems that satisfy the promises of Theorem 7.

Nevertheless, the drawback of our result is that we can only estimate the normalized version of the persistent Betti numbers. It is not known whether the estimation of such normalized persistent Betti number is useful or not. It is an important future direction to understand whether there are practical instances of problems that satisfy the conditions of Theorem 7 and whether the estimation of the normalized persistent Betti number is useful or not.

**Related Work.** The LGZ algorithm [27] estimates the Betti numbers using the quantum phase estimation algorithm. Based on our quantum algorithm for the persistent Betti numbers, we can immediately provide a quantum algorithm for estimating the normalized Betti numbers based on the block-encoding and QSVT by slightly modifying our algorithm (because Betti numbers are special cases of persistent Betti numbers). The gate complexity of implementing a quantum algorithm which outputs 1 with probability  $\tilde{p}_1$  s.t.

$$|\tilde{p}_1 - \beta_q^K/n_q^K| \leq \epsilon$$

is  $\mathcal{O}(\text{poly}(n) \log(1/\epsilon))$ , which is an exponential improvement in terms of the error parameter compared to the LGZ algorithm. (Nonetheless, this exponential improvement does not actually improve the complexity in terms of error due to the sampling error.)

Recently, a NISQ algorithm for the Betti numbers is presented in [38]. Their algorithm is based on the following observations: (a) a boundary map can be represented as a sum of Pauli operators; (b) Instead of using Grover’s quantum search to implement the uniform mixture of the  $q$ -simplex states, a quantum rejection sampling can be used; (c) a stochastic rank estimation can be used to estimate the Betti numbers instead of the quantum phase estimation. Based on these observations, their quantum algorithm achieves a depth

complexity that is linear in  $n$  at the cost of the success probability  $1/d_q^K$  of the rejection sampling. It is an open problem to construct a quantum algorithm for the persistent Betti numbers in a way that is preferable for NISQ devices.

**Organization.** The rest of the paper is organized as follows. In Section 2, we give the preliminaries on persistent homology and TDA. In Section 3, we review the block-encoding and quantum singular value transformation. In Section 4, we provide a quantum algorithm for estimating the persistent Betti numbers, which is our main result. In Section 5 and Section 6, we present the implementation of some unitaries which are required to conclude the proof of the main result.

## 2 Preliminaries on persistent homology

In this section, we introduce persistent homology for simplicial complexes. We first introduce some necessary terminologies such as the simplicial complex, the filtration, the Betti numbers, and the persistent Betti numbers. Then we introduce the recent result of [30] that the persistent Betti number can be calculated as the nullity of the so-called persistent Laplacian operator (Theorem 1) and the persistent Laplacian can be implemented using the Schur complement (Theorem 2). Table 1 may be also helpful to overview the notations.

An abstract simplicial complex  $K$  over a finite ordered set  $V$  of vertices is a collection of subsets of  $V$  such that for any  $\sigma \in K$ , if  $\tau \subseteq \sigma$  then  $\tau \in K$ . An element  $\sigma \in K$  is called a  $q$ -simplex if  $|\sigma| = q + 1$ . For example, a 0-simplex is called a vertex, and a 1-simplex is called a line. We denote the set of  $q$ -simplices of  $K$  by  $S_q^K$ .

An oriented simplex is a simplex in  $K$  with a fixed ordering of vertices inherited from the ordering of  $V$ . We denote an oriented simplex of  $\sigma \in K$  as  $\tilde{\sigma}$ . Let  $\bar{S}_q^K := \{\tilde{\sigma} : \sigma \in S_q^K\}$  be a set of oriented  $q$ -simplices of  $K$ . The  $q$ -th chain group  $C_q^K$  of  $K$  is the vector space over  $\mathbb{R}$  with basis  $\bar{S}_q^K$ <sup>2</sup>. Let  $n_q^K := \dim(C_q^K) = |S_q^K|$  be the dimension of  $C_q^K$ .

The boundary map  $\partial_q^K : C_q^K \rightarrow C_{q-1}^K$  is defined as

$$\partial_q^K([v_0, \dots, v_q]) := \sum_{i=0}^q (-1)^i [v_0, \dots, \hat{v}_i, \dots, v_q] \quad (1)$$

for each  $\tilde{\sigma} = [v_0, \dots, v_q]$ , where the hat indicates that the vertex  $v_i$  is omitted. Then, the  $q$ -th homology group of  $K$  is

$$H_q(K) := \ker(\partial_q^K) / \text{im}(\partial_{q+1}^K),$$

and the  $q$ -th Betti number of  $K$  is

$$\beta_q^K := \text{rank}(H_q(K)).$$

The  $q$ -th combinatorial Laplacian  $\Delta_q^K : C_q^K \rightarrow C_q^K$  is defined as follows:

$$\Delta_q^K := \underbrace{\partial_{q+1}^K \circ (\partial_{q+1}^K)^*}_{=: \Delta_{q,\text{up}}^K} + \underbrace{(\partial_q^K)^* \circ \partial_q^K}_{=: \Delta_{q,\text{down}}^K},$$

---

<sup>2</sup>We work with chain groups with coefficients  $\mathbb{R}$ , which is suitable for our quantum algorithm that relies on the Hodge theory and the persistent Laplacian. The classical algorithms for persistent homology of [42] work for field coefficients such as  $\mathbb{Z}_2$ . The Betti numbers of homology groups over  $\mathbb{R}$  coefficients do not capture the torsion [25]. Nonetheless, this seems to be enough for many of the situations in TDA.

where  $(\partial_q^K)^* : C_{q-1}^K \rightarrow C_q^K$  is the adjoint of  $\partial_q^K$  under the inner product  $\langle \cdot, \cdot \rangle_{C_q^K}$  such that  $\bar{S}_q^K$  is an orthonormal basis of  $C_q^K$ . We have introduced  $\Delta_{q,\text{up}}^K, \Delta_{q,\text{down}}^K$  for the convenience of notation. The Betti number can be calculated as the nullity of the combinatorial Laplacian [16]:

$$\beta_q^K = \text{nullity}(\Delta_q^K) = \# \text{ of zero eigenvalues of } \Delta_q^K. \quad (2)$$

This relationship comes from the following elementary result of Hodge theory:

**Claim 1** ([25]) *Let  $A \in \mathbb{R}^{m \times n}$  and  $B \in \mathbb{R}^{n \times p}$ , and assume  $AB = 0$ . Then we have*

$$\ker(A)/\text{im}(B) \cong \ker(BB^* + A^*A).$$

Eq. (2) follows using this claim observing that  $\partial_q^K \partial_{q+1}^K = 0$ .

## 2.1 Filtration and Persistent homology

A sequence of nested subcomplexes of  $F$

$$\emptyset = F_0 \subseteq F_1 \subseteq \dots \subseteq F_m = F$$

is called filtration. Similarly, we say a pair of simplicial complexes  $K$  and  $L$  is a simplicial pair if  $K$  and  $L$  are the simplicial complexes over the same ordered set  $V$  such that  $K \subseteq L$ . We denote a simplicial pair by  $K \hookrightarrow L$ . For a simplicial pair  $K \hookrightarrow L$ , we choose the ordering  $\{\sigma_q^L(i)\}_{i=1}^{n_q^L}$  such that  $\{\sigma_q^L(i)\}_{i=1}^{n_q^K} = \{\sigma_q^K(i)\}_{i=1}^{n_q^K}$ . We obtain a map between the  $q$ -th homology group of  $K$  and  $L$  induced by the inclusion:

$$f_q^{K,L} : H_q(K) \rightarrow H_q(L).$$

The  $q$ -th persistent homology group  $H_q^{K,L}$  is defined as the image of the map. The  $q$ -th persistent Betti number  $\beta_q^{K,L}$  is defined as the rank of the map. The persistent homology group consists of the homology group of  $K$  that are still alive at  $L$ . That is,

$$H_q^{K,L} = \ker(\partial_q^K) / \left( \text{im}(\partial_{q+1}^L) \cap \ker(\partial_q^K) \right).$$

Note that the difference of  $q$ -th Betti numbers of  $K$  and  $L$  ( $\beta_q^L - \beta_q^K$ ) does not reveal the same information as  $\beta_q^{K,L}$  because  $\beta_q^{K,L}$  is the number of topological invariants that exist at  $K$  that are still alive at  $L$ .

## 2.2 Persistent Laplacian

The persistent Laplacian is first introduced in [39]. Later, the property and implementations of the persistent Laplacian are further studied in [30]. Here, we introduce the  $q$ -th persistent Laplacian for a persistent pair  $K \hookrightarrow L$ . Consider the subspace of  $C_q^L$

$$C_q^{L,K} := \{c \in C_q^L : \partial_q^L(c) \in C_{q-1}^K\}.$$

Then, let  $\partial_q^{L,K}$  be the restriction of the boundary operator  $\partial_q^L$  to  $C_q^{L,K}$  so that the image of  $\partial_q^{L,K}$  is contained in  $C_{q-1}^K$ . Then, the  $q$ -th persistent Laplacian is defined as

$$\Delta_q^{K,L} := \underbrace{\partial_{q+1}^{L,K} \circ (\partial_{q+1}^{L,K})^*}_{=: \Delta_{q,\text{up}}^{K,L}} + \underbrace{(\partial_q^K)^* \circ \partial_q^K}_{=: \Delta_{q,\text{down}}^K}.$$

It is shown in [30] that the nullity of the  $q$ -th persistent Laplacian equals to the  $q$ -th persistent Betti numbers:

**Theorem 1 ([30])** For each integer  $q \geq 0$ ,  $\beta_q^{K,L} = \text{nullity}(\Delta_q^{K,L})$ .

We provide a proof in Appendix A for completeness.

In addition to the persistent Betti numbers from the nullity of the persistent Laplacian, it is suggested in [39] that the spectral property of the persistent Laplacian beyond the zero eigenvalues would provide useful information about the data.

**Matrix representation.** We can give a matrix representation of  $\partial_q^K$  with respect to the canonical basis  $\{\sigma_q^K(i)\}_{i \in [n_q^K]}$  of  $C_q^K$  and  $\{\sigma_{q-1}^{K-1}(i)\}_{i \in [n_{q-1}^{K-1}]}$  of  $C_{q-1}^{K-1}$  according to eq. (1). The matrix representation  $B_q^K$  of  $\partial_q^K$  is  $n_{q-1}^{K-1} \times n_q^K$  matrix with entries from  $\{-1, 0, +1\}$ . For example, if the canonical basis of  $C_1^K$  is  $\{01, 02, 03, 12\}$  and the canonical basis of  $C_0^K$  is  $\{0, 1, 2, 3\}$ , the boundary matrix is represented as

$$B_1^K = \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} 01 \quad 02 \quad 03 \quad 12 \\ \left[ \begin{array}{cccc} -1 & -1 & -1 & 0 \\ 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{array} \right] \end{array}.$$

$B_q^K$  is  $(q+1)$ -column sparse by definition. The matrix representation of  $(\partial_q^K)^*$  is simply  $(B_q^K)^\top$ . Choose any basis of  $C_{q+1}^{L,K}$  represented by a column matrix  $Z \in \mathbb{R}^{n_{q+1}^L \times n_{q+1}^{L,K}}$  and let  $B_{q+1}^{L,K}$  be the corresponding matrix representation of  $\partial_{q+1}^{L,K}$ . It is shown in [30] that the matrix representation of  $(\partial_{q+1}^{L,K})^*$  is  $(Z^\top Z)^{-1} (B_{q+1}^{L,K})^\top$ . Then the matrix representation of the persistent Laplacian is

$$\Delta_q^{K,L} = B_{q+1}^{L,K} (Z^\top Z)^{-1} (B_{q+1}^{L,K})^\top + (B_q^K)^\top B_q^K. \quad (3)$$

In [30], the authors have given two algorithms for computing the matrix representation of  $\Delta_q^{K,L}$ . We adopt the second algorithm which uses the Schur complement. The Schur complement is defined as follows.

**Definition 1 (Schur complement)** Let  $M \in \mathbb{R}^{n \times n}$  be a block matrix  $M = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$

where  $D \in \mathbb{R}^{d \times d}$  is a block matrix. The Schur complement of  $D$  in  $M$  is

$$M/D := A - BD^+C,$$

where  $D^+$  is the Moore-Penrose pseudo-inverse of  $D$ . Similarly, for  $\emptyset \neq I \subseteq [n]$  and  $\emptyset \neq J \subseteq [n]$ , we denote the submatrix of  $M$  consisting of rows and columns of indices of  $I$  and  $J$  by  $M(I, J)$ . Then, the Schur complement of  $M(I, I)$  in  $M$  is

$$M/M(I, I) := M([n] \setminus I, [n] \setminus I) - M([n] \setminus I, I)M(I, I)^+M(I, [n] \setminus I),$$

where  $[n] \setminus I := \{a \in [n] \mid a \notin I\}$  and  $M(I, J)$  is the submatrix of  $M$  that consists of rows and columns of  $M$  indexed by  $I$  and  $J$ , respectively.

It is shown in [30] that  $\Delta_{q,\text{up}}^{K,L}$  can be implemented using the Schur complement:

**Theorem 2 ([30])** Let  $K \hookrightarrow L$  be a simplicial pair. Assume that  $n_q^K < n_q^L$  and let  $I_K^L := [n_q^L] \setminus [n_q^K]$ . Then,

$$\Delta_{q,\text{up}}^{K,L} = \Delta_{q,\text{up}}^L / \Delta_{q,\text{up}}^L(I_K^L, I_K^L).$$

**Remark 1** Finding the orthonormal basis of  $C_q^K$  is non-trivial and therefore building  $B_q^K$  is also non-trivial. If the basis can be found (by matrix reduction, see [30] for detail), we can find  $Z$  and  $B_q^{L,K}$ , and compute  $\Delta_q^{K,L} = B_{q+1}^{L,K} (Z^\top Z)^{-1} (B_{q+1}^{L,K})^\top + (B_q^K)^\top B_q^K$ . For the quantum algorithm we present in this paper, we do not need to classically compute the matrix representation of  $\Delta_q^{K,L}$  because we can bypass the necessity of computing the matrix representation of  $\Delta_q^{K,L}$  by building the block-encoding of  $\Delta_q^{K,L}$  using the quantum membership functions assisted by the method of [30], which uses the Schur complement.

### 2.3 Correspondence with Quantum computation

We can use the Hilbert space of  $n$ -qubits to represent a chain complex of a simplicial complex over  $n$ -vertices. We can correspond a  $q$ -simplex to an  $n$ -bit string of Hamming weight  $q + 1$ , where the indices of 1s correspond to the indices of vertices of a simplex. We denote  $|\tilde{\sigma}_q\rangle$  where  $\tilde{\sigma}_q$  is an  $n$ -bit string with Hamming weight  $q + 1$ . We denote the  $\binom{n}{q+1}$  dimensional Hilbert space in  $\mathbb{C}^{2^n}$  spanned by computational basis states of Hamming weight  $q + 1$  by  $\mathcal{W}_q$ . Let  $\mathcal{H}_q^K$  be the subspace of  $\mathcal{W}_q$  spanned by  $\{|\tilde{\sigma}_q\rangle : \tilde{\sigma}_q \in \bar{S}_q^K\}$ . We call  $\{|\tilde{\sigma}_q\rangle : \tilde{\sigma}_q \in \bar{S}_q^K\}$  the canonical basis of  $\mathcal{H}_q^K$ . The superposition of quantum states  $\{|\tilde{\sigma}_q\rangle : \tilde{\sigma}_q \in \bar{S}_q^K\}$  is the quantum state representation of the  $q$ -chain. The boundary operator maps as  $\partial_q^K : \mathcal{H}_q^K \rightarrow \mathcal{H}_{q-1}^K$ . A summary of notations is given in Table 1.

Table 1: Summary of notations

$K \hookrightarrow L$	A simplicial pair
$n$	The number of vertices
$\bar{S}_q^K$	The set of oriented $q$ -simplices in $K$
$C_q^K$	The $q$ -th chain group of $K$ with coefficients $\mathbb{R}$
$n_q^K$	$\dim(C_q^K) =  S_q^K $
$\{\sigma_q^K(i)\}_{i \in [n_q^K]}$	canonical basis of $C_q^K$
$\mathcal{W}_q$	Subspace of $\mathbb{C}^{2^n}$ spanned by Hamming weight $q + 1$ states
$\mathcal{H}_q^K$	Subspace of $\mathcal{W}_q$ spanned by $\{ \sigma_q^K(i)\rangle\}_{i \in [n_q^K]}$
$\{ \psi_q(i)\rangle\}_{i \in [n_q^K]}$	Eigenbasis of $\Delta_q^{K,L}$ (in $\mathcal{H}_q^K$ ) with eigenvalues $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_{n_q^K}$

## 3 Block-encoding and quantum singular value transformation

In this section, we review the block-encoding of quantum operators and the quantum singular value transformation (QSVT). We refer the reader to [18, 28] for more detail. We use  $\|\cdot\|$  as the spectral norm and  $\|\cdot\|_\diamond$  as the diamond norm. The diamond norm is defined as  $\|\Lambda\|_\diamond := \max_\rho \|(\Lambda \otimes \mathcal{I})(\rho)\|_1$ , where  $\|\cdot\|_1$  is the trace norm. We also use the notation of  $\tilde{O}(\cdot)$  which omits  $\text{poly}(\log n)$  and  $\text{poly}(\log(\log \frac{1}{\epsilon}))$  factors.

### 3.1 Block-encoding and QSVT

Let  $A = \tilde{\Pi}U\Pi$ , where  $\tilde{\Pi}$ ,  $\Pi$  are orthogonal projectors and  $U$  is a unitary. We call such  $U$  a projected unitary encoding of  $A$ . Especially, a block-encoding of a quantum operator is defined as follows:



**Definition 2 (Block-encoding [18])** Suppose that  $A$  is an  $s$ -qubit operator,  $\alpha, \epsilon \in \mathbb{R}_+$ , and  $a \in \mathbb{N}$ . We say that the  $(s+a)$ -qubit unitary  $U$  is an  $(\alpha, a, \epsilon)$ -block-encoding of  $A$ , if

$$\|A - \alpha(|0\rangle^{\otimes a} \otimes I^{\otimes s})U(|0\rangle^{\otimes a} \otimes I^{\otimes s})\| \leq \epsilon.$$

Here,  $I$  is the 1-qubit identity operator.

Such encoding is called a block-encoding because  $U$  approximates the block matrix

$$\begin{bmatrix} A/\alpha & \cdot \\ \cdot & \cdot \end{bmatrix}.$$

It is shown in [18] how to implement a unitary which realizes polynomial transformation of the singular values of the block-encoded matrices. Let  $A = \tilde{\Pi}U\Pi$ ,  $d := \text{rank}(\Pi)$ ,  $\tilde{d} := \text{rank}(\tilde{\Pi})$  and  $d_{\min} := \min(d, \tilde{d})$ . Then by the singular value decomposition, there exists orthonormal bases  $\{|\psi_i\rangle : i \in [d]\}$  and  $\{|\tilde{\psi}_i\rangle : i \in [\tilde{d}]\}$  of  $\text{img}(\Pi)$  and  $\text{img}(\tilde{\Pi})$  and  $a_i \in \mathbb{R}_0^+$  such that

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

where  $a_1 \geq a_2 \geq \dots \geq a_{d_{\min}}$ . Then, the quantum singular value transformation (QSVT) of  $A$  is defined for odd or even polynomial function  $f : \mathbb{R} \rightarrow \mathbb{C}$  as

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

The following is shown in [18]. This is a slightly modified version of Corollary 18 of [18].

**Theorem 3 (Singular Value Transformation by real polynomial)** Let  $\mathcal{H}_U$  be a finite dimensional Hilbert space and let  $U, \Pi, \tilde{\Pi}$  be linear operators that map  $\mathcal{H}_U \rightarrow \mathcal{H}_U$  such that  $U$  is a unitary, and  $\Pi, \tilde{\Pi}$  are orthogonal projectors. Suppose  $f \in \mathbb{R}[x]$  be a degree- $d$  odd or even polynomial satisfying  $|f(x)| \leq 1$  for all  $x \in [-1, 1]$ . Then, there exists an  $\mathcal{O}(\text{poly}(d, \log(1/\delta)))$ -time classical algorithm that outputs the classical description of a quantum circuit  $U_{\Phi^+}$  and  $U'_{\Phi^+}$  such that

$$P_{\mathfrak{R}}^{(\text{SV})}(\tilde{\Pi}U\Pi) = \begin{cases} (|0\rangle \otimes \tilde{\Pi})U_{\Phi^+}(|0\rangle \otimes \Pi), & \text{if } d \text{ is odd,} \\ (|0\rangle \otimes \tilde{\Pi})U'_{\Phi^+}(|0\rangle \otimes \Pi), & \text{if } d \text{ is even} \end{cases}$$

and

$$\|f^{(\text{SV})}(\tilde{\Pi}U\Pi) - P_{\mathfrak{R}}^{(\text{SV})}(\tilde{\Pi}U\Pi)\| \leq \delta.$$

Moreover,  $U_{\Phi^+}$  and  $U'_{\Phi^+}$  can be implemented using  $\mathcal{O}(d)$  use of  $U, U^\dagger, C_{\Pi}\text{NOT}, C_{\tilde{\Pi}}\text{NOT}$  and other single qubit gates, where  $C_{\Pi}\text{NOT}$  is defined as

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

We give a proof in Appendix A. The corollary follows when  $U$  is a block-encoding of  $A$ .

**Corollary 1 (QSVT by real polynomial for block-encoded operator)** Let  $U$  be an  $(\alpha, a, 0)$ -block-encoding of an  $n$ -qubit operator  $A$ . Let  $f \in \mathbb{R}[x]$  be a degree- $d$  odd or even polynomial satisfying  $|f(x)| \leq 1$  for all  $x \in [-1, 1]$ . Then, there exists an  $\mathcal{O}(\text{poly}(d, \log(1/\delta)))$ -time classical algorithm that outputs a description of a quantum circuit  $U_{\Phi^+}$  s.t.  $U_{\Phi^+}$  is a  $(1, a+1, \delta)$ -block encoding of  $f^{(\text{SV})}(A/\alpha)$ . Moreover,  $U_{\Phi^+}$  can be implemented with  $\mathcal{O}(d)$  use of  $U, U^\dagger$  and  $\mathcal{O}(ad)$  other elementary quantum gates.

### 3.2 QSVT with some useful polynomials

In this paper, we use QSVT with three different polynomials. First, we introduce the fixed-point amplitude amplification using polynomial approximations of the sign function (Lemma 25 of [18]):

**Theorem 4 (Fixed-point amplitude amplification [18])** *Let  $U$  be a unitary and  $\Pi$  be an orthogonal projector such that  $\Pi U|\psi_0\rangle = a|\psi_G\rangle$ , and  $a > \delta > 0$ . There is a unitary circuit  $\tilde{U}$  such that*

$$\left\| \tilde{U}|\psi_0\rangle - |\psi_G\rangle \right\| \leq \epsilon,$$

*which uses a single ancilla qubit and consists of  $\mathcal{O}(\log(1/\epsilon)/\delta)$ -use of  $U$ ,  $U^\dagger$ ,  $C_\Pi$ NOT,  $C_{|\psi_0\rangle\langle\psi_0|}$ NOT and other quantum gates.*

We use this theorem for state preparation of our algorithm in Section 5.

Next, we introduce the implementation of the Moore-Penrose pseudo-inverse. Suppose  $\tilde{\Pi}U\Pi = A$  and  $A = W\Sigma V^\dagger$  is a singular value decomposition, where  $\Sigma$  is a diagonal matrix with non-negative and non-increasing entries. Then the pseudo-inverse of  $A$  is defined as  $A^+ := V\Sigma^+W^\dagger$ , where  $\Sigma^+$  contains the inverse of the diagonal elements of  $\Sigma$  except for 0, which remains 0. We can implement the pseudo-inverse using the polynomial approximation of  $1/x$  [18]:

**Theorem 5 (Implementing the Moore-Penrose pseudo-inverse)** *Suppose that  $A$  is an  $n$ -qubit positive-semidefinite hermitian operator and  $U$  is an  $(\alpha, a, 0)$ -block-encoding of  $A$ . Assume that the smallest non-zero eigenvalue of  $A$  is  $\lambda_{\min}$  and let  $0 < 1/\kappa < \lambda_{\min}/\alpha$  and  $0 < \epsilon < 1/2$ . Then there is a  $d = \mathcal{O}(\kappa \log(\frac{\kappa}{\alpha\epsilon}))$  and an efficiently computable  $\Phi \in \mathbb{R}^d$  such that  $U_{\Phi^+}$  is a  $(2\kappa/\alpha, a + 1, \epsilon)$ -block-encoding of  $A^+$ .*

This is a slight modification of Theorem 41 of [18] for the block-encoding of the positive-semidefinite operator. A proof is given in Appendix A. We also use QSVT with a polynomial approximation of the rectangle function in Section 6.2 in order to implement the projector.

### 3.3 Block-measurement

Finally, we review the block-measurement which is introduced in [35]. We can implement a quantum channel that is close to the following map

$$|0\rangle \otimes |\psi\rangle \rightarrow |1\rangle \otimes \Pi|\psi\rangle + |0\rangle \otimes (I - \Pi)|\psi\rangle \quad (4)$$

using the approximate block-encoding of  $\Pi$ . Let  $U_\Pi$  be the *exact* block-encoding of  $\Pi$  with  $m$  ancilla qubits. Then, consider the following circuit  $V$ .

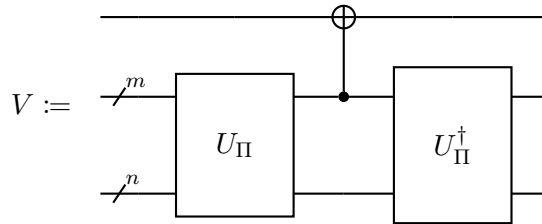


Figure 1: The quantum circuit for block-measurement.

The CNOT gate refers to  $X \otimes |0^m\rangle\langle 0^m| + I \otimes (I_m - |0^m\rangle\langle 0^m|)$  where  $X$  is the Pauli- $X$  matrix. It can be seen that  $V$  satisfies

$$(I \otimes \langle 0^m| \otimes I_n)V(|0\rangle \otimes |0^m\rangle \otimes I_n) = |1\rangle \otimes \Pi + |0\rangle \otimes (I_n - \Pi).$$

This means that  $V$  is a block-encoding of

$$X \otimes \Pi + I \otimes (I_n - \Pi)$$

if  $U_\Pi$  is a block-encoding of  $\Pi$  with no error. Let us define the following quantum channel:

$$\Lambda_\Pi(\rho) := \sum_{i \in \{0,1\}^m} (I \otimes \langle i| \otimes I_n)V(|0^{m+1}\rangle \langle 0^{m+1}| \otimes \rho)V^\dagger(I \otimes |i\rangle \otimes I_n).$$

We also define

$$\Lambda_i(\rho) := (I \otimes \langle i| \otimes I_n)V(|0^{m+1}\rangle \langle 0^{m+1}| \otimes \rho)V^\dagger(I \otimes |i\rangle \otimes I_n).$$

$\Lambda_0(\rho)$  is the quantum channel of (4). Consider the case that what we can implement is not an exact block-encoding of  $\Pi$  but an approximate one. Let  $\tilde{V}$  and  $\tilde{\Lambda}_\Pi$  be the quantum circuit and channel which are similarly implemented as  $V$  and  $\Lambda_\Pi$  using  $\tilde{U}_\Pi$  instead of  $U_\Pi$ . The block-measurement theorem gives an upper bound on the diamond norm between  $\tilde{\Lambda}_\Pi$  and ideal quantum channel  $\Lambda_0$  :

**Theorem 6 (Block-measurement [35])** *Let  $\Pi$  be a projector,  $V_\Pi = X \otimes \Pi + I \otimes (I_n - \Pi)$  be a unitary and  $A$  be a hermitian matrix satisfying  $\|\Pi - A^2\| \leq \epsilon$ . Also,  $A$  has a block-encoding  $\tilde{U}_\Pi$ . Then the channel*

$$\tilde{\Lambda}_\Pi(\rho) := \sum_i (I \otimes \langle i| \otimes I_n)\tilde{V}(|0^{m+1}\rangle \langle 0^{m+1}| \otimes \rho)\tilde{V}^\dagger(I \otimes |i\rangle \otimes I_n)$$

*approximates the quantum channel  $\Lambda_0(\rho)$  in diamond norm:*

$$\|\tilde{\Lambda}_\Pi - \Lambda_0\|_\diamond \leq 4\sqrt{2}\epsilon.$$

## 4 Quantum algorithm for additive error estimation of normalized Persistent Betti numbers

This section presents a quantum algorithm for estimating the persistent Betti numbers of arbitrary dimensions for a simplicial pair  $K \hookrightarrow L$ . We assume we have access to quantum membership oracles  $O_q^K$  and  $O_{q+1}^L$  that return whether a simplex is contained in  $K$  and  $L$  for any  $\sigma \in \{0,1\}^n$  and  $a \in \{0,1\}$ , respectively, as

$$\begin{aligned} O_q^K|\sigma\rangle|a\rangle &= |\sigma\rangle|a \oplus f_q^K(\sigma)\rangle, \\ O_{q+1}^L|\sigma\rangle|a\rangle &= |\sigma\rangle|a \oplus f_{q+1}^L(\sigma)\rangle, \end{aligned}$$

where  $f_q^K(\sigma) = 1$  iff  $\sigma \in \bar{S}_q^K$  and  $f_{q+1}^L(\sigma) = 1$  iff  $\sigma \in \bar{S}_{q+1}^L$ . For a simplicial complex with  $n$  vertices, the number of all possible simplices is exponential in  $n$ . However, we can efficiently implement such a membership function for some constructions of the filtration that is commonly used in TDA, as we see in Appendix B. For example, the most straightforward way to implement the membership function for the Vietoris-Rips complex is to use the QRAM access to the  $n \times n$  size adjacency matrix that represents the connectivity of the points. Moreover, we can efficiently implement the membership function for the Vietoris-Rips complex by using a quantum circuit that returns the adjacency whenever we want because the adjacency matrix is of  $n \times n$  size.

Our main result is as follows:

**Theorem 7** Let  $K \hookrightarrow L$  be a simplicial pair that satisfies the following promises:

(P1)  $K$  is  $q$ -simplex dense:  $d_q^K = n_q^K / \binom{n}{q+1} \in \Omega(1/\text{poly}(n))$

(P2)  $\Delta_{q,\text{up}}^L(I_K^L, I_K^L)$  has inverse-polynomial gap:  $\gamma_{\min}^q \in \Omega(1/\text{poly}(n))$

(P3) the  $q$ -th persistent Laplacian  $\Delta_q^{K,L}$  has inverse-polynomial gap:  $\lambda_{\min}^q \in \Omega(1/\text{poly}(n))$ .

Then, given access to membership oracles  $O_q^K$  and  $O_{q+1}^L$ , there is a quantum algorithm that outputs 1 with probability  $\tilde{p}_1$  s.t. for any  $\epsilon > 0$

$$\left| \frac{\beta_q^{K,L}}{n_q^K} - \tilde{p}_1 \right| \leq \epsilon$$

which uses  $\mathcal{O}(n) + \mathcal{O}(\log n)$  qubits, and  $\tilde{\mathcal{O}}\left(\text{poly}(n) \left(\log \frac{1}{\epsilon}\right)^2\right)$  number of  $O_q^K, O_{q+1}^L$  and other quantum gates.

As a consequence, we obtain the following corollary to estimate the persistent Betti numbers using the Chernoff-Hoeffding inequality.

**Corollary 2** For any simplicial pair  $K \hookrightarrow L$  that satisfies the promises (P1)  $\sim$  (P3) of Theorem 7, given access to membership oracles  $O_q^K$  and  $O_{q+1}^L$ , we can efficiently estimate  $\frac{\beta_q^{K,L}}{n_q^K}$  with additive error  $\epsilon$  with confidence  $\eta$  by taking  $\mathcal{O}\left(\frac{1}{\epsilon^2} \ln\left(\frac{2}{\eta}\right)\right)$  samples from the quantum algorithm in Theorem 7.

Without the promises (P1) $\sim$ (P3), the complexity of our quantum algorithm can be stated as follows:

**Theorem 8** Let  $K \hookrightarrow L$  be a simplicial pair. For any  $\epsilon_{\text{sign}}, \epsilon_{\text{inv}}, \epsilon_{\text{rect}} > 0$ , given access to membership oracles  $O_q^K$  and  $O_{q+1}^L$ , there is a quantum algorithm that outputs 1 with probability  $\tilde{p}_1$  s.t.

$$\left| \frac{\beta_q^{K,L}}{n_q^K} - \tilde{p}_1 \right| \leq 8\sqrt{2}\epsilon_{\Pi} + \epsilon_{\text{sign}},$$

which uses  $\mathcal{O}(n) + \mathcal{O}(\log n)$  qubits and the following number of oracles and gates:

$$\begin{aligned} O_q^K &: \mathcal{O}\left(\sqrt{\frac{1}{d_q^K}} \log\left(\frac{1}{\epsilon_{\text{sign}}}\right)\right) + \mathcal{O}\left(\frac{q^4 n^6}{(\gamma_{\min}^q)^2 \lambda_{\min}^q} \log\left(\frac{1}{\epsilon_{\text{rect}}}\right) \log\left(\frac{1}{\gamma_{\min}^q \epsilon_{\text{inv}}}\right)\right), \\ O_{q+1}^L &: \mathcal{O}\left(\frac{q^4 n^6}{(\gamma_{\min}^q)^2 \lambda_{\min}^q} \log\left(\frac{1}{\epsilon_{\text{rect}}}\right) \log\left(\frac{1}{\gamma_{\min}^q \epsilon_{\text{inv}}}\right)\right), \\ \text{gates} &: \mathcal{O}\left(qn^2 \sqrt{\frac{1}{d_q^K}} \log\left(\frac{1}{\epsilon_{\text{sign}}}\right)\right) + \tilde{\mathcal{O}}\left(\frac{q^4 n^8}{(\gamma_{\min}^q)^2 \lambda_{\min}^q} \log\left(\frac{1}{\epsilon_{\text{rect}}}\right) \log\left(\frac{1}{\gamma_{\min}^q \epsilon_{\text{inv}}}\right)\right). \end{aligned}$$

Here,  $\epsilon_{\Pi} = \epsilon_{\text{rect}} + \mathcal{O}\left(q^4 n^4 \sqrt{\epsilon_{\text{inv}}} \log(1/\epsilon_{\text{rect}}) / (\lambda_{\min}^q \sqrt{\gamma_{\min}})\right)$ .

In the following, we give the proof of Theorem 7 and Theorem 8. The proof consists of three parts. First, We propose the quantum algorithm in Section 4.1, Section 4.2, and Section 4.3. (The construction of some unitaries are given in Section 5 and Section 6.) Second, we give the error analysis in Section 4.4. Finally, we give the complexity analysis in Section 4.5.

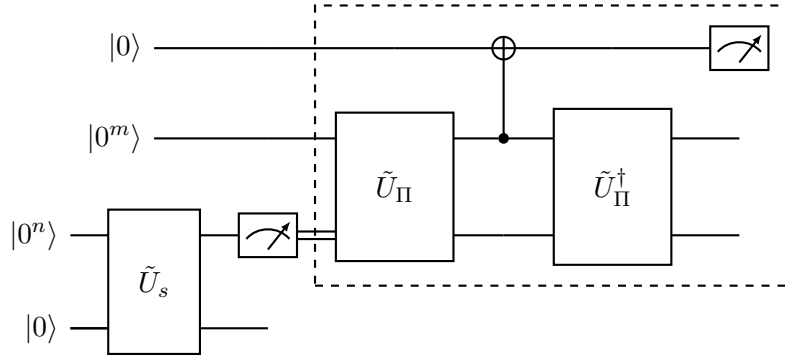


Figure 2: Description of the quantum circuit for estimating the normalized persistent Betti numbers

#### 4.1 Description of the quantum algorithm

The quantum circuit used in our algorithm is described in Fig. 2. In Fig. 2,  $\tilde{U}_s$  is a unitary for state preparation and  $\tilde{U}_\Pi$  is a unitary that is a block-encoding of the projection onto the low-energy subspace of the persistent Laplacian. The  $m$ -qubits ( $m \in \mathcal{O}(\log n)$ ) are ancilla qubits for the block-encoding. The measurement procedure surrounded by the dashed box is the “block-measurement” procedure [35] which we have reviewed in Section 3.3.

#### 4.2 State Preparation $\tilde{U}_s$

The state preparation unitary  $\tilde{U}_s$  is the approximation of the following unitary  $U_s$ :

$$U_s |0^{n+1}\rangle = |\phi_q^K\rangle \otimes |1\rangle,$$

where  $|\phi_q^K\rangle := \frac{1}{\sqrt{n_q^K}} \sum_{i \in [n_q^K]} |\sigma_q^K(i)\rangle$ , which is the uniform superposition over the  $q$ -simplices of  $K$ . In Section 5, we construct a unitary  $\tilde{U}_s$  s.t.

$$\left\| |\phi_q^K\rangle |1\rangle - \tilde{U}_s |0^{n+1}\rangle \right\| \leq \epsilon_{sign} \quad (5)$$

using  $\mathcal{O}\left(qn^2 \sqrt{1/d_q^K} \log(1/\epsilon_{sign})\right)$ -number of gates and  $\mathcal{O}\left(\sqrt{1/d_q^K} \log(1/\epsilon_{sign})\right)$ -number of use of  $O_q^K$ . Here,  $d_q^K := n_q^K / \binom{n}{q+1}$ . We denote  $\tilde{U}_s |0^{n+1}\rangle := |\tilde{\psi}\rangle$ .

We show that by measuring the state prepared by  $\tilde{U}_s$  in the computational basis as in Fig. 2, we can approximately sample from the uniform mixture over the eigenvectors of  $\Delta_q^{K,L}$ .<sup>3</sup> By measuring the first  $n$ -qubit of  $U_s |0^{n+1}\rangle$ , the state will collapse to some  $|\sigma_q^K(i)\rangle$  approximately uniform randomly. Repeating the quantum circuit of Fig. 2 and taking samples multiple times from the collapsed state is equivalent to sampling from a state described by a density matrix  $\tilde{\rho}_q^K$  which is close to  $\rho_q^K := \frac{1}{n_q^K} \sum_{i \in [n_q^K]} |\sigma_q^K(i)\rangle \langle \sigma_q^K(i)|$ . This  $\rho_q^K$  is also the uniform mixture of the eigenvectors of  $\Delta_q^{K,L}$ , i.e.,  $\rho_q^K = \frac{1}{n_q^K} \sum_{i \in [n_q^K]} |\psi_i\rangle \langle \psi_i|$ ,

<sup>3</sup>In the first version of the preprint of this paper [21], we copied the first register to another to obtain  $\frac{1}{\sqrt{n_q^K}} \sum_{i \in [n_q^K]} |\sigma_q^K(i)\rangle \otimes |1\rangle \otimes |\sigma_q^K(i)\rangle$ , and then traced out the latter registers to approximately obtain  $\rho_q^K$ . It is pointed out by an anonymous reviewer that this copy and discarding operation can be omitted by measuring in the computational basis and repeating the algorithm a number of times. The copy and discarding operation is required if we use coherently estimate the amplitude using the amplitude estimation [4] for example.

where we denote the eigenvectors of  $\Delta_q^{K,L}$  as  $|\psi_1\rangle, \dots, |\psi_{n_q^K}\rangle$ . This is because  $\forall i, |\sigma_q(i)\rangle$  can be written as  $|\sigma_q(i)\rangle = \sum_j U_{i,j} |\psi_j\rangle$  for some unknown unitary  $U$  and

$$\rho_q^K = \frac{1}{n_q^K} \sum_i |\sigma_q(i)\rangle \langle \sigma_q(i)| = \frac{1}{n_q^K} \sum_{j,j'} \left( \sum_i U_{i,j} U_{i,j'}^* \right) |\psi_j\rangle \langle \psi_{j'}| = \frac{1}{n_q^K} \sum_j |\psi_j\rangle \langle \psi_j|.$$

We give an upper bound on  $\|\rho_q^K - \tilde{\rho}_q^K\|_1$ , which we use in the error analysis later. Using purification,  $\tilde{\rho}_q^K$  can be written as

$$\tilde{\rho}_q^K = \text{Tr}_s \left[ U_{\text{copy}} \left( |\tilde{\psi}\rangle \langle \tilde{\psi}| \otimes |0^n\rangle \langle 0^n| \right) U_{\text{copy}}^\dagger \right],$$

where  $U_{\text{copy}}$  is a unitary that copies the first  $n$ -qubit register to the last  $n$ -qubit register, and the subscript  $s$  means the last  $n$ -qubit register.  $\rho_q^K$  can be similarly written as

$$\rho_q^K = \text{Tr}_s \left[ U_{\text{copy}} \left( |\psi\rangle \langle \psi| \otimes |0^n\rangle \langle 0^n| \right) U_{\text{copy}}^\dagger \right],$$

where  $|\psi\rangle := |\phi_q^K\rangle \otimes |1\rangle$ . Then

$$\begin{aligned} \left\| \rho_q^K - \tilde{\rho}_q^K \right\|_1 &= \left\| \text{Tr}_s \left[ U_{\text{copy}} \left( |\psi\rangle \langle \psi| \otimes |0^n\rangle \langle 0^n| - |\tilde{\psi}\rangle \langle \tilde{\psi}| \otimes |0^n\rangle \langle 0^n| \right) U_{\text{copy}}^\dagger \right] \right\|_1 \\ &\leq \left\| U_{\text{copy}} \left( |\psi\rangle \langle \psi| \otimes |0^n\rangle \langle 0^n| - |\tilde{\psi}\rangle \langle \tilde{\psi}| \otimes |0^n\rangle \langle 0^n| \right) U_{\text{copy}}^\dagger \right\|_1 \\ &= \left\| |\psi\rangle \langle \psi| \otimes |0^n\rangle \langle 0^n| - |\tilde{\psi}\rangle \langle \tilde{\psi}| \otimes |0^n\rangle \langle 0^n| \right\|_1 \\ &= \left\| |\psi\rangle \langle \psi| - |\tilde{\psi}\rangle \langle \tilde{\psi}| \right\|_1 = \sqrt{1 - |\langle \psi | \tilde{\psi} \rangle|^2} \leq \epsilon_{\text{sign}}. \end{aligned} \quad (6)$$

### 4.3 Block-measurement with respect to $\tilde{U}_\Pi$

In Section 6, we provide the implementation of  $\tilde{U}_\Pi$  which is a  $(1, m, \epsilon_\Pi)$ -block-encoding of  $\Pi := \sum_{j:\lambda_j=0} |\psi_j\rangle \langle \psi_j|$  assuming that the smallest non-zero eigenvalue of  $\Delta_q^{K,L}$  is  $\lambda_{\min}^q$ . We also assume that the smallest non-zero eigenvalue of the submatrix  $\Delta_{q,\text{up}}^L(I_K^L, I_K^L)$  is  $\gamma_{\min}^q$ . The number of oracles and gates that are used in the construction of  $\tilde{U}_\Pi$  is as follows:

$$\begin{aligned} O_q^K &: \mathcal{O} \left( \frac{q^4 n^6}{(\gamma_q)^2 \lambda_q} \log \left( \frac{1}{\epsilon_{\text{rect}}} \right) \log \left( \frac{1}{\gamma_q \epsilon_{\text{inv}}} \right) \right), \\ O_{q+1}^L &: \mathcal{O} \left( \frac{q^4 n^6}{(\gamma_q)^2 \lambda_q} \log \left( \frac{1}{\epsilon_{\text{rect}}} \right) \log \left( \frac{1}{\gamma_q \epsilon_{\text{inv}}} \right) \right), \\ \text{gates} &: \tilde{\mathcal{O}} \left( \frac{q^4 n^8}{(\gamma_q)^2 \lambda_q} \log \left( \frac{1}{\epsilon_{\text{rect}}} \right) \log \left( \frac{1}{\gamma_q \epsilon_{\text{inv}}} \right) \right). \end{aligned}$$

Here,  $\lambda_q$  and  $\gamma_q$  are real numbers s.t.  $0 < \lambda_q < \lambda_{\min}^q$  and  $0 < \gamma_q < \gamma_{\min}^q$ , and  $\epsilon_{\text{rect}}, \epsilon_{\text{inv}} > 0$  are real numbers s.t.  $\epsilon_\Pi = \epsilon_{\text{rect}} + \mathcal{O} \left( q^4 n^4 \sqrt{\epsilon_{\text{inv}}} \log(1/\epsilon_{\text{rect}}) / (\lambda_q \sqrt{\gamma_q}) \right)$ .

$U_\Pi$  satisfies

$$\left\| \Pi - \left( (|0^m\rangle \otimes I_{n+1}) U_\Pi (|0^m\rangle \otimes I_{n+1}) \right)^2 \right\| \leq 2\epsilon_\Pi - \epsilon_\Pi^2 \leq 2\epsilon_\Pi.$$

We perform block-measurement with this  $\tilde{U}_\Pi$ . Let  $\Lambda_\Pi$  be a quantum channel that maps as

$$|0\rangle \otimes |\psi\rangle \rightarrow |1\rangle \otimes \Pi|\psi\rangle + |0\rangle \otimes (I - \Pi)|\psi\rangle.$$

This is a quantum channel realized by ideal block-encoding  $U_\Pi$  of  $\Pi$  with no error, which we denote  $\Lambda_\Pi$ . From Theorem 6, the quantum channel realized by the block-measurement with  $\tilde{U}_\Pi$  satisfies

$$\|\tilde{\Lambda}_\Pi - \Lambda_\Pi\|_\diamond \leq 8\sqrt{2}\epsilon_\Pi. \quad (7)$$

We use this inequality in error analysis in the next subsection.

#### 4.4 Error analysis

It can be seen that for the ideal state preparation and the ideal block-measurement, the probability of outputting 1 is

$$\begin{aligned} p_1 &:= \text{Tr}_{AB} \left[ (|1\rangle\langle 1| \otimes I_{m+n}) V \left( |0^{m+1}\rangle \langle 0^{m+1}| \otimes \rho_q^K \right) V^\dagger \right] \\ &= \text{Tr}_A \left[ (|1\rangle\langle 1| \otimes I_n) \Lambda_0 \left( |0\rangle \langle 0| \otimes \rho_q^K \right) \right] \\ &= \frac{|\{i : \lambda_i = 0\}|}{n_q^K} = \frac{\text{nullity}(\Delta_q^{K,L})}{n_q^K}, \end{aligned}$$

where we have denoted the  $m$ -qubit register as  $A$  and other registers as  $B$ , and  $V$  is the block-measurement circuit of Fig. 1 with  $U_\Pi$ . The actual probability that the measurement outcome is 1 in the final measurement of Fig. 2 is

$$\begin{aligned} \tilde{p}_1 &= \text{Tr}_{AB} \left[ (|1\rangle\langle 1| \otimes I_{m+n}) \tilde{V} \left( |0^{m+1}\rangle \langle 0^{m+1}| \otimes \tilde{\rho}_q^K \right) \tilde{V}^\dagger \right] \\ &= \text{Tr}_A \left[ (|1\rangle\langle 1| \otimes I_n) \tilde{\Lambda}_\Pi \left( |0\rangle \langle 0| \otimes \tilde{\rho}_q^K \right) \right], \end{aligned}$$

where  $\tilde{V}$  is the block-measurement circuit of Fig. 1 with  $\tilde{U}_\Pi$ . Let  $\mathcal{M} = |1\rangle\langle 1| \otimes I_n$ . The difference between the ideal probability and the actual probability can be calculated as follows:

$$\begin{aligned} |p_1 - \tilde{p}_1| &= \left| \text{Tr} \left[ \mathcal{M} \Lambda_0 \left( |0\rangle \langle 0| \otimes \rho_q^K \right) \right] - \text{Tr} \left[ \mathcal{M} \tilde{\Lambda}_\Pi \left( |0\rangle \langle 0| \otimes \tilde{\rho}_q^K \right) \right] \right| \\ &\leq \left| \text{Tr} \left[ \mathcal{M} \left( \Lambda_0 \left( |0\rangle \langle 0| \otimes \rho_q^K \right) - \tilde{\Lambda}_\Pi \left( |0\rangle \langle 0| \otimes \rho_q^K \right) \right) \right] \right| + \left| \text{Tr} \left[ \mathcal{M} \tilde{\Lambda}_\Pi \left( |0\rangle \langle 0| \otimes \left( \rho_q^K - \tilde{\rho}_q^K \right) \right) \right] \right| \\ &\leq \left\| \left( \Lambda_0 - \tilde{\Lambda}_\Pi \right) \left( |0\rangle \langle 0| \otimes \rho_q^K \right) \right\|_1 + \left\| \rho_q^K - \tilde{\rho}_q^K \right\|_1 \\ &\leq \|\Lambda_0 - \tilde{\Lambda}_\Pi\|_\diamond + \left\| \rho_q^K - \tilde{\rho}_q^K \right\|_1 \\ &\leq 8\sqrt{2}\epsilon_\Pi + \epsilon_{\text{sign}}, \end{aligned}$$

where we have used (6) and (7).

#### 4.5 Analysis of the number of gates and the use of the oracles.

The total number of gates and use of oracles can be summarized as follows:

$$\begin{aligned} O_q^K &: \mathcal{O} \left( \sqrt{\frac{1}{d_q^K}} \log \left( \frac{1}{\epsilon_{\text{sign}}} \right) \right) + \mathcal{O} \left( \frac{q^4 n^6}{(\gamma_q)^2 \lambda_q} \log \left( \frac{1}{\epsilon_{\text{rect}}} \right) \log \left( \frac{1}{\gamma_q \epsilon_{\text{inv}}} \right) \right), \\ O_{q+1}^L &: \mathcal{O} \left( \frac{q^4 n^6}{(\gamma_q)^2 \lambda_q} \log \left( \frac{1}{\epsilon_{\text{rect}}} \right) \log \left( \frac{1}{\gamma_q \epsilon_{\text{inv}}} \right) \right), \\ \text{gates} &: \mathcal{O} \left( q n^2 \sqrt{\frac{1}{d_q^K}} \log \left( \frac{1}{\epsilon_{\text{sign}}} \right) \right) + \tilde{\mathcal{O}} \left( \frac{q^4 n^8}{(\gamma_q)^2 \lambda_q} \log \left( \frac{1}{\epsilon_{\text{rect}}} \right) \log \left( \frac{1}{\gamma_q \epsilon_{\text{inv}}} \right) \right). \end{aligned}$$

If we want to make  $|p_1 - \tilde{p}_1| \leq \epsilon$ , we can take  $\epsilon_\Pi \in \mathcal{O}(\epsilon)$ ,  $\epsilon_{sign} \in \mathcal{O}(\epsilon)$ ,  $\epsilon_{rect} \in \mathcal{O}(\epsilon)$  and

$$\epsilon_{inv} \in \mathcal{O}\left(\frac{(\lambda_q)^2 \gamma_q \epsilon^2}{q^8 n^8 \log(1/\epsilon)}\right).$$

Therefore, if  $d_q^K = n_q^K / \binom{n}{q+1} \in \Omega(1/\text{poly}(n))$ ,  $\gamma_{\min}^q \in \Omega(1/\text{poly}(n))$  and  $\lambda_{\min}^q \in \Omega(1/\text{poly}(n))$ , the quantum circuit uses  $\tilde{\mathcal{O}}\left(\text{poly}(n) \left(\log \frac{1}{\epsilon}\right)^2\right)$  number of  $O_q^K, O_{q+1}^L$  and other quantum gates.

## 5 Implementation of the state preparation unitary

In this section, we provide the implementation of the state preparation unitary  $\tilde{U}_s$  of eq. (5). In order to implement  $\tilde{U}_s$ , we use a unitary  $P_q$  s.t.

$$P_q |0^n\rangle = \frac{1}{\sqrt{\binom{n}{q+1}}} \sum_{x \in W_q} |x\rangle, \quad (8)$$

where  $W_q$  is the set of Hamming weight  $q+1$   $n$ -bit strings. In [19], the combinatorial number system [36] is used to implement such a unitary. The uniform superposition of the Hamming weight  $k$  states is also known as the Dicke state [12]. Instead of following the construction of [19] using the combinatorial number system, we can use the construction of [3] for the Dicke state. Then, we can implement  $P_q$  with  $\mathcal{O}(qn)$  gates and  $\mathcal{O}(n)$  depth without using any ancilla qubits. (The gate complexity of [19] is similar, but it additionally requires  $\tilde{\mathcal{O}}(qn^2)$  time of computation for preparing a look-up table.)

Then by adding a qubit and applying the membership oracle  $O_q^K$  to the quantum state of (8), we get

$$\frac{1}{\sqrt{\binom{n}{q+1}}} \left( \sum_{x \in S_q^K} |x\rangle|1\rangle + \sum_{x \notin S_q^K} |x\rangle|0\rangle \right) = \sqrt{d_q^K} |\phi_q^K\rangle |1\rangle + \frac{1}{\sqrt{\binom{n}{q+1}}} \sum_{x \notin S_q^K} |x\rangle|0\rangle.$$

We call  $d_q^K$  the  $q$ -simplex density of  $K$ . Let  $\tilde{\Pi} = I_n \otimes |1\rangle\langle 1|$ ,  $\Pi = |0^{n+1}\rangle\langle 0^{n+1}|$  and  $U = O_q^K P_q$ . Then  $\tilde{\Pi} U \Pi = \sqrt{d_q^K} (|\phi_q^K\rangle |1\rangle)\langle 0^{n+1}|$ . Now, we can use Theorem 4 to implement a unitary  $\tilde{U}_s$  s.t.

$$\left\| |\phi_q^K\rangle |1\rangle - \tilde{U}_s |0^{n+1}\rangle \right\| \leq \epsilon_{sign}$$

with  $\mathcal{O}\left(\sqrt{1/d_q^K} \log(1/\epsilon_{sign})\right)$  use of  $U, U^\dagger, C_{\tilde{\Pi}}\text{NOT}, C_{\Pi}\text{NOT}$ , and single qubit gates. Each of the  $C_{\Pi}\text{NOT}$  gates, which are the generalized Toffoli gates, can be decomposed into  $\mathcal{O}(n)$  number of elementary gates with an ancilla qubit [23]. Therefore,  $\tilde{U}_s$  can be implemented using  $\mathcal{O}\left(qn^2 \sqrt{1/d_q^K} \log(1/\epsilon_{sign})\right)$ -number of gates and  $\mathcal{O}\left(\sqrt{1/d_q^K} \log(1/\epsilon_{sign})\right)$ -number of use of  $O_q^K$ .

## 6 Implementation of the block-encoding of the projector $\tilde{U}_\Pi$

In this section, we give a construction of  $\tilde{U}_\Pi$  which is a  $(1, m, \epsilon_\Pi)$ -block-encoding of  $\Pi$ . We first show the implementation of the block-encoding of the persistent Laplacian

$$\Delta_q^{K,L} = \underbrace{\partial_{q+1}^{L,K} \circ (\partial_{q+1}^{L,K})^*}_{\Delta_{q,\text{up}}^{K,L}} + \underbrace{(\partial_q^K)^* \circ \partial_q^K}_{\Delta_{q,\text{down}}^K}$$



in Section 6.1. Then we show the construction of a  $(1, m, \epsilon_\Pi)$ -block-encoding of  $\Pi$  in Section 6.2 using the block-encoding of  $\Delta_q^{K,L}$ .

The implementation of  $\tilde{U}_\Pi$  includes the inversion of matrices. More precisely, our algorithm includes the implementation of the Schur complement of a matrix that contains the implementation of the Moore-Penrose pseudo-inverse. The implementation of the Schur complement may be of independent interest. The fact that our algorithm includes the matrix inversion implies that it shares the characteristics of both the HHL algorithm [20] and the LGZ algorithm.

## 6.1 Implementation of the block-encoding of $\Delta_q^K$

We provide the procedure of implementing the block-encoding of  $\Delta_q^K$  in the following.

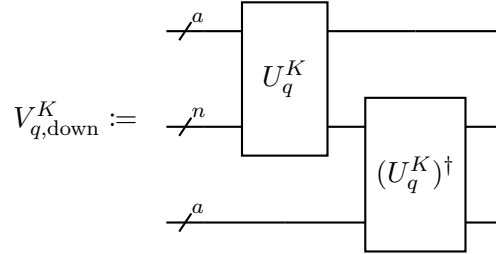
### 6.1.1 Block-encoding of $\Delta_{q,\text{down}}^K = (\partial_q^K)^* \circ \partial_q^K$

Let  $\tilde{n} := 2^{\lceil \log n \rceil}$ , and  $\tilde{q} := 2^{\lceil \log(q+1) \rceil}$ . In Appendix C, we construct a unitary  $U_q^K$  which is an  $(\tilde{n}\tilde{q}, a, 0)$ -block-encoding of  $\partial_q^K$ .  $U_q^K$  is implemented using  $\tilde{\mathcal{O}}(n^2)$ -number of gates and  $\mathcal{O}(1)$ -use of  $O_q^K$ . Here  $a = \mathcal{O}(\log(n))$ .

It follows that  $(U_q^K)^\dagger$  is an  $(\tilde{n}\tilde{q}, a, 0)$ -block-encoding of  $(\partial_q^K)^*$  because the matrix representation of  $(\partial_q^K)^*$  is the transpose of the matrix representation of  $\partial_q^K$  whose elements are real. We can implement the block-encoding of  $\Delta_{q,\text{down}}^K$  using the following lemma:

**Lemma 1 (Product of block-encoded matrices (Lemma 53 of [18]))** *If  $U$  is an  $(\alpha, a, \delta)$ -block-encoding of an  $s$ -qubit operator  $A$ , and  $V$  is a  $(\beta, b, \epsilon)$ -block-encoding of a  $s$ -qubit operator  $B$  then  $(I_b \otimes U)(I_a \otimes V)$  is an  $(\alpha\beta, a + b, \alpha\epsilon + \beta\delta)$ -block-encoding of  $AB$ .*

Using this lemma, we can implement a unitary



which is an  $(\tilde{n}^2\tilde{q}^2, 2a, 0)$ -block-encoding of  $\Delta_{q,\text{down}}^K$ . From the construction above,  $V_{q,\text{down}}^K$  can be implemented with  $\tilde{\mathcal{O}}(n^2)$  gates and  $\mathcal{O}(1)$  use of  $O_q^K$ .

### 6.1.2 Implementation of the submatrices of $\Delta_{q,\text{up}}^L$ and its Schur complement

Let us define the submatrices

$$\begin{aligned}\Delta_1 &:= \Delta_{q,\text{up}}^L([n_q^K], [n_q^K]) \\ \Delta_2 &:= \Delta_{q,\text{up}}^L([n_q^K], I_K^L) \\ \Delta_3 &:= \Delta_{q,\text{up}}^L(I_K^L, [n_q^K]) \\ \Delta_4 &:= \Delta_{q,\text{up}}^L(I_K^L, I_K^L)\end{aligned}$$

for convenience. By Theorem 2,

$$\Delta_{q,\text{up}}^{K,L} = \Delta_{q,\text{up}}^L / \Delta_{q,\text{up}}^L(I_K^L, I_K^L) = \Delta_1 - \Delta_2 \Delta_4^+ \Delta_3.$$

Let us also define  $q' := 2^{\lceil \log(q+2) \rceil}$ . We can similarly implement a unitary  $V_{q,\text{up}}^L$  which is a  $(\tilde{n}^2 q'^2, 2b, 0)$ -block-encoding of  $\Delta_{q,\text{up}}^L : \mathcal{H}_q^L \rightarrow \mathcal{H}_q^L$  where  $b = \mathcal{O}(\log(n))$  with using  $\tilde{\mathcal{O}}(n^2)$  number of gates and  $\mathcal{O}(1)$  number of  $O_{q+1}^L$ .

In order to implement the Schur complement  $\Delta_{q,\text{up}}^L / \Delta_{q,\text{up}}^L(I_K^L, I_K^L)$ , we need to access the submatrices of  $\Delta_{q,\text{up}}^L$ . The membership oracle  $O_q^K : |x\rangle |0\rangle \rightarrow |x\rangle |f_q^K(x)\rangle$  can be used to access those submatrices. Let  $\tilde{O}_q^K : |x\rangle |0\rangle \rightarrow |x\rangle |f_q^K(x) \oplus 1\rangle$ . Then, let us introduce

$$\begin{aligned} V_1 &:= (I \otimes \tilde{O}_q^K \otimes I_{2b})(I_2 \otimes V_q^L)(\tilde{O}_q^K \otimes I \otimes I_{2b}), \\ V_2 &:= (I \otimes \tilde{O}_q^K \otimes I_{2b})(I_2 \otimes V_q^L)(O_q^K \otimes I \otimes I_{2b}), \\ V_3 &:= (I \otimes O_q^K \otimes I_{2b})(I_2 \otimes V_q^L)(\tilde{O}_q^K \otimes I \otimes I_{2b}), \\ V_4 &:= (I \otimes O_q^K \otimes I_{2b})(I_2 \otimes V_q^L)(O_q^K \otimes I \otimes I_{2b}). \end{aligned}$$

By applying  $\tilde{O}_q^K$  and postselecting the ancilla register to  $|0\rangle$ , we can restrict the space to that is spanned by  $\{|\sigma_q^K(i)\rangle\}_{i \in [n_q^K]}$ . Similarly, by applying  $O_q^K$  with postselecting the ancilla register to  $|0\rangle$ , we can restrict the space to that is not spanned by  $\{|\sigma_q^K(i)\rangle\}_{i \in [n_q^K]}$ . Therefore,  $V_1, V_2, V_3$  and  $V_4$  are  $(\tilde{n}^2 q'^2, 2b + 2, 0)$ -block-encodings of  $\Delta_1, \Delta_2, \Delta_3$  and  $\Delta_4$ , respectively.  $V_1 \sim V_4$  can be implemented using  $\tilde{\mathcal{O}}(n^2)$ -gates and  $\mathcal{O}(1)$ -use of  $O_q^K$  and  $O_{q+1}^L$ .

Let  $\gamma_{\min}^q$  be the smallest non-zero eigenvalue of  $\Delta_4$  and let  $\kappa = \frac{\tilde{n}^2 q'^2}{\gamma_q}$  for  $\gamma_q$  s.t.  $0 < \gamma_q < \gamma_{\min}^q$ . Then, using Theorem 5, we can implement a  $(\frac{2}{\gamma_q}, 2b + 3, \epsilon_{\text{inv}})$ -block-encoding of  $\Delta_4^+$  with  $\mathcal{O}(\frac{q^2 n^2}{\gamma_q} \log(\frac{1}{\gamma_q \epsilon_{\text{inv}}}))$  use of  $V_4, V_4^\dagger$  and  $\tilde{\mathcal{O}}(\frac{q^2 n^2}{\gamma_q} \log(\frac{1}{\gamma_q \epsilon_{\text{inv}}}))$  other elementary gates.

We can implement a unitary  $V_+$  which is a  $(\frac{2\tilde{n}^4 q'^4}{\gamma_q}, 6b + 7, \tilde{n}^4 q'^4 \epsilon_{\text{inv}})$ -block-encoding of  $\Delta_2 \Delta_4^+ \Delta_3$  using Lemma 1.

**Remark 2** As in Theorem 2,  $\Delta_q^{K,L} = \Delta_{q,\text{up}}^L / \Delta_{q,\text{up}}^L(I_K^L, I_K^L)$ , where  $I_K^L := [n_q^L] \setminus [n_q^K]$ . However, we do not need to know the values of  $n_q^K$  and  $n_q^L$  to run our algorithm because we can implement the block-encoding of submatrices by  $V_1, V_2, V_3$  and  $V_4$  using the membership oracles. The membership oracles have been used to effectively implement the restrictions to the corresponding subspaces of  $I_k^L$  coherently, bypassing the need to classically learn the values of  $I_k^L$ .

### 6.1.3 Linear combination of block-encoding unitaries

Finally, we linearly combine the block-encoding (BE) of  $\Delta_1, \Delta_2 \Delta_4^+ \Delta_3$  and  $\Delta_{q,\text{down}}^K$ . We implement such a linear combination in a way similar to Lemma 52 of [18]. What we have prepared so far are the following unitaries:

$$\begin{aligned} U_0 := V_1 &\Leftrightarrow (\alpha_0 := \tilde{n}^2 q'^2, 2b + 2, 0) - \text{BE of } \Delta_1 \\ U_1 := V_+ &\Leftrightarrow (\alpha_1 := \frac{2\tilde{n}^4 q'^4}{\gamma_q}, 6b + 7, \tilde{n}^4 q'^4 \epsilon_{\text{inv}}) - \text{BE of } \Delta_2 \Delta_4^+ \Delta_3 \\ U_2 := V_q^K &\Leftrightarrow (\alpha_2 := \tilde{n}^2 q'^2, 2a, 0) - \text{BE of } \Delta_{q,\text{down}}^K, \end{aligned}$$

where we have introduced  $U_0, U_1, U_2$  and  $\alpha_0, \alpha_1, \alpha_2$  for the convenience of notation. What we implement is a block-encoding of

$$\Delta_q^{K,L} = \Delta_1 - \Delta_2 \Delta_4^+ \Delta_3 + \Delta_{q,\text{down}}^K.$$

Let  $\beta = \alpha_0 + \alpha_1 + \alpha_2$  and  $P_R, P_L$  be state preparation unitaries such that

$$\begin{aligned} P_R |0\rangle &\rightarrow \frac{1}{\sqrt{\beta}}(\sqrt{\alpha_0} |0\rangle + \sqrt{\alpha_1} |1\rangle + \sqrt{\alpha_2} |2\rangle), \\ P_L |0\rangle &\rightarrow \frac{1}{\sqrt{\beta}}(\sqrt{\alpha_0} |0\rangle - \sqrt{\alpha_1} |1\rangle + \sqrt{\alpha_2} |2\rangle). \end{aligned}$$

Suppose  $W$  is a unitary such that for all  $j \in 0, 1, 2$  and any state  $|\psi\rangle$ ,

$$W : |j\rangle |\psi\rangle \rightarrow |j\rangle U_j |\psi\rangle.$$

Let  $\tilde{W} = (P_L^\dagger \otimes I_{6b+7} \otimes I_n)W(P_R \otimes I_{6b+7} \otimes I_n)$ . Then, this satisfies

$$\left\| \left( \langle 0^{6b+9} | \otimes I_n \right) \tilde{W} \left( |0^{6b+9}\rangle \otimes I_n \right) - \frac{\Delta_q^{K,L}}{\beta} \right\| \leq \frac{\tilde{n}^4 q'^4 \epsilon_{inv}}{\beta}$$

and  $\tilde{W}$  is a  $(\beta, 6b+9, \tilde{n}^4 q'^4 \epsilon_{inv})$ -block-encoding of  $\Delta_q^{K,L}$ , and  $\beta \in \mathcal{O}(\frac{q^{4n^4}}{\gamma_q})$ . Let us denote

$$\left( \langle 0^{6b+9} | \otimes I_n \right) \tilde{W} \left( |0^{6b+9}\rangle \otimes I_n \right) = \tilde{\Delta}_q^{K,L} / \beta$$

for  $\tilde{\Delta}_q^{K,L}$  s.t.  $\left\| \Delta_q^{K,L} - \tilde{\Delta}_q^{K,L} \right\| \leq \tilde{n}^4 q'^4 \epsilon_{inv}$ .

## 6.2 Implementing the block-encoding of the projector

Using the block-encoding of  $\Delta_q^{K,L}$ , we implement the block-encoding of the projector onto the zero energy space of  $\Delta_q^{K,L}$ :  $\Pi = \sum_{j:\lambda_j=0} |\psi_j\rangle \langle \psi_j|$ . Let  $\lambda_{\min}^q$  be the smallest non-zero eigenvalue of  $\Delta_q^{K,L}$ . In [18], the following polynomial approximation of the rectangle function is shown:

**Lemma 2 (Lemma 29, [18])** *Let  $\delta, \epsilon \in (0, 1/2)$  and  $t \in [-1, 1]$ . There exist an even polynomial  $P \in \mathbb{R}[x]$  of degree  $\mathcal{O}(\log(1/\epsilon)/\delta)$ , such that  $|P(x)| \leq 1$  for all  $x \in [-1, 1]$ , and*

$$\begin{cases} P(x) \in [0, \epsilon], & \forall x \in [-1, -t - \delta] \cup [t + \delta, 1], \text{ and} \\ P(x) \in [1 - \epsilon, 1], & \forall x \in [-t + \delta, t - \delta]. \end{cases}$$

We take  $t = \frac{\lambda_{\min}^q}{2\beta}$  and  $\delta = \frac{\lambda_q}{2\beta}$  for such  $\lambda_q$  that satisfies  $0 < \lambda_q < \lambda_{\min}^q$ . Because the block-encoding of  $\Delta_q^{K,L}$  is not exact, we need to introduce the following lemma on the robustness of singular value transformation:

**Lemma 3 (Lemma 22 of [18])** *If  $P \in \mathbb{C}[x]$  is a degree- $d$  polynomial satisfying the following conditions:*

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

and moreover,  $A, \tilde{A} \in \mathbb{C}^{\tilde{n} \times n}$  are matrices of operator norm at most 1, then we have that

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

Using Lemma 2 and Corollary 1, we can implement a unitary  $\tilde{U}_\Pi$  which is a QSVT of  $\tilde{\Delta}_q^{K,L}/\beta$  with respect to the rectangle function:

$$P^{(\text{SV})}(\tilde{\Delta}_q^{K,L}/\beta) = (|0^{6b+10}\rangle \otimes I_n) \tilde{U}_\Pi (|0^{6b+10}\rangle \otimes I_n).$$

Using Lemma 3 it can be seen that

$$\left\| P^{(\text{SV})}(\Delta_q^{K,L}/\beta) - P^{(\text{SV})}(\tilde{\Delta}_q^{K,L}/\beta) \right\| \leq 4d\sqrt{\|\Delta_q^{K,L}/\beta - \tilde{\Delta}_q^{K,L}/\beta\|},$$

where  $d \in \mathcal{O}(\beta \log(1/\epsilon_{\text{rect}})/\lambda_q)$ .  $P^{(\text{SV})}(\Delta_q^{K,L}/\beta)$  is  $\epsilon_{\text{rect}}$ -close to  $\Pi$  in spectral norm.

Therefore,  $\tilde{U}_\Pi$  is a  $(1, m := 6b + 10, \epsilon_\Pi := \epsilon_{\text{rect}} + 4d\sqrt{\tilde{n}^4 q'^4 \epsilon_{\text{inv}}/\beta})$ -block-encoding of  $\Pi$ . It can be seen that

$$4d\sqrt{\tilde{n}^4 q'^4 \epsilon_{\text{inv}}/\beta} = \mathcal{O}(n^2 q^2 \sqrt{\beta \epsilon_{\text{inv}}} \log(1/\epsilon_{\text{rect}})/\lambda_q) = \mathcal{O}(q^4 n^4 \sqrt{\epsilon_{\text{inv}}/\gamma_q} \log(1/\epsilon_{\text{rect}})/\lambda_q).$$

The number of oracles and gates that are used in the construction of  $\tilde{U}_\Pi$  is as follows:

$$\begin{aligned} O_q^K &: \mathcal{O}\left(\frac{q^4 n^6}{(\gamma_q)^2 \lambda_q} \log\left(\frac{1}{\epsilon_{\text{rect}}}\right) \log\left(\frac{1}{\gamma_q \epsilon_{\text{inv}}}\right)\right), \\ O_{q+1}^L &: \mathcal{O}\left(\frac{q^4 n^6}{(\gamma_q)^2 \lambda_q} \log\left(\frac{1}{\epsilon_{\text{rect}}}\right) \log\left(\frac{1}{\gamma_q \epsilon_{\text{inv}}}\right)\right), \\ \text{gates} &: \tilde{\mathcal{O}}\left(\frac{q^4 n^8}{(\gamma_q)^2 \lambda_q} \log\left(\frac{1}{\epsilon_{\text{rect}}}\right) \log\left(\frac{1}{\gamma_q \epsilon_{\text{inv}}}\right)\right). \end{aligned}$$

## Acknowledgement

RH thanks Tomoyuki Morimae, Seiichiro Tani, and Yuki Takeuchi for the helpful discussion. RH thanks Tomoyuki Morimae for his comments on the draft. RH thanks anonymous reviewers for their helpful comments to improve the manuscript. RH was supported by JSPS KAKENHI Grant Number JP22J11727.

## References

- [1] Mehmet E Aktas, Esra Akbas, and Ahmed El Fatmaoui. Persistence homology of networks: methods and applications. *Applied Network Science*, 4(1):1–28, 2019. DOI: [10.1007/s41109-019-0179-3](https://doi.org/10.1007/s41109-019-0179-3).
- [2] Jonathan Ariel Barmak and Elias Gabriel Minian. Strong homotopy types, nerves and collapses. *Discrete & Computational Geometry*, 47(2):301–328, 2012. DOI: [10.1007/s00454-011-9357-5](https://doi.org/10.1007/s00454-011-9357-5).
- [3] Andreas Bärttschi and Stephan Eidenbenz. Deterministic preparation of dicke states. In *International Symposium on Fundamentals of Computation Theory*, pages 126–139. Springer, 2019. DOI: [10.1007/978-3-030-25027-0\\_9](https://doi.org/10.1007/978-3-030-25027-0_9).
- [4] Gilles Brassard, Peter Hoyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. *Contemporary Mathematics*, 305:53–74, 2002. DOI: [10.1090/conm/305/05215](https://doi.org/10.1090/conm/305/05215).
- [5] Peter Bubenik et al. Statistical topological data analysis using persistence landscapes. *J. Mach. Learn. Res.*, 16(1):77–102, 2015. DOI: [10.5555/2789272.2789275](https://doi.org/10.5555/2789272.2789275).

- [6] Frédéric Chazal and Bertrand Michel. An introduction to topological data analysis: fundamental and practical aspects for data scientists. *Frontiers in artificial intelligence*, 4, 2021. DOI: [10.3389/frai.2021.667963](https://doi.org/10.3389/frai.2021.667963).
- [7] Ho Yee Cheung, Tsz Chiu Kwok, and Lap Chi Lau. Fast matrix rank algorithms and applications. *Journal of the ACM (JACM)*, 60(5):1–25, 2013. DOI: [10.1145/2528404](https://doi.org/10.1145/2528404).
- [8] David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *Discrete & computational geometry*, 37(1):103–120, 2007. DOI: [10.1007/s00454-006-1276-5](https://doi.org/10.1007/s00454-006-1276-5).
- [9] Alex Cole and Gary Shiu. Topological data analysis for the string landscape. *Journal of High Energy Physics*, 2019(3):1–31, 2019. DOI: [10.1007/JHEP03\(2019\)054](https://doi.org/10.1007/JHEP03(2019)054).
- [10] Steven A Cuccaro, Thomas G Draper, Samuel A Kutin, and David Petrie Moulton. A new quantum ripple-carry addition circuit. *arXiv preprint quant-ph/0410184*, 2004. DOI: [10.48550/arXiv.quant-ph/0410184](https://doi.org/10.48550/arXiv.quant-ph/0410184).
- [11] Edoardo Di Napoli, Eric Polizzi, and Yousef Saad. Efficient estimation of eigenvalue counts in an interval. *Numerical Linear Algebra with Applications*, 23(4):674–692, 2016. DOI: [10.1002/nla.2048](https://doi.org/10.1002/nla.2048).
- [12] Robert H Dicke. Coherence in spontaneous radiation processes. *Physical review*, 93(1):99, 1954. DOI: [10.1103/PhysRev.93.99](https://doi.org/10.1103/PhysRev.93.99).
- [13] Herbert Edelsbrunner and John Harer. *Computational topology: an introduction*. American Mathematical Soc., 2010. DOI: [10.1007/978-3-540-33259-6\\_7](https://doi.org/10.1007/978-3-540-33259-6_7).
- [14] Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. Topological persistence and simplification. In *Proceedings 41st annual symposium on foundations of computer science*, pages 454–463. IEEE, 2000. DOI: [10.1007/s00454-002-2885-2](https://doi.org/10.1007/s00454-002-2885-2).
- [15] Herbert Edelsbrunner, John Harer, et al. Persistent homology—a survey. *Contemporary mathematics*, 453:257–282, 2008. DOI: [10.1090/conm/453/08802](https://doi.org/10.1090/conm/453/08802).
- [16] Joel Friedman. Computing betti numbers via combinatorial laplacians. *Algorithmica*, 21(4):331–346, 1998. DOI: [10.1007/PL00009218](https://doi.org/10.1007/PL00009218).
- [17] Robert Ghrist. Barcodes: the persistent topology of data. *Bulletin of the American Mathematical Society*, 45(1):61–75, 2008. DOI: [10.1090/S0273-0979-07-01191-3](https://doi.org/10.1090/S0273-0979-07-01191-3).
- [18] András Gilyén, Yuan Su, Guang Hao Low, and Nathan 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*, pages 193–204, 2019. DOI: [10.1145/3313276.3316366](https://doi.org/10.1145/3313276.3316366).
- [19] Sam Gunn and Niels Kornerup. Review of a quantum algorithm for betti numbers. *arXiv preprint arXiv:1906.07673*, 2019. DOI: [10.48550/arXiv.1906.07673](https://doi.org/10.48550/arXiv.1906.07673).
- [20] Aram W Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical review letters*, 103(15):150502, 2009. DOI: [10.1103/PhysRevLett.103.150502](https://doi.org/10.1103/PhysRevLett.103.150502).
- [21] Ryu Hayakawa. Quantum algorithm for persistent betti numbers and topological data analysis. *arXiv preprint arXiv:2111.00433v1*, 2021. DOI: [10.48550/arXiv.2111.00433](https://doi.org/10.48550/arXiv.2111.00433).
- [22] Ryu Hayakawa, Tomoyuki Morimae, and Suguru Tamaki. Fine-grained quantum supremacy based on orthogonal vectors, 3-sum and all-pairs shortest paths. *arXiv preprint arXiv:1902.08382*, 2019. DOI: [10.48550/arXiv.1902.08382](https://doi.org/10.48550/arXiv.1902.08382).
- [23] Yong He, Ming-Xing Luo, E Zhang, Hong-Ke Wang, and Xiao-Feng Wang. Decompositions of n-qubit toffoli gates with linear circuit complexity. *International Journal of Theoretical Physics*, 56(7):2350–2361, 2017. DOI: [10.1007/s10773-017-3389-4](https://doi.org/10.1007/s10773-017-3389-4).
- [24] He-Liang Huang, Xi-Lin Wang, Peter P Rohde, Yi-Han Luo, You-Wei Zhao, Chang Liu, Li Li, Nai-Le Liu, Chao-Yang Lu, and Jian-Wei Pan. Demonstration of topo-

- logical data analysis on a quantum processor. *Optica*, 5(2):193–198, 2018. DOI: [10.1364/OPTICA.5.000193](https://doi.org/10.1364/OPTICA.5.000193).
- [25] Lek-Heng Lim. Hodge laplacians on graphs. *SIAM Review*, 62(3):685–715, 2020. DOI: [10.1137/18M1223101](https://doi.org/10.1137/18M1223101).
- [26] Lin Lin, Yousef Saad, and Chao Yang. Approximating spectral densities of large matrices. *SIAM review*, 58(1):34–65, 2016. DOI: [10.1137/130934283](https://doi.org/10.1137/130934283).
- [27] Seth Lloyd, Silvano Garnerone, and Paolo Zanardi. Quantum algorithms for topological and geometric analysis of data. *Nature communications*, 7(1):1–7, 2016. DOI: [10.1038/ncomms10138](https://doi.org/10.1038/ncomms10138).
- [28] John M Martyn, Zane M Rossi, Andrew K Tan, and Isaac L Chuang. Grand unification of quantum algorithms. *PRX Quantum*, 2(4):040203, 2021. DOI: [10.1103/PRXQuantum.2.040203](https://doi.org/10.1103/PRXQuantum.2.040203).
- [29] RHAJ Meijer. Clustering using quantum persistent homology. Master’s thesis, 2019.
- [30] Facundo Mémoli, Zhengchao Wan, and Yusu Wang. Persistent laplacians: Properties, algorithms and implications. *SIAM Journal on Mathematics of Data Science*, 4(2):858–884, 2022. DOI: [10.1137/21M1435471](https://doi.org/10.1137/21M1435471).
- [31] Niels Neumann and Sterre den Breeijen. Limitations of clustering using quantum persistent homology. *arXiv preprint arXiv:1911.10781*, 2019. DOI: [10.48550/arXiv.1911.10781](https://doi.org/10.48550/arXiv.1911.10781).
- [32] Nina Otter, Mason A Porter, Ulrike Tillmann, Peter Grindrod, and Heather A Harrington. A roadmap for the computation of persistent homology. *EPJ Data Science*, 6:1–38, 2017. DOI: [10.1140/epjds/s13688-017-0109-5](https://doi.org/10.1140/epjds/s13688-017-0109-5).
- [33] Pratyush Pranav, Herbert Edelsbrunner, Rien Van de Weygaert, Gert Vegter, Michael Kerber, Bernard JT Jones, and Mathijs Wintraecken. The topology of the cosmic web in terms of persistent betti numbers. *Monthly Notices of the Royal Astronomical Society*, 465(4):4281–4310, 2017. DOI: [10.1093/mnras/stw2862](https://doi.org/10.1093/mnras/stw2862).
- [34] Chi Seng Pun, Si Xian Lee, and Kelin Xia. Persistent-homology-based machine learning: a survey and a comparative study. *Artificial Intelligence Review*, pages 1–45, 2022. DOI: [10.1007/s10462-022-10146-z](https://doi.org/10.1007/s10462-022-10146-z).
- [35] Patrick Rall. Faster coherent quantum algorithms for phase, energy, and amplitude estimation. *Quantum*, 5:566, 2021. DOI: [10.22331/q-2021-10-19-566](https://doi.org/10.22331/q-2021-10-19-566).
- [36] Abu Bakar Siddique, Saadia Farid, and Muhammad Tahir. Proof of bijection for combinatorial number system. *arXiv preprint arXiv:1601.05794*, 2016. DOI: [10.48550/arXiv.1601.05794](https://doi.org/10.48550/arXiv.1601.05794).
- [37] Daniel Spitz, Jürgen Berges, Markus Oberthaler, and Anna Wienhard. Finding self-similar behavior in quantum many-body dynamics via persistent homology. *SciPost Phys.*, 11:060, 2021. DOI: [10.21468/SciPostPhys.11.3.060](https://doi.org/10.21468/SciPostPhys.11.3.060). URL <https://scipost.org/10.21468/SciPostPhys.11.3.060>.
- [38] Shashanka Ubaru, Ismail Yunus Akhalwaya, Mark S Squillante, Kenneth L Clarkson, and Lior Horesh. Quantum topological data analysis with linear depth and exponential speedup. *arXiv preprint arXiv:2108.02811*, 2021. DOI: [10.48550/arXiv.2108.02811](https://doi.org/10.48550/arXiv.2108.02811).
- [39] Rui Wang, Duc Duy Nguyen, and Guo-Wei Wei. Persistent spectral graph. *International journal for numerical methods in biomedical engineering*, 36(9):e3376, 2020. DOI: [10.1002/cnm.3376](https://doi.org/10.1002/cnm.3376).
- [40] Larry Wasserman. Topological data analysis. *Annual Review of Statistics and Its Application*, 5:501–532, 2018. DOI: [10.1146/annurev-statistics-031017-100045](https://doi.org/10.1146/annurev-statistics-031017-100045).
- [41] Kelin Xia and Guo-Wei Wei. Persistent homology analysis of protein structure, flexibility, and folding. *International journal for numerical methods in biomedical engineering*, 30(8):814–844, 2014. DOI: [10.1002/cnm.2655](https://doi.org/10.1002/cnm.2655).

## A Proofs of Section 2 and 3

*Proof of Theorem 1.* Choose any orthonormal basis of  $C_{q+1}^{L,K}$  and let  $B_{q+1}^{L,K}$  be the corresponding matrix representation of  $\partial_{q+1}^{L,K}$ . Then  $\Delta_q^{K,L} = B_{q+1}^{L,K} (B_{q+1}^{L,K})^\top + (B_q^K)^\top B_q^K$  using (3). It follows using Claim 1 that

$$\beta^{K,L} = \dim \left( \ker(B_q^K) / \text{im}(B_{q+1}^{L,K}) \right) = \dim \left( \ker(\Delta_q^{K,L}) \right) = \text{nullity} \left( \Delta_q^{K,L} \right)$$

with  $B_q^K B_{q+1}^{L,K} = 0$ .  $\square$

*Proof of Theorem 3.* This is a slight modification of Corollary 18 of [18]. Using Corollary 10 of [18], we can find a degree- $d$  polynomial  $P \in \mathbb{C}[x]$  s.t.  $|\Re[P(x)] - f(x)| \leq \delta$  for all  $x \in [-1, 1]$  and  $P$  satisfies the following conditions:

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

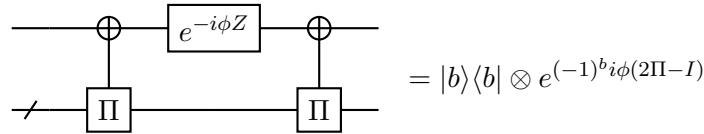
with an  $\mathcal{O}(\text{poly}(d, \log(1/\delta)))$ -time classical algorithm. By Theorem 17 of [18], we can also find a parameter  $\Phi = \{\phi_0, \phi_1, \dots, \phi_d\} \in \mathbb{R}^{d+1}$  in classical  $\mathcal{O}(\text{poly}(d, \log(1/\delta)))$  time s.t. the parametrized circuit  $U_\Phi$  satisfies

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

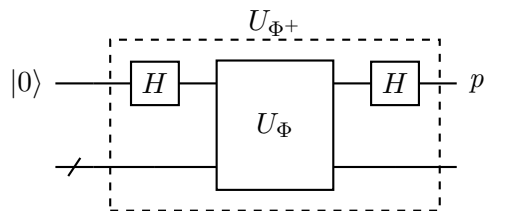
Here,  $U_\Phi$  is a unitary defined as

$$U_\Phi := \begin{cases} e^{i\phi_1(2\tilde{\Pi}-I)}U \prod_{j=1}^{(d-1)/2} \left( e^{i\phi_{2j}(2\Pi-I)}U^\dagger e^{i\phi_{2j+1}(2\tilde{\Pi}-I)}U \right) & \text{for odd } d, \\ \prod_{j=1}^{d/2} \left( e^{i\phi_{2j-1}(2\Pi-I)}U^\dagger e^{i\phi_{2j}(2\tilde{\Pi}-I)}U \right) & \text{for even } d. \end{cases}$$

Each of  $e^{i\phi(2\Pi-I)}$  can be implemented using a single ancilla qubit and  $C_{\Pi}\text{NOT}$  as



for  $b \in \{0, 1\}$ . We can linearly combine  $U_\Phi$  and  $U_{-\Phi}$  by applying Hadamard gates to the ancilla qubit as



where  $p$  means to postselect to  $|0\rangle$ . Define  $U_{\Phi+}$  for odd  $d$  and even  $d$  as the above figure. Let  $\Pi' = \tilde{\Pi}$  for odd  $d$  and let  $\Pi' = \Pi$  for even  $d$ . Then  $P^{(\text{SV})}(\tilde{\Pi}U\Pi) = \Pi'U_{\Phi}\Pi$  and  $P^{*(\text{SV})}(\tilde{\Pi}U\Pi) = \Pi'U_{-\Phi}\Pi$ , and

$$P_{\mathfrak{R}}^{(\text{SV})}(\tilde{\Pi}U\Pi) = \frac{P^{(\text{SV})}(\tilde{\Pi}U\Pi) + P^{*(\text{SV})}(\tilde{\Pi}U\Pi)}{2} = (|0\rangle \otimes \Pi')U_{\Phi+}(|0\rangle \otimes \Pi).$$

This satisfies

$$\left\| f^{(\text{SV})}(\tilde{\Pi}U\Pi) - P_{\mathfrak{R}}^{(\text{SV})}(\tilde{\Pi}U\Pi) \right\| \leq \delta.$$

It is clear from the construction that  $U_{\Phi+}$  can be implemented with  $\mathcal{O}(d)$  use of  $U$ ,  $U^\dagger$ ,  $C_{\Pi}\text{NOT}$ ,  $C_{\tilde{\Pi}}\text{NOT}$  and single-qubit gates.  $\square$

*Proof of Corollary 1.* This follows from the construction of  $U_{\Phi+}$  in the proof of Theorem 3.  $\square$

*Proof of Theorem 5.* This is a slight modification of Theorem 41 of [18] for the block-encoding of the positive semidefinite operator. Let  $\epsilon' = \frac{\alpha\epsilon}{2}$ . It is shown in [18, 28] that there exists an odd real polynomial  $f(x)$  that  $\frac{\epsilon'}{2\kappa}$ -approximates the function  $\frac{1}{2\kappa x}$  for  $\forall x \in [-1, 1] \setminus [-\frac{1}{\kappa}, \frac{1}{\kappa}]$ , and  $|f(x)| \leq 1$  for  $\forall x \in [-1, 1]$  whose degree is  $d = \mathcal{O}(\kappa \log(\kappa/\epsilon'))$ . ( $f(x)$  satisfies  $f(0) = 0$ .)

Let  $A = \sum_i \lambda_i |\tilde{\psi}_i\rangle \langle \psi_i|$  be the singular value decomposition of  $A$ . Then

$$A/\alpha = (|0\rangle^a \otimes I_n)U(|0\rangle^a \otimes I_n) = \sum_i \lambda_i/\alpha |\tilde{\psi}_i\rangle \langle \psi_i|.$$

By Corollary 1, we can construct  $U_{\Phi+}$  that is a QSVT with polynomial  $P(x)$  that  $\frac{\epsilon'}{2\kappa}$ -approximates  $f(x)$  with  $\mathcal{O}(\text{poly}(\kappa \log(\kappa/\epsilon')))$ -time classical algorithm. Then, because  $\lambda_{\min}/\alpha > 1/\kappa$ ,  $U_{\Phi+}$  satisfies

$$(|0\rangle^{a+1} \otimes I_n)U_{\Phi+}(|0\rangle^{a+1} \otimes I_n) = \sum_i P(\lambda_i/\alpha) |\tilde{\psi}_i\rangle \langle \psi_i| \simeq \sum_i \frac{\alpha}{2\kappa\lambda_i} |\tilde{\psi}_i\rangle \langle \psi_i| = \frac{\alpha}{2\kappa} A^+.$$

Therefore,  $(|0\rangle^{a+1} \otimes I_n)U_{\Phi+}(|0\rangle^{a+1} \otimes I_n)$  is an  $(\frac{\epsilon'}{2\kappa} + \frac{\epsilon'}{2\kappa})$ -approximation of  $\frac{\alpha}{2\kappa} A^+$ . This means  $\frac{2\kappa}{\alpha} (|0\rangle^{\otimes a+1} \otimes I_n)U_{\Phi+}(|0\rangle^{\otimes a+1} \otimes I_n)$  is  $\frac{2}{\alpha}\epsilon' = \epsilon$  approximation of  $A^+$ .  $\square$

## B Explicit construction of the membership oracles

It is necessary to be able to construct the membership functions so that we can implement our quantum algorithm efficiently. Although the number of possible  $q$ -simplices can be exponential in the size of the data, we can implement the membership function if we can efficiently verify whether a simplex is contained at some point of the filtration. There are many ways of constructing simplicial complexes from the given data, such as the point cloud, the digital image, or the network [32].

A  $q$ -skeleton of a simplicial complex  $K$  is a union of  $K_p$  for  $p = \{0, 1, \dots, q\}$  where  $K_p$  is the collection of the  $p$ -simplices of  $K$ . We can construct the membership functions efficiently for a class of the simplicial complex which is entirely determined by its 1-skeleton such as the flag complex, clique complex, Vietoris-Rips complex, and lazy witness complex. Besides such simplicial complexes, we are not sure whether we can efficiently implement the membership functions such as the Čech complex and the Alpha complex. In the following, we introduce some constructions of the membership functions.



**Vietoris-Rips complex.** Vietoris-Rips (VR) complex is very commonly used in TDA. This is defined as follows. Let  $S$  be a set of  $n$  points in  $\mathbb{R}^d$ . Then, the VR complex with parameter  $\varepsilon$  denoted by  $\mathcal{R}_\varepsilon(S)$  is the set of all  $\sigma \in S$  such that the largest distance between any of its points is at most  $2\varepsilon$ . We can construct a VR complex first by computing the graph by connecting any of the vertices of which pairwise distance is less than  $2\varepsilon$ . Then, the VR complex is the clique complex of that graph.

In order to check whether a set of vertices is contained in the VR complex, we just need to check all of the pairwise distances of the vertices are less than  $2\varepsilon$ , which can be done in  $\mathcal{O}(n^2)$  time in the worst-case.

It is also possible to implement the membership function for the Vietoris-Rips filtration of the network input [1]. There, an undirected weighted graph is given as input. Then, the filtration is built according to the parameter for the weight of the edges. The membership verification can be done similarly using the adjacency matrix of the undirected graph with some parameters.

**Witness complex.** A small subset of the point cloud is used as a ‘landmark’ to reduce the number of simplices. Let  $S$  be the point set and let  $L \subseteq S$  be the landmark set. Then, the number of possible simplices of the witness complex is  $2^{\mathcal{O}|L|}$ . A point  $s \in S$  is called a weak  $\varepsilon$ -witness for  $\sigma$  iff  $d(s, a) \leq d(s, b)$  for all  $a \in \sigma$  and  $b \in L \setminus \sigma$  where  $d(x, y)$  is the distance points  $x$  and  $y$ . The weak witness complex at scale  $\varepsilon$  is the simplicial complex with vertex  $L$  such that all the faces of that simplicial complex  $\sigma \subset L$  have a weak  $\varepsilon$ -witness in  $S$ . The lazy weak witness complex at scale  $\varepsilon$  has the same 1-skeleton as the weak witness complex. We can efficiently verify membership for the lazy weak witness complex by calculating whether any pair of two points in  $L$  has a weak  $\varepsilon$ -witness.

## C Construction of the block-encoding of the boundary operator

### C.1 Introduction of $U_q^K$

As in eq. (1), the boundary matrix is defined as

$$\partial_q^K([v_0, \dots, v_q]) := \sum_{i=0}^q (-1)^i [v_0, \dots, \hat{v}_i, \dots, v_q].$$

In this section, we construct a unitary  $U_q^K$  that acts as

$$\begin{aligned} & U_q^K \left( |x\rangle \otimes |0^{\otimes \lceil \log(q+1) \rceil}\rangle \otimes |0^{\otimes \lceil \log n \rceil}\rangle \otimes |0^5\rangle \right) \\ &= \frac{1}{\sqrt{\tilde{n}\tilde{q}}} \sum_{s=0}^{\tilde{q}-1} \sum_{t=0}^{\tilde{n}-1} (-1)^s |x \oplus 0^{t-1} 10^{n-t}\rangle (H^{\otimes \lceil \log(q+1) \rceil} |s\rangle) (H^{\otimes \lceil \log n \rceil} |t\rangle) \\ & \quad \otimes |h_{q-1}(s), h_n(t), \delta_{s, g_t(x)} \oplus 1, x_t \oplus 1, f_q^K(x) \oplus 1\rangle \end{aligned} \quad (9)$$

using  $\tilde{\mathcal{O}}(n^2)$ -number of gates and  $\mathcal{O}(1)$ -use of  $O_q^K$ . Here,  $h_a(x) = 0$  if  $x < a$  and  $h_a(x) = 1$  if  $x \geq a$ ,  $g_t(x) := \sum_{i=0}^{t-1} x_i$  ( $x_i$  is the  $i$ -th bit of  $x$ ), and  $\delta_{s, g_t(x)} = 1$  if  $s = g_t(x)$  and  $\delta_{s, g_t(x)} = 0$  otherwise.

## C.2 Verification that $U_q^K$ is the block-encoding of the boundary operator

We show that the unitary  $U_q^K$  satisfies the following equations. First, we show that for  $\forall x \in \bar{S}_q^K \subseteq \{0,1\}^n$ , the unitary  $U_q^K$  satisfies

$$\langle \langle 0|^a \otimes I_n \rangle U_q^K (|0\rangle^a \otimes I_n) |x\rangle = \frac{1}{\tilde{n}\tilde{q}} \partial_q^K |x\rangle = \frac{1}{\tilde{n}\tilde{q}} \sum_{i=0}^q (-1)^i |x \oplus 0^{t_i(x)-1} 10^{n-t_i(x)}\rangle \quad (10)$$

with some  $a = \mathcal{O}(\log(n))$ . Second, we show that for  $\forall x \notin \bar{S}_q^K$ , the unitary  $U_q^K$  satisfies

$$\langle \langle 0|^a \otimes I_n \rangle U_q^K (|0\rangle^a \otimes I_n) |x\rangle = 0. \quad (11)$$

Here,  $t_i(x)$  is the index of the  $i$ -th nonzero bit of  $x$ ,  $\tilde{n} := 2^{\lceil \log n \rceil}$ , and  $\tilde{q} := 2^{\lceil \log(q+1) \rceil}$ . Such  $U_q^K$  is an  $(\tilde{n}\tilde{q}, a, 0)$ -block-encoding of  $\partial_q^K$ .

By projecting the fourth register of eq. (9) to  $|0^5\rangle$ , we obtain

$$\begin{aligned} & \frac{\delta_{f_q^K(x),1}}{\sqrt{\tilde{n}\tilde{q}}} \sum_{s=0}^q \sum_{t=0}^{n-1} (-1)^s \delta_{s,g_t(x)} \delta_{x_t,1} |x \oplus 0^{t-1} 10^{n-t}\rangle (H^{\otimes \lceil \log(q+1) \rceil} |s\rangle) (H^{\otimes \lceil \log n \rceil} |t\rangle) |0^5\rangle \quad (12) \\ & = \frac{\delta_{f_q^K(x),1}}{\sqrt{\tilde{n}\tilde{q}}} \sum_{s=0}^q (-1)^s |x \oplus 0^{t_s(x)-1} 10^{n-t_s(x)}\rangle (H^{\otimes \lceil \log(q+1) \rceil} |s\rangle) (H^{\otimes \lceil \log n \rceil} |t\rangle) |0^5\rangle, \end{aligned}$$

where we denote by  $t_s(x)$  the  $t$  s.t.  $g_t(x) = s$  and  $x_t = 1$ . The  $t_s(x)$  represents the index of the  $s$ -th non-zero bit of  $x$ . Then, by further projecting the second and the third registers to  $|0\rangle$ , we obtain

$$\begin{aligned} & \left( I_n \otimes \langle 0^{\lceil \log(q+1) \rceil} | \otimes \langle 0^{\lceil \log n \rceil} | \otimes \langle 0^5 | \right) U_q^K \left( |x\rangle \otimes |0^{\lceil \log(q+1) \rceil}\rangle \otimes |0^{\lceil \log n \rceil}\rangle \otimes |0^5\rangle \right) \\ & = \frac{\delta_{f_q^K(x),1}}{\tilde{n}\tilde{q}} \sum_{s'=0}^{\tilde{q}-1} \sum_{t'=0}^{\tilde{n}-1} \sum_{s=0}^q (-1)^s |x \oplus 0^{t_s(x)-1} 10^{n-t_s(x)}\rangle \langle s' | s\rangle \langle t' | t_s(x)\rangle \\ & = \frac{\delta_{f_q^K(x),1}}{\tilde{n}\tilde{q}} \sum_{s=0}^q (-1)^s |x \oplus 0^{t_s(x)-1} 10^{n-t_s(x)}\rangle. \end{aligned}$$

Therefore, the unitary  $U_q^K$  satisfies eq. (10) and eq. (11).

## C.3 Construction of $U_q^K$

We construct the unitary  $U_q^K$  in the following 5 steps.

1. Generate the following state:

$$\frac{1}{\sqrt{\tilde{n}\tilde{q}}} \sum_{s=0}^{\tilde{q}-1} \sum_{t=0}^{\tilde{n}-1} |x\rangle |s\rangle |t\rangle |0^5\rangle. \quad (13)$$

This state can be generated by applying  $H^{\otimes \lceil \log(q+1) \rceil}$  and  $H^{\otimes \lceil \log n \rceil}$  to the second and the third registers of  $|x\rangle \otimes |0^{\otimes \lceil \log(q+1) \rceil}\rangle \otimes |0^{\otimes \lceil \log n \rceil}\rangle \otimes |0^5\rangle$ , respectively.

2. Generate the following state:

$$\frac{1}{\sqrt{\tilde{n}\tilde{q}}} \sum_{s=0}^{\tilde{q}-1} \sum_{t=0}^{\tilde{n}-1} |x\rangle |s\rangle |t\rangle |h_{q-1}(s), h_n(t), \delta_{s,g_t(x)} \oplus 1, x_t \oplus 1, f_q^K(x) \oplus 1\rangle, \quad (14)$$

where  $h_a(x) = 0$  if  $x < a$  and  $h_a(x) = 1$  if  $x \geq a$ ,  $g_t(x) = \sum_{i=0}^{t-1} x_i$  ( $x_i$  is the  $i$ -th bit of  $x$ ), and  $\delta_{s,g_t(x)} = 1$  if  $s = g_t(x)$  and  $\delta_{s,g_t(x)} = 0$  otherwise. The computed functions  $|h_{q-1}(s), h_n(t), \delta_{s,g_t(x)} \oplus 1, x_t \oplus 1, f_q^K(x) \oplus 1\rangle$  in the fourth registers are projected to  $|0^5\rangle$  as in eq. (12). By this projection, we can only leave the sums of  $s$  and  $t$  which are needed as the boundary operator, which is the motivation for computing these functions.

The generation of the quantum state of eq. (14) can be done as follows.

- We first compute  $h_{q-1}(s)$  in the first qubit of the fourth register of eq. (13). For computing  $h_{q-1}(s)$ , we first prepare  $|q\rangle$  in an ancilla register and compute whether  $s \leq q-1$  or  $s > q$  using a quantum circuit for subtraction with  $\mathcal{O}(\log n)$  gates [22]. Finally, initialize the ancilla register.
- Similarly, compute  $h_n(t)$  in the second qubit of the fourth register.
- Compute  $g_t(x)$  in ancilla qubits as

$$\frac{1}{\sqrt{\tilde{n}\tilde{q}}} \sum_{s=0}^{\tilde{q}-1} \sum_{t=0}^{\tilde{n}-1} |x\rangle |s\rangle |t\rangle |h_{q-1}(s), h_n(t), 0^3\rangle |g_t(x)\rangle.$$

This can be done using the quantum circuit for addition [10], which uses another single ancilla qubit and  $\tilde{\mathcal{O}}(n^2)$ -gates.

- Flip the first qubit of the fourth register iff  $s \neq g_t(x)$ :

$$\frac{1}{\sqrt{\tilde{n}\tilde{q}}} \sum_{s=0}^{\tilde{q}-1} \sum_{t=0}^{\tilde{n}-1} |x\rangle |s\rangle |t\rangle |h_{q-1}(s), h_n(t), \delta_{s,g_t(x)} \oplus 1, 0^2\rangle |g_t(x)\rangle.$$

- Initialize the ancilla register:

$$\frac{1}{\sqrt{\tilde{n}\tilde{q}}} \sum_{s=0}^{\tilde{q}-1} \sum_{t=0}^{\tilde{n}-1} |x\rangle |s\rangle |t\rangle |h_{q-1}(s), h_n(t), \delta_{s,g_t(x)} \oplus 1, 0^2\rangle |0\dots 0\rangle.$$

- Copy the  $t$ -th bit of  $x$  to the fourth qubit of the fourth register and also apply  $X$  gate to the same qubit:

$$\frac{1}{\sqrt{\tilde{n}\tilde{q}}} \sum_{s=0}^{\tilde{q}-1} \sum_{t=0}^{\tilde{n}-1} |x\rangle |s\rangle |t\rangle |h_{q-1}(s), h_n(t), \delta_{s,g_t(x)} \oplus 1, x_t \oplus 1, 0\rangle.$$

- Apply the membership oracle  $O_q^K$  to the first register and the fifth qubit of the fourth register, and also apply  $X$ -gate to the fifth qubit of the fourth register:

$$\frac{1}{\sqrt{\tilde{n}\tilde{q}}} \sum_{s=0}^{\tilde{q}-1} \sum_{t=0}^{\tilde{n}-1} |x\rangle |s\rangle |t\rangle |h_{q-1}(s), h_n(t), \delta_{s,g_t(x)} \oplus 1, x_t \oplus 1, f_q^K(x) \oplus 1\rangle.$$

3. Apply a  $Z$ -gate to the first qubit of the second register:

$$\frac{1}{\sqrt{\tilde{n}\tilde{q}}} \sum_{s=0}^{\tilde{q}-1} \sum_{t=0}^{\tilde{n}-1} (-1)^s |x\rangle |s\rangle |t\rangle |h_{q-1}(s), h_n(t), \delta_{s,g_t(x)} \oplus 1, x_t \oplus 1, f_q^K(x) \oplus 1\rangle.$$

4. Flip the  $t$ -th bit of the first register:

$$\frac{1}{\sqrt{\tilde{n}\tilde{q}}} \sum_{s=0}^{\tilde{q}-1} \sum_{t=0}^{\tilde{n}-1} (-1)^s |x \oplus 0^{t-1} 10^{n-t}\rangle |s\rangle |t\rangle |h_{q-1}(s), h_n(t), \delta_{s,g_t(x)} \oplus 1, x_t \oplus 1, f_q^K(x) \oplus 1\rangle. \quad (15)$$

5. Apply  $H^{\otimes \lceil \log(q+1) \rceil}$  and  $H^{\otimes \lceil \log n \rceil}$  to the second and the third registers of eq. (15) to obtain

$$\frac{1}{\sqrt{\tilde{n}q}} \sum_{s=0}^{\tilde{q}-1} \sum_{t=0}^{\tilde{n}-1} (-1)^s |x \oplus 0^{t-1} 10^{n-t}\rangle (H^{\otimes \lceil \log(q+1) \rceil} |s\rangle) (H^{\otimes \lceil \log n \rceil} |t\rangle) \\ \otimes |h_{q-1}(s), h_n(t), \delta_{s, g_t(x)} \oplus 1, x_t \oplus 1, f_q^K(x) \oplus 1\rangle.$$

We have implemented the unitary of eq. (9) in the above 5 steps. We have used  $\tilde{\mathcal{O}}(n^2)$ -number of gates and  $\mathcal{O}(1)$ -number of  $O_q^K$  in this construction.