Quantum Persistent Homology

Bernardo Ameneyro, 1,* Vasileios Maroulas, 1,† and George Siopsis^{2,‡}

¹Department of Mathematics, The University of Tennessee, Knoxville, TN 37996-1320, USA ²Department of Physics and Astronomy, The University of Tennessee, Knoxville, TN 37996-1200, USA (Dated: March 1, 2022)

Persistent homology is a powerful mathematical tool that summarizes useful information about the shape of data allowing one to detect persistent topological features while one adjusts the resolution. However, the computation of such topological features is often a rather formidable task necessitating the sub-sampling the underlying data. To remedy this, we develop an efficient quantum computation of persistent Betti numbers, which track topological features of data across different scales. Our approach employs a persistent Dirac operator whose square yields the persistent combinatorial Laplacian, and in turn the underlying persistent Betti numbers which capture the persistent features of data. We also test our algorithm on point cloud data.

I. Introduction

In recent years, topology and geometry blended with statistical methods have seen increasing application to the study of data analysis, visualization, and dimensionality reduction [1–9]. These applications range from classification and clustering in fields such as action recognition [10], handwriting analysis [11], and biology [12–16], to classification of high entropy alloy [17] and gas separation [18], to the analysis of complex biological networks [19], and other complex dynamical systems [20, 21] and sensor networks [22–26].

Persistent homology [27–31], the workhorse of topological data analysis (TDA), has helped to compress nonlinear point cloud data from a new geometrically faithful point of view. In the realm of signal analysis, classification and clustering based on topological features of the signal identifies features that traditional signal analysis techniques fail to detect [10, 12, 32–34]. Topological features, such as the number of connected components, cycles, and higher-dimensional voids, respectively, represent multi-stability, periodicity, and chaos in a dynamical system [12, 35].

The effective computation of persistence diagrams has become an area of intense active research, including a significant successful effort toward facilitating previously challenging computations. For example, the creation of persistence diagrams with packages such as Dionysus [36] and Ripser [37] take advantage of certain properties of simplicial complexes [38]. Point clouds typically consist of many points. A set of n such points possesses 2^n potential subsets that could contribute to the topology. The best classical algorithm for estimating these homological features of a data set with accuracy α takes time $O(2^n \log(1/\alpha))$.

More recently, quantum algorithms that compute topological features of data were developed. The first such algorithm was introduced in [39] with run time $O(n^5/\alpha)$ – exponentially faster than the best known classical algorithms. This algorithm was designed for a discrete-variable (DV) quantum system based on qubits, and it was extended to a continuous-variable (CV) substrate in [40]. These two algorithms compute Betti numbers by analyzing a linear operator called the

^{*} bamenevr@vols.utk.edu

[†] vasileios.maroulas@utk.edu

[‡] siopsis@tennessee.edu

Dirac operator whose square is the combinatorial Laplacian. However, as was noted in [41], these algorithms do not compute *persistent* topological features so that one may track how topological features persist as the resolution of data changes and the underlying noise may vary. Recently, a quantum computation of persistent Betti numbers based on block-encoding and quantum singular value transformation was introduced in [42]. However, the study did not address any real implementation on data. To that end, a few recent attempts on data for tracking the non-persistent topological features were studied. Indeed, the work in [43] demonstrates a quantum algorithm presented in [39] by employing a six-photon quantum processor to successfully analyze the topological features of a network including three data points. Moreover the work in [44] developed a quantum annealing approach for topologically analyzing point cloud data, and [45] gives a quantum circuit to construct all maximal cliques of an n-node network using Grover's search algorithm.

The persistent combinatorial Laplacian is expressed as a quadratic function of a pertinent linear map called the boundary operator. However, it is computationally efficient to consider a map linear on the boundary. Here, we introduce a quantum algorithm that relies on a (persistent) Dirac operator which is linear in the boundary operator. Its square yields the persistent combinatorial Laplacian, hence the persistent Betti numbers, allowing us to track topological features across different scales and obtain persistence information. To that end, our algorithm is a generalization of the algorithm introduced in [39], and it similarly offers an exponential speedup over known classical algorithms of persistent homology.

Our discussion is organized as follows. In Section II, we review persistent homology introducing the important concepts relevant to the algorithm, such as the persistent combinatorial Laplacian and persistent Betti numbers. In Section III, we present our quantum algorithm introducing the persistent Dirac operator together with an outline of the quantum tools that are needed, such as quantum random access memory (QRAM), Grover's search algorithm, and phase estimation. In Section IV, we apply our algorithm to a data set whose points are organized in squares, which was suggested in [41] as a case in which persistent Betti numbers differ from Betti numbers. Finally in Section V, we offer concluding remarks. Details of subroutines needed for our quantum algorithm are provided in Appendices A, B, and C (Grover's algorithm, implementation of an exponential operator, and phase estimation, respectively).

II. Persistent Homology

Persistent homology studies objects called simplicial complexes. The classical algorithms that perform topological data analysis use the data to construct simplicial complexes, e.g., by connecting all the points in a point cloud within a certain distance from each other (Vietoris-Rips complex) and then varying the distance to obtain a filtration of simplicial complexes. After that, the algorithms proceed to compute the eigenvalues and eigenvectors of linear operators, such as the boundary operator and the persistent combinatorial Laplacian that act on the complexes.

Next, we briefly discuss simplicial complexes and homology, an algebraic descriptor for coarse shape in topological spaces. In turn, persistent homology, and its summary, persistence diagrams, are techniques for bringing the power and convenience of homology to the description of subspace filtrations of topological spaces. We consider topological spaces of discernible dimension called manifolds.

Definition 1. A topological space X is called a k-dimensional manifold if every point $x \in X$ has a neighborhood which is homeomorphic to an open neighborhood in k-dimensional Euclidean space.

We generalize the fixed-dimension notion of a manifold in order to define simplicial homology for simplicial complexes. We then discuss the Vietoris-Rips construction which is used to associate simplicial complexes to datasets.

Definition 2. A k-simplex is a collection of k+1 linearly independent vertices along with all convex combinations of these vertices:

$$\sigma = [v_0, ..., v_k] = \left\{ \sum_{i=0}^k \alpha_i v_i : \sum_{i=0}^k \alpha_i = 1 \text{ and } \alpha_i \ge 0 \,\forall i \right\}. \tag{1}$$

Topologically, a k-simplex is treated as a k-dimensional manifold (with boundary). An oriented simplex is typically described by a list of its vertices, such as $[v_0, v_1, v_2]$. The faces of a simplex consist of all the simplices built from a subset of its vertex set; for example, the edge $[v_1, v_2]$ and vertex $[v_2]$ are both faces of the triangle $[v_0, v_1, v_2]$.

Definition 3. A simplicial complex K is a collection of simplices wherein

- (i) if $\sigma \in \mathcal{K}$, then all its faces are also in \mathcal{K} , and
- (ii) the intersection of any pair of simplices in K is another simplex in K.

We denote the collection of k-simplices within K by K_k .

Conditions (i) and (ii) in Definition 3 establish a unique topology on the realization of a simplicial complex which restricts to the subspace topology on each open simplex. For finite simplicial complexes realized in \mathbb{R}^d , this topology is also consistent with the Euclidean subspace topology.

Here we define the homology groups for a simplicial complex through purely combinatorial means, which allows for automated computation.

Definition 4. The chain group (over \mathbb{Z}_3) on a simplicial complex K of dimension k is denoted by $C_k(K)$ and is defined as formal sums of k-simplices in K:

$$C_k(\mathcal{K}) = \left\{ \sum_{\sigma \in \mathcal{K}_k} n_{\sigma} \sigma : n_{\sigma} \in \mathbb{Z}_3 \right\}.$$
 (2)

Definition 5. The k-th boundary map is a homomorphism $\partial_k : C_k(\mathcal{K}) \to C_{k-1}(\mathcal{K})$ defined on each simplex as an alternating sum over the faces of dimension k-1:

$$\partial_k[v_0, ..., v_k] = \sum_{n=0}^k (-1)^n [v_0, ..., v_{n-1}, v_{n+1}, ..., v_k].$$
(3)

Chain groups give an algebraic way to describe subsets of simplices as a formal sum. Toward this viewpoint, the chain group is often defined over $\mathbb{Z}_2 = \{0,1\}$ instead of \mathbb{Z}_3 . In this case, the boundary maps can be understood classically; e.g., the boundary of a triangle yields (the sum of) its three edges and the boundary of an edge yields (the sum of) its endpoints. When viewed over \mathbb{Z}_3 , the presence of sign specifies simplex orientation.

Putting chain groups of every dimension together along with the boundary maps successively defined between them, we obtain a chain complex:

$$\{0\} \stackrel{\mathbf{0}}{\leftarrow} C_0(\mathcal{K}) \stackrel{\partial_1}{\leftarrow} C_1(\mathcal{K}) \stackrel{\partial_2}{\leftarrow} C_2(\mathcal{K}) \stackrel{\partial_3}{\leftarrow} \dots$$
 (4)

The composition of subsequent boundary maps yields the trivial map [3]; this property is typically rephrased as $\operatorname{im}(\partial_{k+1}) \subset \ker(\partial_k)$ which enables definition of the following modular groups.

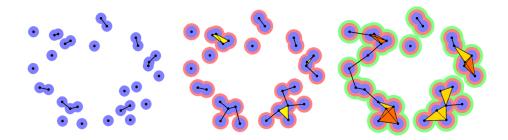


FIG. 1. The neighborhood space and VR complex of matching scale plotted at three different scales. Yellow indicates a triangle while orange indicates a tetrahedron. This family of simplicial complexes is the filtration utilized to compute and define persistent homology.

Definition 6. The homology group of dimension k is given by

$$H_k(\mathcal{K}) = \ker(\partial_k) / \operatorname{im}(\partial_{k+1}) = \{ \tilde{x} = x + \operatorname{im}(\partial_{k+1}) : x \in \ker(\partial_k) \}, \tag{5}$$

where $\tilde{x} = \{x + y : y \in \operatorname{im}(\partial_{k+1})\}$ defines the coset equivalence class of x.

The dimension of H_k is the k-th Betti number, denoted by β_k . It is shown in [46] that the k-th Betti number is also the dimension of the kernel of the k-th combinatorial Laplacian $\mathcal{L}_k^{\mathcal{K}}: C_k(\mathcal{K}) \to C_k(\mathcal{K})$ given by $\mathcal{L}_k^{\mathcal{K}} = \partial_k^{\dagger} \partial_k + \partial_{k+1} \partial_{k+1}^{\dagger}$, where ∂_k^{\dagger} denotes the dual.

The generators of the homology group correspond to topological features of the complex K; for example, generators for the 0-homology group correspond to connected components, generators of 1-homology group correspond to holes in K, etc. The interpretation of these features is exemplified by taking the topological boundary of a k+1 ball (that is, a k-sphere); for example, the boundary of an interval is two (disconnected) points while the boundary of a disc is a loop.

We wish to extend the notion of homology for a discrete set of data $\boldsymbol{x} = \{v_i\}_{i=1}^N$ within a metric space (X, d_X) . Treating the set itself as a simplicial complex, its homology yields only the cardinality of the data points. So, we utilize the metric to obtain more information. Here we denote by $B(x_0, r_0)$ a metric ball centered at x_0 of radius r_0 . Fix a radius r > 0 and consider the collection of neighborhoods $U = \{U_i\} = \{B(v_i, r)\}$ along with its union $\mathcal{U}_r = \cup_i B(v_i, r)$. The collection of sets $\{\mathcal{U}_r\}_{r \in \mathbb{R}^+}$ naturally yields information about the arrangement within X of the dataset \boldsymbol{x} at various scales. To make homology computations more tractable for \mathcal{U}_r , people often consider instead the associated Vietoris-Rips (VR) complexes which is (topologically) very similar.

Definition 7. The Vietoris-Rips complex of the data $\{v_i\}$ at scale ϵ , denoted by $S^{\epsilon}(\boldsymbol{x})$ (or simply S^{ϵ}), is the simplicial complex where a k-simplex $\sigma = [v_{i_0}, ..., v_{i_k}]$ is in $S^{\epsilon}(\boldsymbol{x})$ if and only if $diam(\sigma) = \max_{j,j'} \left\{ d_X(v_{i_j}, v_{i_{j'}}) \right\} \leq \epsilon$.

Examples of the VR complex for the same data at different scales are depicted in Fig. 1, where they are superimposed with the associated neighborhood space.

Unfortunately the tools defined so far would only allow to obtain the Betti numbers of a fixed scale ϵ , which is not very useful to analyze data because there is no way to differentiate between true topological features and noise. To tell them apart one must track the topological features across scales, those that persist for a long time are the true topological features, while those that

persist for a short time are just noise. So, we must extend the notions above to include multiple simplicial complexes at once.

Consider a nested sequence or filtration of simplicial complexes

$$\emptyset \subseteq \mathcal{K}_1 \subseteq \mathcal{K}_2 \subseteq \dots \subseteq \mathcal{K}_n \tag{6}$$

each complex \mathcal{K}_i has associated chain groups $C_k(\mathcal{K}_i)$, boundary operators ∂_k^i and Homology groups H_k^i as before. But now the inclusion maps $\iota: \mathcal{K}_i \to \mathcal{K}_j$ between complexes induce homomorphisms $h_k^{i,j} = \iota_*: C_k(\mathcal{K}_i) \to C_k(\mathcal{K}_j)$ between their corresponding chain groups.

Definition 8. For two nested simplicial complexes $K_i \subseteq K_j$ we define their k-th persistent Homology group $H_k^{i,j}$ as

$$H_k^{i,j} = \operatorname{Im}(h_k^{i,j}) = \operatorname{Ker}(\partial_k^i) / (\operatorname{Im}(\partial_{k+1}^j) \cap \operatorname{Ker}(\partial_k^i))$$
 (7)

where $Ker(\partial_k^i)$ is viewed as a subgroup of $Ker(\partial_k^j)$. The dimension of $H_k^{i,j}$ is the k-th persistent Betti number $\beta_k^{i,j}$.

Let $\tilde{C}_{k+1}^{i,j} = \left\{ x \in C_{k+1}(\mathcal{K}_j) : \partial_{k+1}^j x \in C_k(\mathcal{K}_i) \right\}$, that is, the subgroup of $C_{k+1}(\mathcal{K}_j)$ defined by the k+1-simplices in \mathcal{K}_j with boundary in \mathcal{K}_i . Then we can define the k-th persistent combinatorial Laplacian

$$\mathcal{L}_{k}^{i,j} = \tilde{\partial}_{k}^{i,i\dagger} \tilde{\partial}_{k}^{i,i} + \tilde{\partial}_{k+1}^{i,j} \tilde{\partial}_{k+1}^{i,j\dagger}, \tag{8}$$

where $\tilde{\partial}_{k+1}^{i,j}$ is the restriction of the boundary operator ∂_{k+1}^{j} to $\tilde{C}_{k+1}^{i,j}$. Just as before, [46] shows that the dimension of the kernel of $\mathcal{L}_{k}^{i,j}$ is the k-th persistent Betti number $\beta_{k}^{i,j}$.

Going back to the VR complexes introduced earlier, it follows from their definition that if $\epsilon \leq \epsilon'$ then $S^{\epsilon} \subseteq S^{\epsilon'}$. Moreover, since there is only a finite number of data points, S^{ϵ} will remain unchanged for all $\epsilon > \operatorname{diam}(\boldsymbol{x})$. So, there is a maximal VR complex S that contains all other VR complexes S^{ϵ} , it's left to the reader to check that if the data consists of n points then S has exactly 2^n simplices. Choosing an increasing sequence of scales $0 < \epsilon_1 \leq \epsilon_2 \leq \ldots \epsilon_N$ yields a filtration of VR complexes $\emptyset \subseteq S^{\epsilon_1} \subseteq S^{\epsilon_2} \subseteq \ldots \subseteq S^{\epsilon_N} = S$.

III. Quantum algorithm

Quantum algorithms proceed by first mapping the simplices, simplicial complexes, and filtrations onto quantum states, and then implementing linear unitary operators acting on them in order to extract topological features through measurements of quantum states. All 2^n possible simplices can be mapped onto quantum states in an n-qubit Hilbert space enabling an exponential speedup.

Our quantum algorithm uses a quantum Random Access Memory (QRAM) to access (or estimate) data in quantum parallel and create quantum states that encode the VR complexes. Indeed, the k-simplices, which are fully connected sets of k+1 vertices, are encoded as quantum states over n qubits with k+1 qubits corresponding to the vertices of the simplex in the state $|1\rangle$ and the rest of the n-k-1 qubits in the state $|0\rangle$. The boundary of a k-simplex can be obtained as a superposition of all the (k-1)-simplices that result from flipping one of the qubits in the state $|1\rangle$ to $|0\rangle$. Furthermore, the projection operators necessary to express the persistent combinatorial Laplacian

can be implemented using an oracle based on Grover's search algorithm. Then, one may form a superposition of simplices and use a quantum phase estimation algorithm to extract information about the kernel of the persistent combinatorial Laplacian, thus obtaining the desired persistent Betti numbers. It is important to note that the resources needed to encode the simplicial complex (number of qubits) are only polynomial in the number of data points n, while in the classical case, the resources needed grow exponentially with n.

A. Encoding simplices and filtrations as quantum states

A k-simplex $\sigma = [v_{i_1}, \ldots, v_{i_k}]$ can be stored in a n-qubit register as $|\sigma\rangle = |v_1\rangle \otimes \cdots \otimes |v_n\rangle$, with 1s at the positions of its k+1 vertices v_{i_1}, \ldots, v_{i_k} and 0s elsewhere. Let S denote the collection of all possible states of n qubits, then S effectively encodes the maximal VR complex described in section II with all possible simplices that can be formed with n vertices or data points. Moreover, \mathcal{H} , the Hilbert space over \mathbb{Z}_3 with basis S encodes the chain group of that complex.

We denote by S_k the subset of states in S encoding k-simplices, and by \mathcal{H}_k the corresponding subspace of \mathcal{H} which encodes the k-th chain group defined in (2). We can encode the order k of a simplex σ in a state $|k\rangle$ ($k = 0, 1, \ldots, n-1$) by starting from $|0\rangle$ and performing permutations $0 \to 1 \to \cdots \to n-1 \to 0$, conditional upon the corresponding digit of σ being 1. Thus, we perform k permutations mapping $|0\rangle \to |k\rangle$. This can be implemented efficiently because the permutation is a 1-sparse matrix.

Similarly, we write S^{ϵ} for the subset of S encoding simplices of diameter at most ϵ , in other words the VR complex at scale ϵ as in Def. 7, and \mathcal{H}^{ϵ} for the corresponding subspace of \mathcal{H} . To encode the scale ϵ we need information on the data points that can be stored in quantum parallel in QRAM, if it is available, and accessed efficiently [47, 48]. For any $i, j = 1, 2, \ldots, n$, QRAM $|i\rangle|j\rangle|0\rangle = |i\rangle|j\rangle|d(i,j)\rangle$, where d(i,j) is the distance between points i and j. We introduce a register of qubits to record the parameter ϵ as $|\epsilon\rangle$. We need to know when $d(i,j) \leq \epsilon$ to form a VR complex. This information will be stored in a qubit initially in the state $|0\rangle$, and flipped if the membership condition is satisfied. This is implemented with a unitary test that uses the qubit registers storing d(i,j) and ϵ as controls to flip the last qubit,

$$U_{\text{test}}^{\epsilon}|d(i,j)\rangle|\epsilon\rangle|0\rangle = |d(i,j)\rangle|\epsilon\rangle|a^{\epsilon}(i,j)\rangle , \quad a^{\epsilon}(i,j) = \begin{cases} 0 , d(i,j) > \epsilon \\ 1 , d(i,j) \le \epsilon \end{cases}$$
 (9)

Next, in order to know if $\sigma \in S^{\epsilon}$, we must check if $d(i,j) \leq \epsilon$ for all (i,j) pairs such that $v_i = v_j = 1$. To this end, we make $\mathcal{O}(k^2)$ calls to QRAM, where k is the dimension of σ . For each pair (i,j), we use $|\sigma\rangle$ as control to call QRAM and apply the test provided $v_i = v_j = 1$,

$$QRAM^{\dagger}U_{\text{test}}^{\epsilon}QRAM|\sigma\rangle|i\rangle|j\rangle|0\rangle|\epsilon\rangle|0\rangle = |\sigma\rangle|i\rangle|j\rangle|0\rangle|\epsilon\rangle|a^{\epsilon}(i,j)\rangle$$
(10)

Membership of σ in the VR complex S^{ϵ} is decided if for all (i, j) we end up with $a^{\epsilon}(i, j) = 1$. The above steps allow us to implement the quantum oracle

$$\mathcal{O}_k^{\epsilon} |\sigma\rangle |0\rangle = |\sigma\rangle |a_{\sigma}^{\epsilon}\rangle , \quad a_{\sigma}^{\epsilon} = \begin{cases} 0 , \sigma \notin S_k^{\epsilon} \\ 1 , \sigma \in S_k^{\epsilon} \end{cases}$$
 (11)

determining membership in $S_k^{\epsilon} = S_k \cap S^{\epsilon}$, which encodes the k-simplices in the ϵ -complex. The oracle in Eq. (11) can be used for the implementation of the projection operators,

$$P_k^{\epsilon} = \sum_{\sigma \in S_k^{\epsilon}} |\sigma\rangle\langle\sigma|,\tag{12}$$

onto \mathcal{H}_k^{ϵ} , the subspace of the Hilbert space \mathcal{H}_k spanned by the simplices in S_k^{ϵ} , which are needed to construct the persistent combinatorial Laplacian. To implement the projection P_k^{ϵ} , we perform amplitude amplification [49] based on Grover's search algorithm [50] (for details, see Appendix A).

B. Persistent Dirac operator

The boundary operator ∂_k mapping \mathcal{H}_k to \mathcal{H}_{k-1} as defined in (3) may be encoded using 1-qubit Pauli X gates, and expressed similarly by its action on the bases S_k and S_{k-1}

$$\partial_k |\sigma_k\rangle := \sum_{l=0}^k (-)^l X_{i_l} |\sigma_k\rangle = \sum_{l=0}^k (-)^l |\sigma_{k-1}(l)\rangle, \tag{13}$$

where $|\sigma_{k-1}(l)\rangle$ is the (k-1)-simplex obtained from $|\sigma\rangle$ by flipping its l-th non-zero qubit. One easily deduces $\partial_{k-1}\partial_k = 0$.

For persistent homology we need to restrict the boundary operator (13) as in the prelude to Eq. (8),

$$\tilde{\partial}_k^{\epsilon,\epsilon'} = P_{k-1}^{\epsilon} \partial_k P_k^{\epsilon'} , \qquad (14)$$

where the projections are defined in (12). Then the persistent combinatorial Laplacian $\mathcal{L}_k^{\epsilon,\epsilon'}$ is as in (8) with $\mathcal{K}_i = S^{\epsilon}$ and $\mathcal{K}_j = S^{\epsilon'}$.

Next, we introduce the persistent Dirac operator which plays a central role in our quantum algorithm. For quantum computation, it is advantageous to work with an operator which is linear in the boundary map. Unlike the persistent combinatorial Laplacian, the persistent Dirac operator is linear in the boundary operator, and its square yields the persistent combinatorial Laplacian given in Eq. (8).

We define the k-th persistent Dirac operator as the Hermitian operator

$$B_k^{\epsilon,\epsilon'} = \begin{pmatrix} 0 & \tilde{\partial}_k^{\epsilon,\epsilon} & 0\\ \tilde{\partial}_k^{\epsilon,\epsilon\dagger} & 0 & \tilde{\partial}_{k+1}^{\epsilon,\epsilon'}\\ 0 & \tilde{\partial}_{k+1}^{\epsilon,\epsilon'\dagger} & 0 \end{pmatrix} - \xi \begin{pmatrix} P_{k-1}^{\epsilon} & 0 & 0\\ 0 & -P_k^{\epsilon} & 0\\ 0 & 0 & P_{k+1}^{\epsilon'} \end{pmatrix}, \tag{15}$$

where $\xi \in \mathbb{R}$ is an arbitrary parameter. Its square is

$$\left(B_{k}^{\epsilon,\epsilon'}\right)^{2} = \begin{pmatrix}
\tilde{\partial}_{k}^{\epsilon,\epsilon}\tilde{\partial}_{k}^{\epsilon,\epsilon\dagger} & 0 & \tilde{\partial}_{k}^{\epsilon,\epsilon}\tilde{\partial}_{k+1}^{\epsilon,\epsilon'} \\
0 & \mathcal{L}_{k}^{\epsilon,\epsilon'} & 0 \\
\tilde{\partial}_{k+1}^{\epsilon,\epsilon'\dagger}\tilde{\partial}_{k}^{\epsilon,\epsilon\dagger} & 0 & \tilde{\partial}_{k+1}^{\epsilon,\epsilon'\dagger}\tilde{\partial}_{k+1}^{\epsilon,\epsilon'}
\end{pmatrix} + \xi^{2} \begin{pmatrix}
P_{k-1}^{\epsilon} & 0 & 0 \\
0 & P_{k}^{\epsilon} & 0 \\
0 & 0 & P_{k+1}^{\epsilon'}
\end{pmatrix}$$
(16)

written in block form with one of the blocks being the kth persistent combinatorial Laplacian shifted by ξ^2 .

Let λ be an eigenvalue of $B_k^{\epsilon,\epsilon'}$. Evidently, the corresponding eigenspace corresponds to an eigenspace of the Laplacian $\mathcal{L}_k^{\epsilon,\epsilon'}$ with eigenvalue $\gamma=\lambda^2-\xi^2$. Considering $\gamma=0$, one gets the the kernel of $\mathcal{L}_k^{\epsilon,\epsilon'}$, whose dimension is the kth persistent Betti number, $\beta_k^{\epsilon,\epsilon'}=|\ker\mathcal{L}_k^{\epsilon,\epsilon'}|$. In other words, one is interested in the eigenspaces of $B_k^{\epsilon,\epsilon'}$ with eigenvalues $\lambda=\pm\xi$. For the persistent Betti numbers, we focus on the eigenspace with $\lambda=\xi$, because there is a one-one correspondence between its states and those in the null space of the Laplacian. Thus, although ξ is arbitrary, we need to choose $\xi\neq 0$ in order to separate the eigenspaces with $\lambda=-\xi$ and $\lambda=\xi$. The choice $\xi=0$ leads to overcounting of the states in the null space of the Laplacian and an incorrect estimate of Betti numbers.

To see this, consider the action on vectors of the form $(X_{k-1}^{\epsilon}, Y_k^{\epsilon}, Z_{k+1}^{\epsilon'})^T$ which satisfy the constraints $P^{\epsilon}X_k^{\epsilon} = X_k^{\epsilon}$, $P^{\epsilon}Y_k^{\epsilon} = Y_k^{\epsilon}$, and $P^{\epsilon'}Z_k^{\epsilon'} = Z_k^{\epsilon'}$. For eigenvectors of $B_k^{\epsilon,\epsilon'}$ corresponding to eigenvalue λ , we obtain the system of equations

$$-\xi X_{k-1}^{\epsilon} + \tilde{\partial}_{k}^{\epsilon,\epsilon} Y_{k}^{\epsilon} = \lambda X_{k-1}^{\epsilon}$$

$$\xi Y_{k}^{\epsilon} + \tilde{\partial}_{k}^{\epsilon,\epsilon\dagger} X_{k-1}^{\epsilon} + \tilde{\partial}_{k+1}^{\epsilon,\epsilon'} Z_{k+1}^{\epsilon'} = \lambda Y_{k}^{\epsilon}$$

$$-\xi Z_{k+1}^{\epsilon'} + \tilde{\partial}_{k+1}^{\epsilon,\epsilon'\dagger} Y_{k}^{\epsilon} = \lambda Z_{k+1}^{\epsilon'}$$
(17)

For $\lambda \neq -\xi$, we can use two of the equations to express X_{k-1}^{ϵ} and $Z_{k+1}^{\epsilon'}$ in terms of Y_k^{ϵ} , and the second equation to show that Y_k^{ϵ} is an eigenvector of the Laplacian with eigenvalue γ . Conversely, given an eigenvector Y_k^{ϵ} of the Laplacian with eigenvalue γ , the vector $(X_{k-1}^{\epsilon}, Y_k^{\epsilon}, Z_{k+1}^{\epsilon'})^T$ with $X_{k-1}^{\epsilon} = \frac{1}{\lambda + \xi} \tilde{\partial}_k^{\epsilon, \epsilon} Y_k^{\epsilon}$ and $Z_{k+1}^{\epsilon'} = \frac{1}{\lambda + \xi} \tilde{\partial}_{k+1}^{\epsilon, \epsilon' \dagger} Y_k^{\epsilon}$ is an eigenvector of the Dirac operator with eigenvalue λ , demonstrating a one-one correspondence between the two eigenspaces.

For $\lambda = -\xi$, we obtain eigenvectors with Y_k^{ϵ} in the kernel of the Laplacian. We need to separate them from the eigenvectors with $\lambda = \xi$ in order not to overcount the vectors in the null space of the Laplacian.

The persistent Dirac operator (15) can be decomposed as

$$B_{k}^{\epsilon,\epsilon'} = P_{k}^{\epsilon,\epsilon'} B_{k} P_{k}^{\epsilon,\epsilon'}, \ B_{k} = \begin{pmatrix} -\xi I & \partial_{k} & 0\\ \partial_{k}^{\dagger} & \xi I & \partial_{k+1}\\ 0 & \partial_{k+1}^{\dagger} & -\xi I \end{pmatrix}, \ P_{k}^{\epsilon,\epsilon'} = \begin{pmatrix} P_{k-1}^{\epsilon} & 0 & 0\\ 0 & P_{k}^{\epsilon} & 0\\ 0 & 0 & P_{k+1}^{\epsilon'} \end{pmatrix}, \tag{18}$$

To encode these 3×3 block matrices, we introduce a qutrit label $|b\rangle$, with $b \in \{-1, 0, +1\}$, and consider the space spanned by $|b\rangle|\sigma_{k+b}^b\rangle$, where $\sigma_{k+b}^b \in S_{k+b}^{\epsilon_b}$, with $\epsilon_b = \epsilon$ for b = -1, 0 and $\epsilon_b = \epsilon'$ for b = +1. Then we may write

$$P_k^{\epsilon,\epsilon'} = \sum_{b \in \{-1,0,+1\}} |b\rangle\langle b| \otimes P_{k+b}^{\epsilon_b} , \qquad (19)$$

and

$$B_k = \sum_{b \in \{0,+1\}} |b-1\rangle\langle b| \otimes \partial_{k+b} + \text{h.c.} + \xi \sum_{b \in \{-1,0,+1\}} (-)^b |b\rangle\langle b| \otimes I$$
 (20)

The projections $P_k^{\epsilon,\epsilon'}$ can be implemented using Grover's algorithm as outlined above and described in detail in Appendix A, whereas the unrestricted Dirac operator B_k can be encoded in a qubit register. For $\xi = 1$, a qutrit suffices for the encoding of B_k , because all of its matrix elements are in the set $\{-1,0,+1\}$.

C. Persistent Betti numbers

As discussed earlier, the kth persistent Betti number $\beta_k^{\epsilon,\epsilon'}$ is the dimension of the eigenspace of the persistent Dirac operator $B_k^{\epsilon,\epsilon'}$ (Eq. (15)) with eigenvalue $\lambda = \xi$. To compute it, we apply phase estimation (for details, see Appendix C), and derive the probability distribution

$$\mathcal{P}(p) = \frac{1}{N} \sum_{\lambda} g_{\lambda}(p) , \quad g_{\lambda}(p) = \frac{1}{M^2} \frac{\sin^2 \pi l \lambda}{\sin^2 \frac{\pi (l\lambda - p)}{M}} , \qquad (21)$$

where l, M are arbitrary positive integers, $p = 0, 1, \ldots, M-1$, and N is the number of eigenvalues λ of $B_k^{\epsilon,\epsilon'}$. One easily checks that $\sum_{p=0}^{M-1} g_{\lambda}(p) = 1$ for any λ , therefore, $\sum_{p=0}^{M-1} \mathcal{P}(p) = 1$, as expected. If one is interested in the spectrum of the persistent Dirac operator, M and l must be chosen in a way that the spectrum is covered with the different values of p, i.e., each eigenvalue is approximated as $\lambda \approx \frac{p}{l}$. Then $g_{\lambda}(p)$ is strongly peaked at $p \approx l\lambda$. At the peak, $g_{\lambda}(p) = 1$. Therefore, the probability distribution $\mathcal{P}(p)$ has a peak at each eigenvalue of the persistent Dirac operator whose height is proportional to the dimension of the corresponding eigenspace.

Since we are only interested in the eigenvalue $\lambda = \xi$, we only need to capture the peak at $p \approx l\xi$ and make sure it is resolved from other neighboring peaks. The persistent Betti number is obtained as $\beta_k^{\epsilon,\epsilon'} = g_{\xi}(p)|_{p\approx l\xi}$.

IV. An application

This Section demonstrates how our algorithm computes persistence Betti numbers, which track topological features across different scales. This extends previous work in [39] and [40] where the proposed quantum algorithms calculated Betti numbers only without addressing persistence features. To do this, we apply our method to the data set suggested in [41] and described below for the sake of completeness.

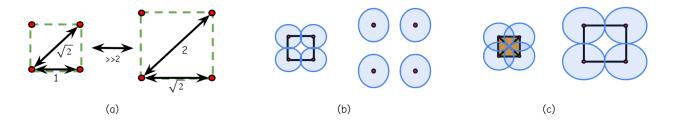


FIG. 2. (a) The two squares example as appeared in [41]; (b) Construction of the VR for ϵ_1 ; (c) VR for ϵ_2

Consider a point cloud of 8 points consisting of two well-separated squares, as in Figure 2. The smaller square has sides of length 1, while the larger square has sides of length $\sqrt{2}$. The distance between the two squares exceeds 2. It is easy to see (Figure 2) that the smaller square produces a loop in the VR complex at scale 1 which disappears at scale $\sqrt{2}$, while at the same time the larger square produces a new loop. It follows that for $1 < \epsilon_1 < \sqrt{2} < \epsilon_2 < 2$, the one-dimensional

persistent Betti numbers corresponding to the point cloud of Figure 2 are

$$\beta_1^{\epsilon_1,\epsilon_1} = 1 \; , \; \beta_1^{\epsilon_2,\epsilon_2} = 1 \; , \; \beta_1^{\epsilon_1,\epsilon_2} = 0 \; .$$
 (22)

It should be pointed out that the algorithms proposed in [39, 40] can detect the number of loops in the VR complex at scales ϵ_1 and ϵ_2 by computing $\beta_1^{\epsilon_1,\epsilon_1}$ and $\beta_1^{\epsilon_2,\epsilon_2}$, respectively. However these Betti numbers do not hold any persistence information. The algorithms in [39, 40] cannot calculate the persistence Betti number $\beta_1^{\epsilon_1,\epsilon_2}$ which holds the persistence information (number of loops present at scale ϵ_1 that persist to scale ϵ_2). Thus, these algorithms cannot track topological features across different scales. Moreover, even though $\beta_1^{\epsilon_1,\epsilon_1} = \beta_1^{\epsilon_2,\epsilon_2}$, the persistence Betti number cannot be deduced from $\beta_1^{\epsilon_1,\epsilon_1}$ and $\beta_1^{\epsilon_2,\epsilon_2}$. This is because it indicates that the loops at scales across $\sqrt{2}$ are different, which is additional information to the existence of a loop encoded in the other two Betti numbers.

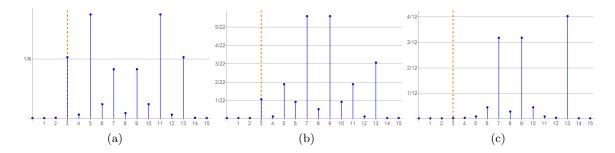


FIG. 3. Probability density corresponding to the persistent Dirac operator (a) $B_1^{\epsilon_1,\epsilon_1}$, (b) $B_1^{\epsilon_2,\epsilon_2}$, and (c) $B_1^{\epsilon_1,\epsilon_2}$, with $\xi=1,\,l=3,\,M=16$. The heights at p=3, multiplied by the dimension of the Hilbert space (8, 22, and 12, respectively), yield the Betti numbers $\beta_1^{\epsilon_1,\epsilon_1}=\beta_1^{\epsilon_2,\epsilon_2}=1$, and the persistent Betti number $\beta_1^{\epsilon_1,\epsilon_2}=0$.

Figure 4 shows the quantum circuit for the calculation of Betti number $\beta_1^{\epsilon_1,\epsilon_2}$. The Dirac operator $B_1^{\epsilon_1,\epsilon_2}$ acts on an 12-dimensional Hilbert space. For $\xi=1$ are $\lambda=-1,\pm\sqrt{5}$, each of degeneracy 4. Notice that 1 is not an eigenvalue, therefore $\beta_1^{\epsilon_1,\epsilon_2}=0$. This is confirmed by the quantum algorithm depicted in Figure 4 as explained below.

A register of 4 qubits initially in the state $|0000\rangle_1$ is brought into the state $\frac{1}{4}\sum_{x=0}^{15}|x\rangle_1$, by acting with the Hadamard matrix on each. Then we use them as control to apply CNOT on each of 4 qubits in an additional register in the state $|0000\rangle_2$, thus entangling them to the state $\frac{1}{4}\sum_{x=0}^{15}|x\rangle|x\rangle_2$. We introduce a third register of 4 qubits (choosing M=16) in the state $|0000\rangle_R$ and act on each with the Hadamard gate to bring them into the state $\frac{1}{4}\sum_{y=0}^{15}|y\rangle_R$. We then use them as control to act on register 2 with the exponential Dirac operator (see Appendix C for details). Finally, we measure all 4 qubits in the register R. The result is the probability distribution $\mathcal{P}(p)$, where $p=0,1,\ldots,15$ (Eq. (21)) depicted in Figure 3(c). With the choice $l=3,\ M=16$, a measurement of the register of 4 qubits yields no peak at p=3, showing that $\beta_1^{\epsilon_1,\epsilon_2}=0$.

The persistent Betti number $\beta_1^{\epsilon_1,\epsilon_1}$ is calculated using an eight-dimensional Hilbert space. The eigenvalues of the persistent Dirac operator for $\xi=1$ are $\pm 1, \pm \sqrt{3}, \pm \sqrt{5}$. Two of the eigenvalues $(\pm \sqrt{3})$ are degenerate with multiplicity 2. We are interested in the multiplicity of the eigenvalue $\lambda=\xi=1$, which is shown to be 1 by the peak at p=3 of height 1/8 (see Fig.3(a)). The closest eigenvalue to $\lambda=1$ is $\lambda=\sqrt{3}$ which is near p=5. Thus, the peak at the nearest eigenvalue is

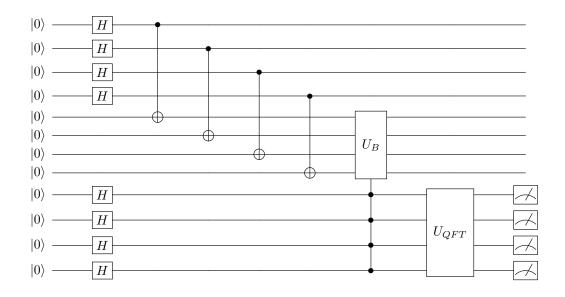


FIG. 4. Quantum circuit for the calculation of Betti number $\beta_1^{\epsilon_1,\epsilon_2}$.

well separated from the one of interest (in Figure 3(a), one can see a dip at p=4, and the height $\mathcal{P}(5)\approx\frac{2}{8}$, confirming the double degeneracy of the eigenvalue $\lambda=\sqrt{3}$). The quantum circuit flows as the one in Figure 4 but with registers 1 and 2 having 3 qubits each.

The calculation of Betti number $\beta_1^{\epsilon_2,\epsilon_2}$ proceeds similarly. The Hilbert space is 22-dimensional and for $\xi=1$, the eigenvalues are the same as for $\beta_1^{\epsilon_1,\epsilon_1}$, but with degeneracies 1, 3, 2, 7 for $\lambda=1,-1,\pm\sqrt{3},\pm\sqrt{5}$, respectively. The distance from the eigenvalue of interest $(\lambda=1)$ to its closest one $(\lambda=\sqrt{3})$ is same as before, therefore we can choose l=3, M=16, again. The quantum circuit is similar to the one in Figure 4 except that registers 1 and 2 need 5 qubits each. The resulting probability distribution is depicted in Figure 3(b) showing that $\beta_1^{\epsilon_2,\epsilon_2}=1$.

V. Conclusion

Our work established a quantum persistent homology algorithm that detects and computes the topological features of point cloud data as their resolution changes. Our method considered the persistent Laplacian operator, generalizing the Laplacian operator discussed in [39, 40]. We provided an implementation on the challenging problem of two particular point cloud squares proposed in [41] as an example in which persistent Betti numbers cannot be deduced from Betti numbers. Classical algorithms for persistent homology (e.g., see [51]) typically need $\mathcal{O}(2^{3n})$ operations to diagonalize a $2^n \times 2^n$ boundary matrix and obtain the topological information. Our quantum algorithm estimates the same information after a number of operations which is only polynomial in the number of data points n. Moreover, our algorithm can encode the full simplicial complex using n qubits, while a classical computer needs 2^n bits for the same process. Last, our algorithm needs a small number of measurements to obtain reliable results, and thus, its application to scientific data including signal processing or image analysis problems is advantageous.

Acknowledgments

Research supported by the National Science Foundation award DMS-2012609. G. Siopsis also acknowledges the Army Research Office award W911NF-19-1-0397, and the National Science Foundation award OMA-1937008.

- [1] Gunnar Carlsson. Topology and data. Bulletin of the American Mathematical Society, 46(2):255–308, 2009.
- [2] Erik Carlsson, Gunnar Carlsson, and Vin De Silva. An algebraic topological method for feature identification. *International Journal of Computational Geometry*, 16(4):291–314, 2006.
- [3] Herbert Edelsbrunner and John Harer. Computational topology: an introduction. American Mathematical Society, 2010.
- [4] Ramanarayan Vasudevan, Aaron Ames, and Ruzena Bajcsy. Persistent homology for automatic determination of human-data based cost of bipedal walking. *Nonlinear Analysis: Hybrid Systems*, 7(1):101–115, 2013.
- [5] Genki Kusano, Kenji Fukumizu, and Yasuaki Hiraoka. Persistence weighted Gaussian kernel for topological data analysis. arXiv:1601.01741, 2016.
- [6] Herbert Edelsbrunner. Persistent homology in image processing. In *International Workshop on Graph-Based Representations in Pattern Recognition*, pages 182–183. Springer, 2013.
- [7] Moo Chung, Jamie Hanson, Jieping Ye, Richard Davidson, and Seth Pollak. Persistent homology in sparse regression and its application to brain morphometry. *IEEE transactions on medical imaging*, 34(9):1928–1939, 2015.
- [8] Elizabeth Munch. Applications of persistent homology to time varying systems. PhD thesis, Duke University, 2013.
- [9] Firas A Khasawneh and Elizabeth Munch. Chatter detection in turning using persistent homology. *Mechanical Systems and Signal Processing*, 70:527–541, 2016.
- [10] Vinay Venkataraman, Karthikeyan Natesan Ramamurthy, and Pavan Turaga. Persistent homology of attractors for action recognition. arXiv:1603.05310, 2016.
- [11] Aaron Adcock, Erik Carlsson, and Gunnar Carlsson. The ring of algebraic functions on persistence bar codes. *Homology, Homotopy and Applications*, 18(1):381–402, 2016.
- [12] Saba Emrani, Thanos Gentimis, and Hamid Krim. Persistent homology of delay embeddings and its application to wheeze detection. *IEEE Signal Processing Letters*, 21(4):459–463, 2014.
- [13] Cássio MM Pereira and Rodrigo F de Mello. Persistent homology for time series and spatial data clustering. Expert Systems with Applications, 42(15):6026–6038, 2015.
- [14] Monica Nicolau, Arnold J Levine, and Gunnar Carlsson. Topology based data analysis identifies a subgroup of breast cancers with a unique mutational profile and excellent survival. *Proceedings of the National Academy of Sciences*, 108(17):7265–7270, 2011.
- [15] Joshua Mike and Vasileios Maroulas. Combinatorial hodge theory for equitable kidney paired donation. Foundations of Data Science, 1(1):87–101, 2019.
- [16] Joshua Mike, Colin D Sumrall, Vasileios Maroulas, and Fernando Schwartz. Nonlandmark classification in paleobiology: computational geometry as a tool for species discrimination. *Paleobiology*, pages 1–11, 2016.
- [17] Vasileios Maroulas, Farzana Nasrin, and Christopher Oballe. A bayesian framework for persistent homology. SIAM Journal on Mathematics of Data Science, 2(1):48–74, 2020.
- [18] J. Townsend, C.P. Micucci, J.H. Hymel, V. Maroulas, and K. D. Vogiatzis. Representation of molecular structures with persistent homology for machine learning applications in chemistry. *Nat Commun*, 11:3230, 2020.

- [19] Vasileios Maroulas, Cassie Putman Micucci, and Farzana Nasrin. Bayesian Topological Learning for Classifying the Structure of Biological Networks. *Bayesian Analysis*, pages 1 26, 2021.
- [20] Jose A Perea and John Harer. Sliding windows and persistence: An application of topological methods to signal analysis. *Foundations of Computational Mathematics*, 15(3):799–838, 2015.
- [21] David Rouse, Adam Watkins, David Porter, John Harer, Paul Bendich, Nate Strawn, Elizabeth Munch, Jonathan DeSena, Jesse Clarke, and Jeffrey Gilbert. Feature-aided multiple hypothesis tracking using topological and statistical behavior classifiers. In SPIE Defense+ Security, pages 94740L-94740L. International Society for Optics and Photonics, 2015.
- [22] Vin De Silva and Robert Ghrist. Coordinate-free coverage in sensor networks with controlled boundaries via homology. *The International Journal of Robotics Research*, 25(12):1205–1222, 2006.
- [23] Vin De Silva and Robert Ghrist. Homological sensor networks. *Notices of the American mathematical society*, 54(1), 2007.
- [24] Pawel Dlotko, Robert Ghrist, Mateusz Juda, and Marian Mrozek. Distributed computation of coverage in sensor networks by homological methods. Applicable Algebra in Engineering, Communication and Computing, 23(1-2):29–58, 2012.
- [25] Gunnar Carlsson and Vin De Silva. Zigzag persistence. Foundations of computational mathematics, 10(4):367–405, 2010.
- [26] Gunnar Carlsson, Vin De Silva, and Dmitriy Morozov. Zigzag persistent homology and real-valued functions. In *Proceedings of the twenty-fifth annual symposium on Computational geometry*, pages 247–256. ACM, 2009.
- [27] Vin De Silva and Robert Ghrist. Coverage in sensor networks via persistent homology. Algebraic & Geometric Topology, 7(1):339–358, 2007.
- [28] Jacopo Binchi, Emanuela Merelli, Matteo Rucco, Giovanni Petri, and Francesco Vaccarino. jholes: A tool for understanding biological complex networks via clique weight rank persistent homology. *Electronic Notes in Theoretical Computer Science*, 306:5–18, 2014.
- [29] Kelin Xia, Xin Feng, Yiying Tong, and Guo Wei Wei. Persistent homology for the quantitative prediction of fullerene stability. *Journal of computational chemistry*, 36(6):408–422, 2015.
- [30] Mijail Guillemard and Armin Iske. Signal filtering and persistent homology: an illustrative example. Proc. Sampling Theory and Applications (SampTA'11), 2011.
- [31] Paul Bendich, James Stephen Marron, Ezra Miller, Alex Pieloch, and Sean Skwerer. Persistent homology analysis of brain artery trees. *The Annals of Applied Statistics*, 10(1):198–218, 2016.
- [32] Andrew Marchese and Vasileios Maroulas. Topological learning for acoustic signal identification. In Information Fusion (FUSION), 2016 19th International Conference on, pages 1377–1381. ISIF, 2016.
- [33] A. Marchese and V. Maroulas. Signal classification with a point process distance on the space of persistence diagrams. Advances in Data Analysis and Classification, 12(3):657–682, 2018.
- [34] Lee M Seversky, Shelby Davis, and Matthew Berger. On time-series topological data analysis: New data and opportunities. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 59–67, 2016.
- [35] Jose A Perea, Anastasia Deckard, Steve B Haase, and John Harer. Sw1pers: Sliding windows and 1-persistence scoring; discovering periodicity in gene expression time series data. *BMC bioinformatics*, 16(1):257, 2015.
- [36] Dmitriy Morozov. Dionysus, a C++ library for computing persistent homology, 2007.
- [37] Ulrich Bauer. Ripser, 2015. https://github.com/Ripser/ripser.
- [38] Chao Chen and Michael Kerber. Persistent homology computation with a twist. In *Proceedings 27th European Workshop on Computational Geometry*, volume 11, 2011.
- [39] Seth Lloyd, Silvano Garnerone, and Paolo Zanardi. Quantum algorithms for topological and geometric analysis of data. *Nature communications*, 7(1):1–7, 2016.
- [40] George Siopsis. Quantum topological data analysis with continuous variables. Foundations of Data Science, 4(1):419–431, 2019.
- [41] Sam Gunn and Niels Kornerup. Review of a quantum algorithm for betti numbers. arXiv preprint arXiv:1906.07673, 2019.

- [42] Ryu Hayakawa. Quantum algorithm for persistent betti numbers and topological data analysis. arXiv preprint arXiv:2111.00433, 2021.
- [43] H. Huang and et al. Demonstration of topological data analysis on a quantum processor. Optica, 5:193, 2018.
- [44] R. Dridi and H. Alghassi. Homology computation of large point clouds using quantum annealing. arXiv:1512.09328, 2016.
- [45] Chu Ryang Wie. A quantum circuit to construct all maximal cliques using grover search algorithm. arXiv preprint arXiv:1711.06146, 2017.
- [46] Rui Wang, Duc Duy Nguyen, and Guo-Wei Wei. Persistent spectral graph. *International journal for numerical methods in biomedical engineering*, 36(9):e3376, 2020.
- [47] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. Quantum random access memory. *Phys. Rev. Lett.*, 100:160501, Apr 2008.
- [48] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. Architectures for a quantum random access memory. *Physical Review A*, 78(5):052310, 2008.
- [49] Gilles Brassard, Peter Hoyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. *Contemporary Mathematics*, 305:53–74, 2002.
- [50] Lov K. Grover. Quantum computers can search rapidly by using almost any transformation. *Phys. Rev. Lett.*, 80:4329–4332, May 1998.
- [51] Afra Zomorodian and Gunnar Carlsson. Computing persistent homology. Discrete & Computational Geometry, 33(2):249–274, 2005.
- [52] Patrick Rebentrost, Adrian Steffens, Iman Marvian, and Seth Lloyd. Quantum singular-value decomposition of nonsparse low-rank matrices. *Physical review A*, 97(1):012327, 2018.

A. Grover's algorithm

Here we review the salient features of amplitude amplification [49] and Grover's search algorithm [50] which are needed for the implementation of the projection P_k^{ϵ} (Eq. (12)), for completeness.

Let $|\Psi_k\rangle \in \mathcal{H}_k$ be a state in the span of the k-simplex states. We wish to construct the normalized projected state $|\Psi_k^{\epsilon}\rangle = \frac{P^{\epsilon}|\Psi_k\rangle}{\|P^{\epsilon}|\Psi_k\rangle\|} \in \mathcal{H}_k^{\epsilon}$, assuming that it exists. To this end, we introduce the unitary operator

$$U_G = -U_{\Psi_k} U^{\epsilon}$$
, with $U_{\Psi_k} = I - 2|\Psi_k\rangle\langle\Psi_k|$ and $U^{\epsilon} = I - 2P^{\epsilon}$ (A1)

Splitting $|\Psi_k\rangle$ as

$$|\Psi_k\rangle = \sin\theta |\Psi_k^{\epsilon}\rangle + \cos\theta |\bar{\Psi}_k^{\epsilon}\rangle , \quad \sin\theta = ||P^{\epsilon}|\Psi_k\rangle|| ,$$
 (A2)

where $P^{\epsilon}|\Psi_{k}^{\epsilon}\rangle = |\Psi_{k}^{\epsilon}\rangle$ and $P^{\epsilon}|\bar{\Psi}_{k}^{\epsilon}\rangle = 0$, we can think of $|\Psi_{k}\rangle$ as the vector $\begin{pmatrix} \sin\theta\\\cos\theta \end{pmatrix}$ in the two-dimensional space spanned by $\{\Psi_{k}^{\epsilon}, \bar{\Psi}_{k}^{\epsilon}\}$. Then U_{G} acts as a rotation by an angle 2θ and can be represented as $U_{G} = \begin{pmatrix} \cos 2\theta & \sin 2\theta\\ -\sin 2\theta & \cos 2\theta \end{pmatrix}$. Applying it K times, we obtain the state

$$U_G^K |\Psi_k\rangle = \begin{pmatrix} \sin(2K+1)\theta \\ \cos(2K+1)\theta \end{pmatrix}$$
 (A3)

This is close to the desired state for $(2K+1)\theta \approx \frac{\pi}{2}$. Therefore, the number of Grover steps needed is $K = \lfloor \frac{\pi}{4\theta} \rfloor$.

B. Implementation of an exponential operator

Here we review the construction of the exponential operator $e^{itB_k^{\epsilon,\epsilon'}}$ [52] which is needed for phase estimation (Eq. (C6)). We start by constructing the SWAP_B operator from the Dirac matrix $B_k^{\epsilon,\epsilon'}$ (Eq. (18)),

$$S \equiv \text{SWAP}_B = \sum_{x,y} B_k^{\epsilon,\epsilon'}(x,y)|y\rangle\langle x| \otimes |x\rangle\langle y| , \quad B_k^{\epsilon,\epsilon'}(x,y) = \langle x|B_k^{\epsilon,\epsilon'}|y\rangle , \quad (B1)$$

where we have used labels x,y to collectively represent $\{b,\sigma_{k+b}^b\}$, where $b\in\{0,\pm1\}$ and $\sigma_{k+b}^b\in S_{k+b}^{\epsilon_b}$, with $\epsilon_b=\epsilon$ for b=-1,0 and $\epsilon_b=\epsilon'$ for b=+1. With the choice $\xi=1$, all matrix elements $B_k^{\epsilon,\epsilon'}(x,y)\in\{0,\pm1\}$.

Then we construct the exponential SWAP_B operator $e^{i\Delta tS}$, which can be done efficiently because S is a one-sparse matrix.

Next, we act on the state $|s\rangle \otimes |\Psi_k^{\epsilon,\epsilon'}\rangle$, where $|s\rangle$ is the uniform state

$$|s\rangle \propto P^{\epsilon,\epsilon'} \sum_{x} |x\rangle$$
 (B2)

and $|\Psi_k^{\epsilon,\epsilon'}\rangle$ is an arbitrary state in the subspace on which $B_k^{\epsilon,\epsilon'}$ acts obeying $P^{\epsilon,\epsilon'}|\Psi_k^{\epsilon,\epsilon'}\rangle = |\Psi_k^{\epsilon,\epsilon'}\rangle$. They can be constructed using Grover's algorithm, as outlined in Appendix A. After tracing over the space in which $|s\rangle$ lives, we obtain

$$\operatorname{tr}_{1}\left[e^{-i\Delta t\mathcal{S}}|\boldsymbol{s}\rangle\langle\boldsymbol{s}|\otimes|\Psi_{k}^{\epsilon,\epsilon'}\rangle\langle\Psi_{k}^{\epsilon,\epsilon'}|e^{i\Delta t\mathcal{S}}\right] = e^{-iB_{k}^{\epsilon,\epsilon'}\Delta t/\boldsymbol{N}}|\Psi_{k}^{\epsilon,\epsilon'}\rangle\langle\Psi_{k}^{\epsilon,\epsilon'}|e^{iB_{k}^{\epsilon,\epsilon'}\Delta t/\boldsymbol{N}} + \mathcal{O}(\Delta t^{2})$$
(B3)

where N is the number of basis vectors comprising the state $|s\rangle$ in (B2). Then we apply Grover's algorithm to implement the projection $P^{\epsilon,\epsilon'}$ and obtain the state

$$e^{-iB_k^{\epsilon,\epsilon'}\Delta t/\mathbf{N}}|\Psi_k^{\epsilon,\epsilon'}\rangle$$
 (B4)

up to second order in Δt .

The desired state $e^{-itB_k^{\epsilon,\epsilon'}}|\Psi_k^{\epsilon,\epsilon'}\rangle$ for finite t can be obtained by repeating the above construction as many times as needed.

C. Phase estimation

Here we outline the steps of the phase estimation subroutine which plays a major role in our quantum algorithm leading to the probability distribution $\mathcal{P}(p)$, where $p = 0, 1, \ldots, M-1$ and M is an arbitrarily chosen integer (Eq. (21)).

We start with the state (B2) which can be expanded in the computational basis as

$$|s\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle$$
 (C1)

where x collectively represents $\{b, \sigma_{k+b}^b\}$, as detailed in Appendix B.

Next, we copy the basis states to create the maximally entangled state

$$|\tilde{s}\rangle_{12} = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle_1 |x\rangle_2 \tag{C2}$$

Let the eigenvalue problem of the persistent Dirac operator $B_k^{\epsilon,\epsilon'}$ (Eq. (18)) be

$$B_k^{\epsilon,\epsilon'}|\lambda\rangle = \lambda|\lambda\rangle$$
 (C3)

By considering components, using the completeness of the set of eigenvectors, and the fact that they are real, it is easy to see that

$$|\tilde{s}\rangle_{12} = \frac{1}{\sqrt{N}} \sum_{\lambda} |\lambda\rangle_1 |\lambda\rangle_2$$
 (C4)

Next, we introduce a register of qubits R in the state

$$|R\rangle_R = \frac{1}{\sqrt{M}} \sum_{y=0}^{M-1} |y\rangle_R \tag{C5}$$

and apply the operator

$$U = e^{2\pi i l y B_k^{\epsilon, \epsilon'}/M} \tag{C6}$$

entangling the registers 2 and R, where M and l are positive integers that can be adjusted at will. For details of the implementation of (C6), see Appendix B. We obtain

$$U|\tilde{s}\rangle_{12}|R\rangle_R = \frac{1}{\sqrt{NM}} \sum_{\lambda} \sum_{y=0}^{M-1} e^{2\pi i l y \lambda/M} |\lambda\rangle_1 |\lambda\rangle_2 |y\rangle_R \tag{C7}$$

Finally, we perform the quantum Fourier transform on the register R,

$$U_{QFT}|y\rangle_R = \frac{1}{\sqrt{M}} \sum_{p=0}^{M-1} e^{-2\pi i p y/M} |p\rangle_R$$
 (C8)

and the state of the system becomes

$$U_{QFT}U|\tilde{\boldsymbol{s}}\rangle_{12}|R\rangle_{R} = \frac{1}{M\sqrt{N}} \sum_{\lambda} \sum_{y=0}^{M-1} \sum_{p=0}^{M-1} e^{2\pi i(l\lambda - p)y/M} |\lambda\rangle_{1}|\lambda\rangle_{2}|p\rangle_{R}$$
 (C9)

Summing over y, we deduce

$$U_{QFT}U|\tilde{s}\rangle_{12}|R\rangle_{R} = \frac{1}{M\sqrt{N}} \sum_{\lambda} \sum_{p=0}^{M-1} \frac{e^{2\pi i l\lambda} - 1}{e^{2\pi i (l\lambda - p)/M} - 1} |\lambda\rangle_{1}|\lambda\rangle_{2}|p\rangle_{R}$$
 (C10)

The coefficients are strongly peaked at $p \approx l\lambda$. At the peaks, the coefficients are approximately equal to M. Therefore,

$$U_{QFT}U|\tilde{s}\rangle_{12}|R\rangle_R \approx \frac{1}{\sqrt{N}} \sum_{\lambda} |\lambda\rangle_1|\lambda\rangle_2|l\lambda\rangle_R$$
 (C11)

A measurement of the register R yields p with probability

$$\mathcal{P}(p) = \|_R \langle p | U_{QFT} U | \tilde{\boldsymbol{s}} \rangle_{12} | R \rangle_R \|^2 = \frac{1}{M^2 N} \sum_{\lambda} \left| \frac{e^{2\pi i l \lambda} - 1}{e^{2\pi i (l\lambda - p)/M} - 1} \right|^2$$
 (C12)

which completes the derivation of Eq. (21).

Approximately, the probability $\mathcal{P}(p)$ vanishes for all p, except at $p \approx l\lambda$. At the peaks, each eigenvalue contributes 1/N. Therefore, each peak is proportional to the multiplicity of the corresponding eigenvalue. In particular, for p = l, which corresponds to the eigenvalue of interest $\lambda = \xi = 1$, we have

$$\mathcal{P}(l) = \frac{\beta_k^{\epsilon, \epsilon'}}{N} + \frac{1}{M^2 N} \sum_{\lambda \neq 1} \frac{\sin^2 \pi l \lambda}{\sin^2 \frac{\pi l (\lambda - 1)}{M}}$$
(C13)

where $\beta_k^{\epsilon,\epsilon'}$ is the dimension of the kernel of the persistent Laplacian (kth persistent Betti number). In the limit $M \to \infty$, the sum over eigenvalues $\lambda \neq 1$ vanishes, and we obtain

$$\beta_k^{\epsilon,\epsilon'} = N\mathcal{P}(l) . \tag{C14}$$