A quantum algorithm for spectral clustering via Hodge Laplacians

Anonymous Authors[1]

Abstract

Recognizing higher-order interactions within complex networks is crucial for predicting their behavior. The identification of communities is particularly important to discover regions devoted to critical activities within the network. However, the computational cost of this task grows rapidly with the size of the interactions, surpassing the capabilities of current technologies. To overcome this challenge, we propose leveraging quantum computing to perform clustering over the simplices of these networks. We define a spectral clustering based on the bottom eigenvalues of the Hodge Laplacian, which is compact to represent and efficient to analyze on a quantum computer even for networks with large simplices. This guarantees a significant speedup compared to its classical analogs. Our work unveils a novel field of application for quantum machine learning, making quantum algorithms for topological data analysis directly applicable to the study of biological, societal, and technological networks.

1. Introduction

Complex networks allow the study of endless tasks and problems (Albert & Barabási, 2002; Newman, 2018; Battiston et al., 2021). Among the most important aspects to consider, the identification of densely connected communities is crucial as these can be associated with certain activities in the network. When studying this problem from the perspective of graph theory, it is denoted as *graph clustering* (Girvan & Newman, 2002; Schaeffer, 2007) and formulated as follows:

Problem 1.1 (Graph clustering)**.** *Given a graph $G = (V, E)$ and a natural number m, return a partition of the vertex set V, denoted as $\mathfrak{C} = \{C_1, \dots, C_m\}$, that maximizes the edge density within each cluster, $\max \sum_{i=1}^{m} |E(C_i)|$. (Another common measure is the modularity, Newman (2006).)*

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Due to the hardness of graph clustering, heuristics have been proposed. Among the most popular is the *spectral clustering* (Von Luxburg, 2007), which uses the eigensystem of the graph Laplacian $\mathbf{L}_0$. The algorithm works in two steps. Firstly, we have to define a spectral embedding, which is a mapping $\boldsymbol{\psi} : V \to \mathbb{R}^q$ from vertices to a low-dimensional Euclidean space. Each component $\boldsymbol{\psi}_i(v)$ measures how much the vertex v aligns with the i-bottom eigenvectors of $\mathbf{L}_0$. Secondly, we apply a Euclidean subspace clustering method to discover the clusters using, for example, k-means (MacQueen, 1967), independent component analysis (Comon, 1994), or SUBCLU (Kailing et al., 2004).

Despite its popularity in recent decades, this approach is significantly constrained by its inability to capture higher-order interactions within the nodes (Battiston et al., 2020; Hu et al., 2021). These higher-order interactions embody crucial information for various tasks: in genetics, such as epistasis detection (Zhou et al., 2022; Hoffmann et al., 2023), and protein interactions (Palukuri et al., 2023); in neuroscience (Giusti et al., 2016); in social networks, including disease spread (Iacopini et al., 2019; St-Onge et al., 2022), and social interactions (Alvarez-Rodriguez et al., 2021); as well as technological networks like transportation (flow) networks (Roddenberry & Segarra, 2019; Barbarossa & Sardellitti, 2020) and trade networks (Fagiolo et al., 2013).

Thus, we need a more general model than graphs: the abstract simplicial complexes (ASCs). It is described by a collection Σ of subsets of V closed under inclusion, with $\Sigma_k \subset \Sigma$ being the set of k-simplices (subsets of $k+1$ vertices). These entities have been extensively studied in algebraic topology (Hatcher, 2002) and combinatorics (Duval & Reiner, 2002), play a crucial role in generalizing graph clustering to higher-order systems, and lead to the definition of simplicial clustering:

Problem 1.2 (Simplicial clustering)**.** *Given an ASC (V, Σ) and two natural numbers m, k, return a partition of the k-simplices $\sigma \in \Sigma$, denoted as $\mathfrak{S} = \{S_1, \dots, S_m\}$, that maximizes the density of k-simplices within each cluster, $\max \sum_{i=1}^{m} |\Sigma_k(S_i)|$.*

We can use the Hodge Laplacian $\mathbf{L}_k$ (Lim, 2020) to capture higher-order relationships, much like $\mathbf{L}_0$ captures pairwise ones. This allows us to generalize of the spectral approach from graphs to ASCs, as shown by Ebli & Spreemann.

From a computational perspective, the operator $\mathbf{L}_k$ has a dimensionality of up to $\binom{n}{k+1} \times \binom{n}{k+1}$, posing a challenge in constructing it for large networks and moderately large interactions.

To overcome this challenge, an intriguing road is to explore the possibilities of quantum computing. Over the last decade, several algorithms have been proposed to extract specific topological information from network data. Examples include the extraction of Betti numbers (Hatcher, 2002) using the Lloyd-Garnerone-Zanardi algorithm (Lloyd et al., 2016) and spectral entropy (De Domenico & Biamonte, 2016) using the approach introduced in Gyurik et al. (2022). State-of-the-art analyses indicate that these algorithms can provide high-polynomial speedups (Schmidhuber & Lloyd, 2023) and even super-polynomial speedups in estimating certain topological properties and for certain families of networks (Gyurik et al., 2022). These algorithms exploit the capabilities of quantum computers to express $\exp(-it\mathbf{L}_k)$ efficiently despite being very large, and the ability of quantum phase estimation algorithm (Manenti & Motta, 2023) to eigendecompose this operator.

In this paper, we investigate the use of quantum computing and quantum phase estimation to address the simplicial clustering problem. We demonstrate the advantages in terms of speedup and discuss its implications in the realm of complex networks.

1.1. Contributions

Our first contribution is the generalization of the spectral embedding $\boldsymbol{\psi} : \Sigma_k \mapsto \mathbb{R}^q$ defined by Ebli & Spreemann. The original definition relies solely on the eigenvectors in the kernel of $\mathbf{L}_k$, which are interpreted in algebraic topology as multidimensional holes or loops. However, the approach assumes relatively small Betti numbers $\beta_k = q \leq 10$ to ensure a low-dimensional Euclidean space $\mathbb{R}^q$, and is needed to keep the subsequent Euclidean clustering phase efficient. This condition can be limiting for large graphs.

We propose an extension to the spectral embedding by considering not only the eigenvectors in the kernel of $\mathbf{L}_k$ but also those with small-but-nonzero associated eigenvalues. We demonstrate that incorporating this information effectively enhances the clustering method's power, similar to the spectral embedding in graphs. This observation becomes crucial in implementing quantum algorithms, where distinguishing an eigenvector with exactly zero eigenvalues from one with a close-to-zero eigenvalue may be challenging. The resulting increase in dimensionality q can be mitigated by introducing a random projection. Theoretical evidence, such as the Johnson-Lindenstrauss lemma (1984), suggests that the random projection will not negatively affect the results.

Our second contribution is the construction of a quantum algorithm to perform the spectral clustering, by calculating the spectral embedding on a quantum computer and then clustering the resulting Euclidean points classically by using k-means. Our algorithm revolves around the following statement:

Proposition 1.3. *Let (V, Σ) be an ASC, $|V| = n$, $0 < \tau \ll 1$ and $k \in \mathbb{N}^+$. Suppose we can efficiently sample the k-simplices of the ASC. Then, it exists a quantum channel Φ_{HS} that uniformly samples the bottom eigenvectors of $\mathbf{L}_k$, those with corresponding eigenvalues $< \tau/\|\mathbf{L}_k\|_2$, in time $\tilde{O}((\zeta_k^{-1/2}kn + n^2)\xi_k^{-1})$. Here, ζ_k is the density of k-simplices, $\zeta_k = |\Sigma_k|/\binom{n}{k+1}$, and ξ_k is the fraction of eigenvalues of $\mathbf{L}_k$ lower than $\tau/\|\mathbf{L}_k\|_2$.*

This means we can create a quantum operation Φ_{HS} allowing us to access efficiently the subset of eigenvectors needed for a spectral clustering. Thanks to the recent theory of random projection with random quantum unitaries (Kumaran et al., 2024), we can reduce these vectors to an arbitrary number of randomly projected vectors $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_q$; these are the ones used for constructing the spectral embedding $\boldsymbol{\psi}$ on the quantum computer. The random projected vectors are built by the means of a random unitary, U_{RND}, which is an approximate 2-design and whose quantum circuit has quadratic depth. By estimating the inner product between any k-simplex σ and $\boldsymbol{w}_i$, $i = 1, \ldots, q$, which can be done via the Hadamard test, we can construct the spectral embedding $\boldsymbol{\psi}(\sigma)$ on a quantum computer. Then, k-means is used (on a classical device) to finish the clustering.

The asymptotic performance comparison between the classical and quantum procedures is outlined in Table 1 and further detailed later in Section 4.2.

1.2. Related works

The concept of spectral clustering on simplicial complexes was first introduced empirically by Ebli & Spreemann (2019) and later theoretically justified in the work by Chen & Meila (2021). Grande & Schaub (2023a) examined these spectral clustering techniques by breaking it down in har-

Table 1. Stepwise performance comparison of the state-of-the-art classical algorithms and this approach for constructing a spectral embedding for k-simplices on n vertices. Here, ζ_k is the density of k-simplices and ξ_k is the fraction of low eigenvalues. Logarithmic factors are ignored by $\tilde{O}$.

Step	Classical	Quantum
Simplices sampling	$\tilde{O}\left(\zeta_k^{-1}\right)$	$\tilde{O}\left(\zeta_k^{-1/2}kn\right)$
Eigenvectors sampling	$\tilde{O}\left(\binom{n}{k+1}^3\right)$	$\tilde{O}\left((\zeta_k^{-1/2}kn + n^2)\xi_k^{-1}\right)$
Random projection	$\tilde{O}\left(\binom{n}{k+1}^2\right)$	$\tilde{O}\left(n^2\right)$

monics, curl, and gradient eigenmodes separately to highlight their distinct characteristics. In contrast, some properties of the graph Laplacian have been directly extended to higher orders: Krishnagopal & Bianconi (2021) showed how to detect higher-order connected components in the ASC, while Saito et al. (2024) defined a higher-order Fiedler's vector, allowing to partition the ASC into two loosely connected regions. Grande & Schaub (2023b) focused specifically on point clouds. Topological clustering has not been exclusively explored through spectral techniques: Serrano & Gómez (2020) relied on centrality measures, Reddy et al. (2023) used higher-order Cheeger inequality, and Schaub et al. (2020) employed random walks.

Turning to quantum algorithms for topological data analysis, Lloyd et al. (2016) proposed the first algorithm to estimate Betti numbers, Hayakawa (2022) modified the approach for persistent Betti numbers, while Ubaru et al. (2021) enhanced the original approach with a faster stochastic quadrature for rank estimation. Berry et al. (2023) further improved overall complexity with the use of Dicke states, Hamiltonian simulation by Quantum Signal Processing, and Kaiser windows in quantum phase estimation. Gyurik et al. (2022) showed that the problem of estimating the spectral entropy, a quantity related to the Betti numbers, is DCQ1-complete. They identified a family of graphs with Laplacian's spectral gap inversely polynomial in n, where quantum techniques for Betti number estimation have a superpolynomial advantage over classical techniques. Conversely, Apers et al. (2023) showed that it is efficient to estimate classically the Betti numbers for Laplacians with a constant spectral gap. Beyond Betti number estimation, Leditto et al. (2023) contextualized a possible quantum advantage in topological signal processing. A quantum walk-based approach for sampling dense communities was used in the work by Song (2024).

Finally, some quantum algorithms for graph clustering (not ASCs) have been proposed: Kerenidis & Landman (2021); Li et al. (2022) focused on speeding up the k-means portion of the algorithm, later extended to weighted graphs (Liu et al., 2023). Cade et al. (2023) studied the problem from the perspective of connectivity patterns (*motifs*).

1.3. Broader impact

This work focuses on analyzing complex networks using a quantum computer, and as far as we can discern, we do not anticipate any potential negative societal impact. Conversely, it has the potential to positively influence researchers interested in complex networks, such as those in genetics and neuroscience. Finally, our approach aims to promote the use of quantum machine learning in real-world tasks, as the field has garnered significant interest but still lacks applications (Schuld & Killoran, 2022).

2. Preliminaries

We introduce the background and notation used throughout the paper. For an introduction to quantum computing, the reader can refer to Manenti & Motta (2023) or Appendix A.

2.1. Hodge theory and algebraic topology

An *undirected graph* G is a set of vertices $V = \{1, \dots, n\}$ connected via edges in the set $E \subseteq \binom{V}{2}$, i.e. subsets of two distinct elements in V. This structure can be described by means of an adjacency matrix $A \in \mathbb{R}^{n\times n}$, where $[A]_{i,j} = 1$ if and only if the node i and j are connected by an edge, and zero otherwise. The degree matrix is $D = \mathrm{diag}(d_1, \dots, d_n)$, with d_i being the number of edges acting on the i-th vertex.

While a graph can capture only pairwise interactions, the *abstract simplicial complex* (ASC) can capture relationships of arbitrary order. An ASC (V, Σ) is a finite, non-empty set of vertices $V = \{1, \dots, n\}$ and a collection Σ of subsets of V closed under inclusion. A *k-simplex* is an ordered set $\sigma \in \Sigma$ in the form

$$\sigma = [v_0, \dots, v_k]. \tag{1}$$

The set of k-simplices in Σ is denoted by Σ_k. Each k-simplex is assigned an *orientation* $\Omega(\sigma) \in \{\pm 1\}$. The orientation is positive for $v_0 < \dots < v_k$; while if we permute the positive order of vertices according to π, the orientation is determined by $\Omega(\pi(\sigma)) = \mathrm{sign}(\pi)$.

The *k-chain space* of Σ is the real vector space over k-simplices, $C_k(\Sigma; \mathbb{R}) = \mathrm{span}\, \Sigma_k$. Given a global ordering of vertices, we can identify the standard orthonormal basis $\{\sigma_i\}$ chosen to be lexicographically ordered.[1] As C_k is isomorphic to $\mathbb{R}^{|\Sigma_k|}$, the standard inner product on $\mathbb{R}^{|\Sigma_k|}$ induces an inner product on C_k. The *k-boundary map* $\partial_k : C_k \to C_{k-1}$ maps a k-simplex to the alternating sum of its faces,

$$\partial_k \sigma = \sum_{v_j \in \sigma} (-1)^j (\sigma \setminus \{v_j\}). \tag{2}$$

Note that $\mathrm{im}(\partial_{k+1}) \subseteq \ker(\partial_k)$ or, equivalently, $\partial_{k-1} \circ \partial_k = 0$. A *$k$-cycle* is a k-chain c such that $c \in \ker(\partial_k)$. A *k-boundary* is a k-cycle such that $c \in \mathrm{im}(\partial_{k+1})$. The space of independent k-cycles that are *not* k-boundaries, $H_k = \ker(\partial_k)/\,\mathrm{im}(\partial_{k+1})$, is called the *$k$-homology group*, with each equivalence class corresponding to a *k-hole* in Σ. Its dimension is called the *k-Betti number*, $\beta_k = \dim H_k$.

The dual space of C_k, i.e., the space of homomorphisms over k-simplices is the *k-cochain space* and denoted as C^k. To accommodate the orientation of the simplices, the functions $f \in C^k$ must be alternating. The dual of ∂_{k+1} is

[1] The lexicographic order between k-simplices can be defined as $\tau \prec \sigma \iff \bigvee_{i=1}^{k} [(\bigwedge_{j=1}^{i-1} \tau_j = \sigma_j) \wedge (\tau_i < \sigma_i)]$.

the *k-coboundary map* δ_k, $\delta_k = \partial_{k+1}^\top$, which leads to the definition of cocycle, coboundary and cohomology space $H^k = \ker(\delta_k)/\operatorname{im}(\delta_{k-1})$.

The graph Laplacian of G describes the diffusion over a network and can be equivalently described in terms of the adjacency A and degree matrices D, or in terms of the incidence matrix $\mathbf{B}_1 \in \mathbb{R}^{n\times|E|}$, the latter being the matrix representation of the 1-boundary operator ∂_1,

$$\mathbf{L}_0 = A - D = \mathbf{B}_1\mathbf{B}_1^\top. \tag{3}$$

By construction, $\mathbf{L}_0$ is positive semi-definite (and thus Hermitian), with non-negative eigenvalues only.

Similarly, the k-Hodge Laplacian generalizes this concept, representing the diffusion over higher-order simplices. It is defined in terms of $\mathbf{B}_k$, the matrix representation of ∂_k with respect to the standard orthonormal basis $\{\sigma_i\}$ of C_k,

$$[\mathbf{B}_k]_{\tau,\sigma} = \begin{cases} (-1)^\ell, & \tau = \sigma \setminus \{v_\ell\} \\ 0, & \text{otherwise} \end{cases}. \tag{4}$$

The k-Hodge Laplacian is defined as:

$$\mathbf{L}_k = \mathbf{B}_k^\top\mathbf{B}_k + \mathbf{B}_{k+1}\mathbf{B}_{k+1}^\top. \tag{5}$$

The dimension of $\ker\mathbf{L}_k$ corresponds to the Betti number β_k. Due to the duality between boundary and coboundary map, $\mathbf{L}_k$ can be interpreted as both an operator on C_k and C^k. According to Hodge's decomposition,

$$C^k = \operatorname{im}(\mathbf{B}_k^\top) \oplus \ker(\mathbf{L}_k) \oplus \operatorname{im}(\mathbf{B}_{k+1}), \tag{6}$$

we can decompose a k-cochain into its solenoidal ($\operatorname{im}(\mathbf{B}_k^\top)$), irrotational ($\operatorname{im}(\mathbf{B}_{k+1})$) and *harmonic* ($\ker(\mathbf{L}_k)$) parts.

2.2. Group theory

Let $\mathcal{U}(N)$ be the group of $N \times N$ complex unitary matrices. A *Haar measure* for the unitary group $\mathcal{U}(N)$ is its unique left- and right-invariant probability measure μ. A *Haar-random unitary* is sampled from μ, and captures the idea of sampling 'uniformly' from the unitary group.

A *unitary k-design* is a probability distribution ν over a set of unitaries $S \subseteq \mathcal{U}(N)$ such that:

$$\mathop{\mathbb{E}}_{V\sim\nu}[V^{\otimes k}O(V^\dagger)^{\otimes k}] = \mathop{\mathbb{E}}_{U\sim\mu}[U^{\otimes k}O(U^\dagger)^{\otimes k}] \tag{7}$$

for all linear operators O, meaning the distribution ν can mime the behavior of the Haar-random distribution up to the k-th statistical moment (Sim et al., 2019).

3. Spectral clustering

For graphs and pairwise interactions, the spectral graph clustering is performed as follows (Von Luxburg, 2007). Consider the graph $G = (V, E)$ and integer m (number of clusters):

1. Compute the m bottom eigenvectors $\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_m$ of the graph Laplacian $\mathbf{L}_0$.
2. To each vertex $v \in C_0$ associate the point $\boldsymbol{\psi}(v) \in \mathbb{R}^q$, where $(\boldsymbol{\psi}(v))_j = \langle v, \boldsymbol{\varphi}_j\rangle_{C_0}$.[2]
3. Calculate k-means for m clusters on the set of points.

When generalizing the approach from graphs to ASCs in the following way (Ebli & Spreemann, 2019). Consider the ASC (V, Σ), natural numbers k (size of the interaction) and m (clusters):

1. Compute the null space eigenvectors of $\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_{\beta_k}$ of the Hodge Laplacian $\mathbf{L}_k$. Check that the Betti number β_k is relatively low (≤ 10).
2. To each k-simplex $\sigma \in C_k$ associate the point $\boldsymbol{\psi}(\sigma) \in \mathbb{R}^{\beta_k}$, where $(\boldsymbol{\psi}(\sigma))_j = \langle \sigma, \boldsymbol{\varphi}_j\rangle_{C_k}$.
3. Calculate k-means for m clusters on the set of points.

Note that the former (graphs) approach employs near-harmonics, while the latter (ASCs) relies solely on harmonics. It is crucial in the subsequent section to determine whether the presence of near-harmonics can be advantageous or detrimental to the clustering algorithm. For this purpose, we have some theoretical evidence supporting the utility of near-harmonics.

Firstly, from a signal processing perspective (Barbarossa & Sardellitti, 2020; Schaub et al., 2021), we can treat the eigenvectors of the Hodge Laplacian as the Fourier basis on cochains, analogous to the continuous case. In this scenario, a harmonic and its eigenvalue represent a component and its frequency (or mode). Clearly, harmonic and near-harmonics correspond to constant and low-frequency components and behave similarly (slowly varying), while high eigenvalues-eigenvectors correspond to high-frequency components (rapidly varying). Secondly, the Fiedler vector, used to bipartite the vertices of a connected graph into two internally well-connected regions but loosely connected to each other, can be generalized to bipartite an ASCs according to its k-simplices, as shown in Saito et al. (2024). Thirdly, the Cheeger inequalities, used to connect the smallest nonzero eigenvalues to the sparsest cut of a graph, admit a generalization to Hodge Laplacians as well (Gundert & Szedlák, 2014).

Although the use of near-harmonics appears beneficial, they will result in embedding the simplices in a high-dimensional space. To bring it back to a more manageable dimension

[2] We recall that v is an element of the standard basis of C_0, $\boldsymbol{\varphi}_j$ are eigenvectors of an operator acting on C_k and $\langle\cdot,\cdot\rangle_{C_0}$ is the inner product induced by the isomorphism between C_0 and $\mathbb{R}^n$.

for clustering, we can employ random projection (Ghojogh et al., 2021). According to the Johnson-Lindenstrauss lemma, there always exists a linear dimensionality reduction transformation that preserves the distances between points.

Theorem 3.1 (Johnson & Lindenstrauss 1984)**.** *Let $0 < \varepsilon < 1$, positive integers N and $k \in O(\log N/\varepsilon^2)$. Then, for any set of points $\boldsymbol{p}_1, ..., \boldsymbol{p}_N \in \mathbb{R}^d$ there exists a linear transformation $\boldsymbol{\xi} : \mathbb{R}^d \to \mathbb{R}^k$ such that, for all pair of points $\boldsymbol{p}, \boldsymbol{q}$ we have*

$$(1-\varepsilon)\|\boldsymbol{p}-\boldsymbol{q}\|_2 \leq \|\boldsymbol{\xi}(\boldsymbol{p})-\boldsymbol{\xi}(\boldsymbol{q})\|_2 \leq (1+\varepsilon)\|\boldsymbol{p}-\boldsymbol{q}\|_2. \quad (8)$$

Then, a popular linear dimensionality reduction transformation $\boldsymbol{\xi} : \mathbb{R}^N \to \mathbb{R}^k$, with $k \in O(\varepsilon^{-2} \log N)$, is given by

$$\boldsymbol{\xi} = \Pi U, \quad (9)$$

where $U \in \mathbb{R}^{N\times N}$ is a Haar-random unitary matrix, and $\Pi \in \mathbb{R}^{k\times N}$ is a projection matrix selecting k out of N rows.

Our approach for spectral simplicial clustering is then determined as follows. Consider the ASC (V, Σ), naturals k (size of interaction), m (clusters), q (target dimensionality of the Euclidean space), and a real $\tau > 0$ (eigenvalue threshold). Then,

1. Compute the q'-bottom harmonic and near-harmonics of the Hodge Laplacian $\mathbf{L}_k$, $\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_{q'}$, where q' is the number of eigenvalues $< \tau$.
2. Pick a single Haar random unitary matrix $U \in \mathbb{R}^{N\times N}$, where $N = |\Sigma_k|$.
3. Pick q projection matrices $\Pi_1, \ldots, \Pi_q \in \mathbb{R}^{q\times N}$.
4. Compute the q projected vectors $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_q \in \mathbb{R}^q$ where $\boldsymbol{w}_i = \Pi_i U \sum_{j=1}^{q'} \boldsymbol{\varphi}_j$.
5. To each k-simplex $\sigma \in C^k$, associate the point $\boldsymbol{\psi}(\sigma) \in \mathbb{R}^q$, where $(\psi(\sigma))_j = \langle \sigma, \boldsymbol{w}_j \rangle_{C^k}$.
6. Calculate k-means for m clusters on the set of points.

Finally, it is worth noting that Grande & Schaub have previously explored the use of low-eigenvalues eigenvectors, and in particular have distinguished between curl and gradient contributions. Our approach differs by emphasizing the need for random projections.

4. Quantum spectral clustering

We now show the implementation of our approach to spectral clustering on a quantum computer. Firstly, we provide an overview of the algorithm for estimating the Betti numbers, introduced in Lloyd et al. (2016). Secondly, we present an overview of the quantum algorithm for the spectral embedding of k-simplices. Finally, we give an analysis of our approach.

4.1. Overview of Lloyd et al. (2016)

Let (V, Σ) be an ASC with n vertices. Any k-simplex σ is encoded as an n-bit binary string with a Hamming weight of $k+1$. This encoding partitions the Hilbert space $\mathcal{H}$ into

$$\mathcal{H} = \mathcal{H}_0 \oplus \mathcal{H}_1 \oplus \ldots \oplus \mathcal{H}_n \quad (10)$$

where k-simplices $|\sigma\rangle$ belong to $\mathcal{H}_{k+1}$. The subspace of $\mathcal{H}_{k+1}$ of the k-simplices of Σ is denoted as $\mathcal{H}^{\Sigma}_{k+1}$.

Note that $\mathbf{L}_k$ acts only on $C_k(\Sigma)$, here encoded as $\mathcal{H}^{\Sigma}_{k+1}$. The core idea of the algorithm exploits the quantum phase estimation algorithm (Manenti & Motta, 2023) to force the decomposition of a quantum state according to the eigenvalues and eigenvectors of a Hermitian operator. We start with the (normalized) classical mixture of states spanning $\mathcal{H}^{\Sigma}_{k+1}$,

$$\rho_k = \frac{\mathbb{I}_{\mathcal{H}^{\Sigma}_{k+1}}}{|\mathcal{H}^{\Sigma}_{k+1}|} = \frac{1}{|\mathcal{H}^{\Sigma}_{k+1}|} \sum_{\sigma\in\Sigma_k} |\sigma\rangle\langle\sigma| . \quad (11)$$

Then, we use an auxiliary register of p qubits (p chosen arbitrarily), to apply the quantum phase estimation of the operator $\exp(-i/\|\mathbf{L}_k\|_2 \mathbf{L}_k)$, obtaining

$$|\boldsymbol{\varphi}\rangle\langle\boldsymbol{\varphi}| = \mathrm{QPE}\left((|0\rangle\langle 0|)^{\otimes p} \otimes \rho_k\right) \mathrm{QPE}^{\dagger} \quad (12)$$

$$= \frac{1}{|\mathcal{H}^{\Sigma}_{k+1}|} \sum_{i=1}^{|\mathcal{H}^{\Sigma}_{k+1}|} |\tilde{\lambda}_i\rangle |\boldsymbol{\varphi}_i\rangle \langle\tilde{\lambda}_i| \langle\boldsymbol{\varphi}_i| \quad (13)$$

$$= \sum_{\eta=0}^{2^p-1} \alpha_\eta |\lambda_\eta\rangle |\boldsymbol{\varphi}_\eta\rangle \langle\lambda_\eta| \langle\boldsymbol{\varphi}_\eta| . \quad (14)$$

Equation (13) illustrates that when the QPE is fed with the classical mixture of states spanning C_k, it produces a classical mixture of pairs $(\tilde{\lambda}_i, \boldsymbol{\varphi}_i)$ consisting of (p-bit approximations) eigenvalues and their corresponding eigenvectors. This p-bit approximation leads to the discretization of the eigenvalues λ_i, placing them within intervals $[\eta 2^{-p}, (\eta+1)2^{-p})$. This allows us to factorize, leading to Equation (14), where $|\lambda_\eta\rangle$ represents one of the possible discretized eigenvalues, and $|\boldsymbol{\varphi}_\eta\rangle$ is the sum of eigenvectors associated with that particular eigenvalue.

Lloyd et al. (2016) uses Equation (14) to estimate α_0, which is the fraction of eigenvalues in the range $[0, 2^{-p})$. If the quantity 2^{-p} is lower than the spectral gap of $\mathbf{L}_k$, the estimation of α_0 is called the *normalized Betti numbers*, and corresponds to $\tilde{\beta}_k = \beta_k/|\Sigma_k|$. There are no known bounds on the spectral gap of $\mathbf{L}_k$ in general; however, for certain families of graphs, this is known to be inversely polynomial in n (Gyurik et al., 2022).

4.2. Overview of the quantum spectral embedding

Consider we need to calculate the spectral embedding $\boldsymbol{\psi}(\sigma) \in \mathbb{R}^q$ for a k-simplex σ given as input. Let $0 < \tau \ll 1$ be the eigenvalue threshold.

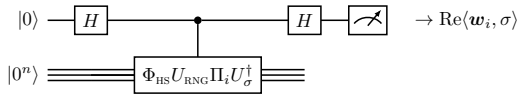


Figure 1. High-level description of the quantum circuit used to estimate the inner product between the i-th projected component $\boldsymbol{w}_i$ and the input simplex σ.

In contrast with the quantum algorithm for Betti number estimation, for the spectral embedding we will exploit the eigenvectors in Equation (14), and not the eigenvalues. Let $p = \lceil \log_2(1/\tau) \rceil$. We define the quantum channel Φ_{HS} as follows,

$$\begin{aligned}\Phi_{\mathrm{HS}}(|0\rangle\langle 0|)^{\otimes n} &= \left(\langle 0|^{\otimes p} \otimes \mathbb{I}^{\otimes n}\right) |\boldsymbol{\varphi}\rangle\langle\boldsymbol{\varphi}| \left(|0\rangle^{\otimes p} \otimes \mathbb{I}^{\otimes n}\right) \\ &= |\boldsymbol{\varphi}_0\rangle\langle\boldsymbol{\varphi}_0| \,. \end{aligned} \tag{15}$$

This is obtained by first loading ρ_k, pairing the state with an auxiliary register $(|0\rangle\langle 0|)^{\otimes p}$, then applying the QPE as in Equation (14), finally performing a post-selection on the eigenvalue register of the binary value 0^p, resulting in a state that is the superposition of eigenvectors having eigenvalue $[0, 2^{-p})$. The post-selection makes the operation non-unitary; therefore, we have to describe it as a quantum channel. The post-selection succeeds with probability $\xi_k = \alpha_0$, which needs to be $\mathrm{poly}(1/n)$.

Now, to obtain the random projections, we implement U_{RND}, an approximate unitary 2-design, which approximates the behavior of a Haar-random unitary up to the second order moment. As shown in Kumaran et al. (2024), this is enough to obtain random projected vectors that are well-behaved, meaning they satisfy the Johnson-Lindenstrauss lemma. The use of a unitary 2-design instead of Haar-random unitaries is advantageous from the computational point of view.

To define the q components of the spectral embedding $\boldsymbol{\psi}$, we need to define the projectors Π_i, $i = 1, \ldots, q$. For simplicity, we set $q = 2^m$. In this case, each projector is defined to be the sum of $N = 2^n/2^m$ states of the computational basis,

$$\Pi_i = \sum_{j=1}^{N} |iN + j\rangle\langle iN + j| \,. \tag{16}$$

Finally, the i-th random projected vector is

$$|\boldsymbol{w}_i\rangle = \Pi_i \cdot U_{\mathrm{RND}} \cdot |\boldsymbol{\varphi}_0\rangle \,. \tag{17}$$

Loading the k-simplex σ is trivial, and requires defining a constant depth quantum circuit $U_\sigma |0\rangle^{\otimes n} = |\sigma\rangle$ by placing a X gate on the i-th qubit of the ASC register (n qubits in total) if the i-th vertex belongs to σ.

Finally, the inner product between $\boldsymbol{w}_i$ and σ can be calculated via the Hadamard test, and uses one single ancillary qubit,

$$(H \otimes \mathbb{I}^{\otimes n}) \cdot \text{C-}(U_{\boldsymbol{w}_i} U_\sigma^\dagger) \cdot (H \otimes \mathbb{I}^{\otimes n}) \, |0\rangle_1 \, |0 \ldots 0\rangle_n \tag{18}$$

for which the expectation value $\langle \mathrm{Z} \rangle$ calculated on the ancillary qubit corresponds to $\mathrm{Re}(\langle \boldsymbol{w}_i, \sigma \rangle)$. There is no need to estimate the imaginary part. The process is depicted in Figure 1.

The detailed construction of these quantum circuits is shown in Appendix B.

4.3. Analysis of the quantum spectral clustering

Let Σ be the ASC and $0 < \tau \ll 1$ the eigenvalue threshold. $\zeta_k = |\Sigma_k| / \binom{n}{k+1}$ the density of k-simplices in Σ, and ξ_k the fraction of eigenvalues of $\mathbf{L}_k / \|\mathbf{L}_k\|_2$ lower than τ. Then, the algorithm works as follows:

1. Sample an arbitrary amount of k-simplices of the ASC Σ on the quantum computer.
2. For each k-simplex σ, estimate $\boldsymbol{\psi}(\sigma)$ on a quantum computer.
3. Calculate k-means on the set of points $\{\boldsymbol{\psi}(\sigma)\}$ for m clusters on a classical computer.

The sampling of k-simplices on a classical computer has complexity $\tilde{O}(\zeta_k^{-1})$ obtained by randomly generating a set of $k+1$ distinct vertices, checking if the set is a k-simplex (cost of the checking assumed to be negligible), as the procedure succeeds with probability ζ_k. $\tilde{O}$ ignores logarithmic factors. In contrast, a quantum search based on Grover's algorithm performs $O(\zeta_k^{-1/2})$ repetitions of a walk operator having cost $\tilde{O}(1) + \tilde{O}(kn)$, for the oracle and diffusion operator built with Dicke states, respectively (construction detailed in Appendix B); this leads to an overall cost of $\tilde{O}(\zeta_k^{-1/2} kn)$.

The spectral clustering on a classical computer is doable only if the Laplacian $\mathbf{L}_k$ is very sparse. If that's the case, as $\mathbf{L}_k$ is symmetric, we could rely on Lanczos's or Davidson's algorithms for extremal eigenvalues (Golub & Van Loan, 2013), and there is no benefit using the quantum approach. In contrast, for denser[3] $\mathbf{L}_k$, classically we need to rely on direct diagonalization $\tilde{O}(\binom{n}{k+1}^3)$ or divide-and-conquer approaches. The random projection has a cost as the vector-matrix multiplication, if we consider constant and negligible the number of projections q, the cost is $\tilde{O}(\binom{n}{k+1}^2)$.

In contrast, the quantum algorithm has a cost $\tilde{O}(\zeta_k^{-1/2} kn)$ for the state preparation, $\tilde{O}(n^2)$ for the Hamiltonian simulation of $\mathbf{L}_k$, and ξ_k^{-1} the factor due to the postselection on the eigenvalue register. This has an overall cost

[3] It can have up to $n2^n$ nonzero coefficients, n per row, as the operator is n-sparse.

of $\tilde{O}((\zeta_k^{-1/2}kn + n^2)\xi_k^{-1})$. The unitary 2-design can be implemented in $O(n^2)$, and the cost of the projection is considered negligible.

The cost of k-means is the same for both classical and quantum procedures.

5. Numerical example

We have created an artificial numerical example to highlight certain characteristics of the spectral clustering. Firstly, we want to prove that there are networks with a large number of zero eigenvalues, for which the spectral clustering by Ebli & Spreemann (2019) cannot be immediately applied and thus we need dimensionality reduction. Secondly, we want to see the effect of the dimensionality reduction on the performances.

Briefly, we have considered networks of 20 vertices, whose 3-simplices are arranged in two clusters, densely connected internally and loosely connected within each other. An arbitrary fraction of 4-simplices is added to simulate the presence and absence of holes. We test spectral clustering with various configurations: the dimensionality of the embedding Euclidean space is $q \in \{3, 5, 7, 9\}$, the eigenvalue threshold is $\tau \in \{1e{-}7, 1e{-}2, 1e{-}1, 5e{-}1, 1e0\}$, and the dimensionality reduction technique can be no technique at all, Haar-based, or PCA-based. For a detailed explanation of our experimental setup, the reader can refer to Appendix C.

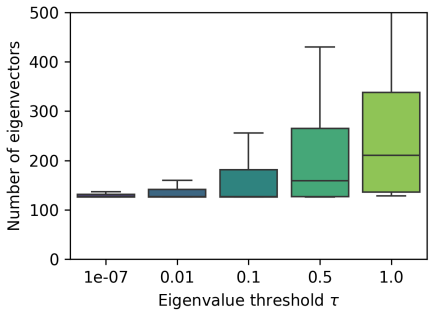


Figure 2. Number of eigenvectors of $\mathbf{L}_k$ having corresponding eigenvalue lower than τ for ASCs with few 4-simplices.

In Figure 2 we show that is effectively possible to construct networks with a large number of zero eigenvalues. This case has been generated by keeping only one-fourth of the possible 4-simplices in Σ.

In Figure 3, we depict the score, defined as the normalized mutual information score, between the ground truth and the result of clustering with spectral embedding followed by k-means. In this case, we observe that increasing the threshold τ for small values of q has little effect, although it becomes non-negligible for higher values of q. There is often an optimal value of τ; in our example, it is 5×10^{-1}. Beyond this threshold, the inclusion of other eigenvectors worsens the performance. There is no clear best dimensionality reduction technique, and this aspect is expected to vary from task to task. For a further discussion of these results, the reader can refer to Appendix C.

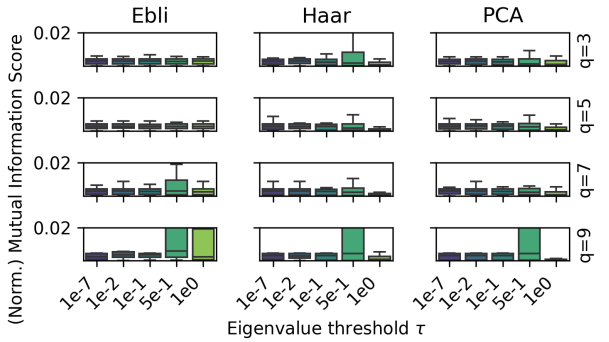


Figure 3. Normalized mutual information score of the result of spectral clustering and the ground truth for different threshold τ, dimensionality reduction technique (columns) and embedding size (rows).

6. Conclusions

We have introduced a quantum algorithm to perform spectral clustering on the k-simplices of a network. Our work exploits quantum computing to efficiently encode and eigendecompose the Laplacian operator $\mathbf{L}_k$ associated with the network. Building upon and improving the algorithm by Ebli & Spreemann (2019), which introduced the concept of spectral clustering on classical computers, our approach overcomes limitations by allowing the use of a richer set of eigenvectors. In contrast to the previous approach, our method leverages random projection, enabling to consider of an arbitrary number of eigenvectors and potentially providing a more comprehensive understanding of the relationships between simplices. We conclude by briefly commenting on potential applications and future directions of this research.

6.1. Potential applications in biological networks

We argue that, among the myriad applications of spectral clustering for complex networks, a particularly promising domain is that which intersects with biological networks.

In genetics, systems biology, and neuroscience, the networks are often constructed as correlation networks (graphs): an edge (i, j) is present if its score $s(i, j)$ falls below a specific threshold. Here, the score function is a metric of statistical significance between the data associated with member i of the population and member j. Notably, these structures

can and have been generalized to utilize simplices, thereby capturing higher-order interactions.

A convenient aspect of applying quantum algorithms to correlation networks lies in our ability to verify the presence of a k-simplex through an $O(\mathrm{poly}(n,k))$ procedure in superposition. This avoids the need for classical enumeration of all simplices and loading them into a quantum state (e.g. via a QRAM, Giovannetti, Lloyd, and Maccone 2008).[4]

Another critical point concerns the asymptotic runtime of our approach: it depends on the density of k-simplices ζ_k and the fraction of low eigenvalues ξ_k. For the procedure to be feasible for large networks, this fraction must be (small) inversely polynomial in n. We can use various score functions, including the χ^2-test, quadratic regression, K2-score, and maximum likelihood model, see e.g. Blumenthal et al. (2021). The choice of the score function can lead to the construction of radically different networks, some of which may exhibit more favorable characteristics than others, i.e. higher ζ_k and ξ_k. This flexibility is a unique advantage not possible when the network's topology is given a priori.

6.2. Future directions

Three important topics that have yet to be explored are the following ones.

Dequantization Apers et al. (2023) have shown that it is possible to estimate normalized Betti numbers efficiently on a classical computer, for certain classes of networks. Other properties, such as spectral entropy, are shown to be much harder to capture without a quantum device. Whether the spectral clustering can be executed efficiently on a classical device is definitively an open problem. An interesting direction can be the dequantization, i.e. the search for a classical algorithm running in a time polynomial in the time of the quantum algorithm; this might be obtained via the path integral Monte Carlo or tensor networks. Anyway, even if we succeed in the dequantization of this algorithm, a polynomial speedup of the quantum algorithm is still of great interest in a practical setting.

Use of the qubiterate The *qubiterate* operator Ω proposed by Berry et al. (2018) has been used to sample the eigenvalues of a Hamiltonian operator H with a constant amount of query to its block encoding, without the need of the Hamiltonian simulation – leading to shorter circuits and basically no error in the eigenvalue estimation. This is possible because the eigenvalues of H and Ω are easily related. As for the spectral clustering, we are instead relying on the eigenvectors; it is still to investigate how the eigenvectors of H and Ω are related and if we can exploit the qubiterate to build much shorter quantum circuits.

Timeframe for wide availability The algorithm described is not by any means suited to be executed on near-term quantum devices. While the Betti number estimation algorithm has been modified in Ubaru et al. (2021) reaching a quantum circuit depth of only $O(n)$ and making their algorithm feasible for early-term fault-tolerant quantum devices, our spectral clustering algorithm requires the expensive quantum phase estimation. Exploring whether is possible to access the eigenvectors of $\mathbf{L}_k$ without quantum phase estimation might help achieve a runtime similar to Ubaru et al. (2021).

[4] This is not an issue for ASCs effectively described by Maximal Simplicial Trees (Boissonnat et al., 2017), which efficiently represents them as a list of maximal simplices, albeit complex networks benefiting this representation are rare.

References

Aktar, S., Bärtschi, A., Badawy, A. H., and Eidenbenz, S. A divide-and-conquer approach to Dicke state preparation. *IEEE Transactions on Quantum Engineering*, 3:1–16, 2022.

Albert, R. and Barabási, A.-L. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1): 47–97, 2002.

Alvarez-Rodriguez, U., Battiston, F., de Arruda, G. F., Moreno, Y., et al. Evolutionary dynamics of higher-order interactions in social networks. *Nature Human Behaviour*, 5(5):586–595, 2021.

Apers, S., Gribling, S., Sen, S., and Szabó, D. A (simple) classical algorithm for estimating Betti numbers. *Quantum*, 7:1202–1216, 2023.

Barbarossa, S. and Sardellitti, S. Topological signal processing over simplicial complexes. *IEEE Transactions on Signal Processing*, 68:2992–3007, 2020.

Battiston, F., Cencetti, G., Iacopini, I., Latora, V., et al. Networks beyond pairwise interactions: Structure and dynamics. *Physics Reports*, 874:1–92, 2020.

Battiston, F., Amico, E., Barrat, A., Bianconi, G., et al. The physics of higher-order interactions in complex systems. *Nature Physics*, 17(10):1093–1098, 2021.

Berry, D. W., Kieferová, M., Scherer, A., Sanders, Y. R., et al. Improved techniques for preparing eigenstates of fermionic hamiltonians. *npj Quantum Information*, 4(1): 22–29, 2018.

Berry, D. W., Su, Y., Gyurik, C., King, R., Basso, J., Barba, A. D. T., Rajput, A., Wiebe, N., Dunjko, V., and Babbush, R. Analyzing prospects for quantum advantage in topological data analysis. *arXiv preprint arXiv:2209.13581*, 2023.

Blumenthal, D. B., Baumbach, J., Hoffmann, M., Kacprowski, T., et al. A framework for modeling epistatic interaction. *Bioinformatics*, 37(12):1708–1716, 2021.

Boissonnat, J.-D., Karthik, C., and Tavenas, S. Building efficient and compact data structures for simplicial complexes. *Algorithmica*, 79(2):530–567, 2017.

Boyer, M., Brassard, G., Høyer, P., and Tapp, A. Tight bounds on quantum searching. *Fortschritte der Physik: Progress of Physics*, 46(4-5):493–505, 1998.

Brassard, G., Høyer, P., and Tapp, A. Quantum counting. In *Automata, Languages and Programming: 25th International Colloquium, ICALP'98*, pp. 820–831. Aalborg, Denmark, 1998.

Cade, C., Labib, F., and Niesen, I. Quantum motif clustering. *Quantum*, 7:1046–1097, 2023.

Chen, Y.-C. and Meila, M. The decomposition of the higher-order homology embedding constructed from the k-laplacian. *Advances in Neural Information Processing Systems*, 34:15695–15709, 2021.

Comon, P. Independent component analysis, a new concept? *Signal processing*, 36(3):287–314, 1994.

De Domenico, M. and Biamonte, J. Spectral entropies as information-theoretic tools for complex network comparison. *Physical Review X*, 6(4):41062–41076, 2016.

Dong, Y., Meng, X., Whaley, K. B., and Lin, L. Efficient phase-factor evaluation in quantum signal processing. *Physical Review A*, 103(4):042419, 2021.

Duval, A. and Reiner, V. Shifted simplicial complexes are Laplacian integral. *Transactions of the American Mathematical Society*, 354(11):4313–4344, 2002.

Ebli, S. and Spreemann, G. A notion of harmonic clustering in simplicial complexes. In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pp. 1083–1090. Boca Raton, Florida, USA, 2019.

Fagiolo, G., Squartini, T., and Garlaschelli, D. Null models of economic networks: the case of the world trade web. *Journal of economic interaction and coordination*, 8:75–107, 2013.

Ghojogh, B., Ghodsi, A., Karray, F., and Crowley, M. Johnson-Lindenstrauss lemma, linear and nonlinear random projections, random fourier features, and random kitchen sinks: Tutorial and survey. *arXiv preprint arXiv:2108.04172*, 2021.

Gilyén, A., Su, Y., Low, G. H., and Wiebe, N. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pp. 193–204, 2019.

Giovannetti, V., Lloyd, S., and Maccone, L. Quantum random access memory. *Physical review letters*, 100(16): 160501–160505, 2008.

Girvan, M. and Newman, M. E. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002.

Giusti, C., Ghrist, R., and Bassett, D. S. Two's company, three (or more) is a simplex: Algebraic-topological tools for understanding higher-order structure in neural data. *Journal of computational neuroscience*, 41:1–14, 2016.

Golub, G. H. and Van Loan, C. F. *Matrix computations*. JHU press, 4th edition, 2013.

Grande, V. P. and Schaub, M. T. Disentangling the spectral properties of the Hodge Laplacian: Not all small eigenvalues are equal. *arXiv preprint arXiv:2311.14427*, 2023a.

Grande, V. P. and Schaub, M. T. Topological point cloud clustering. *arXiv preprint arXiv:2303.16716*, 2023b.

Gundert, A. and Szedlák, M. Higher dimensional Cheeger inequalities. In *Proceedings of the thirtieth annual symposium on Computational geometry, SOCG'14*, pp. 181–188. Kyoto, Japan, 2014.

Gyurik, C., Cade, C., and Dunjko, V. Towards quantum advantage via topological data analysis. *Quantum*, 6: 855–893, 2022.

Hatcher, A. *Algebraic topology*. Cambridge University Press, Cambridge, 1st edition, 2002.

Hayakawa, R. Quantum algorithm for persistent Betti numbers and topological data analysis. *Quantum*, 6:873–900, 2022.

Hoffmann, M., Poschenrieder, J. M., Incudini, M., Baier, S., et al. Network medicine-based epistasis detection in complex diseases: ready for quantum computing. *Medrxiv preprint 2023.11.07.23298205*, pp. 1–11, 2023.

Hu, L., Zhang, J., Pan, X., Yan, H., et al. Hiscf: leveraging higher-order structures for clustering analysis in biological networks. *Bioinformatics*, 37(4):542–550, 2021.

Iacopini, I., Petri, G., Barrat, A., and Latora, V. Simplicial models of social contagion. *Nature communications*, 10 (1):2485–2494, 2019.

Johnson, W. B. and Lindenstrauss, J. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary Mathematics*, 26:189–206, 1984.

Kailing, K., Kriegel, H.-P., and Kröger, P. Density-connected subspace clustering for high-dimensional data. In *Proceedings of the 2004 SIAM international conference on data mining*, pp. 246–256. Lake Buena Vista, Florida, USA, 2004.

Kerenidis, I. and Landman, J. Quantum spectral clustering. *Physical Review A*, 103(4):042415, 2021.

Krishnagopal, S. and Bianconi, G. Spectral detection of simplicial communities via Hodge laplacians. *Physical Review E*, 104(6):64303–64321, 2021.

Kumaran, K., Sajjan, M., Oh, S., and Kais, S. Random projection using random quantum circuits. *Physical Review Research*, 6(1):13010–13018, 2024.

Leditto, C. M. G., Southwell, A., Tonekaboni, B., White, G. A., et al. Topological signal processing on quantum computers for higher-order network analysis. *arXiv preprint arXiv:2312.07672*, 2023.

Li, Q., Huang, Y., Jin, S., Hou, X., et al. Quantum spectral clustering algorithm for unsupervised learning. *Science China Information Sciences*, 65(10):200504–200514, 2022.

Lim, L.-H. Hodge Laplacians on graphs. *Siam Review*, 62 (3):685–715, 2020.

Lin, L. Lecture notes on quantum algorithms for scientific computation. *arXiv preprint arXiv:2201.08309*, 2022.

Liu, H.-L., Wan, L.-C., Yu, C.-H., Pan, S.-J., Qin, S.-J., et al. A quantum algorithm for solving eigenproblem of the laplacian matrix of a fully connected weighted graph. *Advanced Quantum Technologies*, pp. 2300031–2300047, 2023.

Lloyd, S., Garnerone, S., and Zanardi, P. Quantum algorithms for topological and geometric analysis of data. *Nature communications*, 7(1):10138–10145, 2016.

Low, G. H. and Chuang, I. L. Optimal hamiltonian simulation by quantum signal processing. *Physical review letters*, 118(1):010501, 2017.

MacQueen, J. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pp. 281–297. Oakland, CA, USA, 1967.

Manenti, R. and Motta, M. *Quantum Information Science*. Oxford University Press, Oxford, 2023.

McArdle, S., Gilyén, A., and Berta, M. A streamlined quantum algorithm for topological data analysis with exponentially fewer qubits. *arXiv preprint arXiv:2209.12887*, 2022.

McClean, J. R., Boixo, S., Smelyanskiy, V. N., Babbush, R., et al. Barren plateaus in quantum neural network training landscapes. *Nature communications*, 9(1):4812–4818, 2018.

Newman, M. *Networks*. Oxford University Press, Oxford, 2nd edition, 2018.

Newman, M. E. Modularity and community structure in networks. *Proceedings of the national academy of sciences*, 103(23):8577–8582, 2006.

Palukuri, M. V., Patil, R. S., and Marcotte, E. M. Molecular complex detection in protein interaction networks through reinforcement learning. *BMC bioinformatics*, 24(1):306–333, 2023.

Reddy, T. S., Chepuri, S. P., and Borgnat, P. Clustering with simplicial complexes. *arXiv preprint arXiv:2303.07646*, 2023.

Roddenberry, T. M. and Segarra, S. Hodgenet: Graph neural networks for edge data. In *2019 53rd Asilomar Conference on Signals, Systems, and Computers*, pp. 220–224. Pacific Grove, CA, USA, 2019.

Saito, N., Schonsheck, S. C., and Shvarts, E. Multiscale transforms for signals on simplicial complexes. *Sampling Theory, Signal Processing, and Data Analysis*, 22(1):2–27, 2024.

Schaeffer, S. E. Graph clustering. *Computer Science Review*, 1(1):27–64, 2007.

Schaub, M. T., Benson, A. R., Horn, P., Lippner, G., et al. Random walks on simplicial complexes and the normalized hodge 1-laplacian. *SIAM Review*, 62(2):353–391, 2020.

Schaub, M. T., Zhu, Y., Seby, J.-B., Roddenberry, T. M., et al. Signal processing on higher-order networks: Livin'on the edge... and beyond. *Signal Processing*, 187: 108149–108168, 2021.

Schmidhuber, A. and Lloyd, S. Complexity-theoretic limitations on quantum algorithms for topological data analysis. *PRX Quantum*, 4(4):40349–40365, 2023.

Schuld, M. and Killoran, N. Is quantum advantage the right goal for quantum machine learning? *Prx Quantum*, 3(3): 30101–30114, 2022.

Serrano, D. H. and Gómez, D. S. Centrality measures in simplicial complexes: Applications of topological data analysis to network science. *Applied Mathematics and Computation*, 382:125331, 2020.

Sim, S., Johnson, P. D., and Aspuru-Guzik, A. Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. *Advanced Quantum Technologies*, 2(12):1900070–1900085, 2019.

Song, E. Quantum walk on simplicial complexes for simplicial community detection. *arXiv preprint arXiv:2401.00699*, 2024.

St-Onge, G., Iacopini, I., Latora, V., Barrat, A., et al. Influential groups for seeding and sustaining nonlinear contagion in heterogeneous hypergraphs. *Communications Physics*, 5(1):25–41, 2022.

Ubaru, S., Akhalwaya, I. Y., Squillante, M. S., Clarkson, K. L., et al. Quantum topological data analysis with linear depth and exponential speedup. *arXiv preprint arXiv:2108.02811*, 2021.

Von Luxburg, U. A tutorial on spectral clustering. *Statistics and computing*, 17:395–416, 2007.

Yoder, T. J., Low, G. H., and Chuang, I. L. Fixed-point quantum search with an optimal number of queries. *Physical review letters*, 113(21):210501–210506, 2014.

Zhou, J., Wong, M. S., Chen, W.-C., Krainer, A. R., et al. Higher-order epistasis and phenotypic prediction. *Proceedings of the National Academy of Sciences*, 119(39): e2204233119, 2022.

A. Background in quantum computing

We here review some basic concepts of quantum computation. The reader can refer to Manenti & Motta (2023) for a more detailed explanation.

A.1. Fundamentals of quantum mechanics in the Dirac notation

A n-qubit quantum state is described by a vector $|\psi\rangle$, denoted as 'ket', that lives in the Hilbert space $\mathcal{H}(\mathbb{C})^{\otimes n} \cong \mathbb{C}^{2^n}$ and satisfies

$$\||\psi\rangle\|_2 = 1. \tag{19}$$

The space is spanned by the elements of the computational basis, denoted with $|0\rangle, |1\rangle, \dots, |2^n - 1\rangle$, with the number being sometimes denoted in binary. The vector $|i\rangle$ admit representation as the column vector $[\delta_{ij}]_{j=0}^{2^n-1} \in \mathbb{C}^{2^n}$. Through the document, the initial state of the quantum system is always $|0\rangle = [\delta_{0j}]_{j=0}^{2^n-1}$. An element of the dual space of $\mathcal{H}$ is $\langle\psi|$, denoted as 'bra', admit representation as a row vector that is the conjugate transpose of the corresponding ket. The notion $\langle\phi|\psi\rangle$ denotes an inner product, while $|\phi\rangle\langle\psi|$ denotes the outer product and results in a rank-1 projector.

A quantum system evolves according to the Schrödinger equation,

$$i\hbar \frac{\partial}{\partial t} |\psi(t)\rangle = H |\psi(t)\rangle , \tag{20}$$

where $\hbar$ is set to one and the solution of the equation is a *unitary operator* $U = \exp(-itH)$, i.e. an operator such that $UU^\dagger = U^\dagger U = \mathbb{I}_{2^n}$. Thus, the system evolves according to

$$|\psi\rangle \mapsto U |\psi\rangle \tag{21}$$

A *quantum measurement* or positive-operator valued measure (POVM) is a set of positive semi-definite operators $\{M_i\}_{i=1}^m$ such that $\sum_i M_i^\dagger M_i = \mathbb{I}_{2^n}$. When measuring a quantum system in the state $|\psi\rangle$ via the given POVM, we obtain the outcome $i \in \{1, \dots, m\}$ with probability

$$p(\text{outcome } i) = \langle\psi|M_i^\dagger M_i|\psi\rangle , \tag{22}$$

after which the quantum state collapses to the state

$$|\psi\rangle \mapsto \frac{M_i |\psi\rangle}{\langle\psi|M_i^\dagger M_i|\psi\rangle}. \tag{23}$$

A.2. Fundamentals of quantum mechanics in the density matrix notation

A more powerful formalism that takes into account a mixture of quantum states is the formalism of density matrices. A n-qubit quantum state is described by a positive semi-definite density matrix ρ, that lives in the Hilbert space $\mathcal{L}(\mathcal{H}) \cong \mathbb{C}^{2^n \times 2^n}$ and satisfies

$$\mathrm{Tr}[\rho] = 1. \tag{24}$$

If ρ is a rank-1 projector, it is called a pure state; otherwise, it is called a mixed state and can be expressed as a convex combination of pure states,

$$\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i| \tag{25}$$

with $p_i \geq 0$ for all i, and $\sum_i p_i = 1$. The mixed state can be interpreted as a probability distribution over pure states. A quantum computation via the unitary U maps the system to

$$\rho \mapsto U\rho U^\dagger. \tag{26}$$

A POVM $(M_1, \dots, M_m)$, applied to the quantum system in the state ρ ends up with the outcome $i \in \{1, \dots, m\}$ with probability

$$p(\text{outcome } i) = \mathrm{Tr}[M_i^\dagger M_i \rho] \tag{27}$$

after which the quantum state collapses to the state

$$\rho \mapsto \frac{M_i \rho M_i^\dagger}{\mathrm{Tr}[M_i^\dagger M_i \rho]}. \tag{28}$$

For a bipartite quantum system $\mathcal{H} = \mathcal{H}_A \otimes \mathcal{H}_B$ in the state ρ, a partial trace is defined as

$$\rho_A = \mathrm{Tr}_B[\rho] = \sum_{i_B} (\mathbb{I}_A \otimes \langle i_B|)\, \rho\, (\mathbb{I}_A \otimes |i_B\rangle) \tag{29}$$

where $|i_B\rangle$ is an element of the computational basis of system B, and is equivalent to measuring the system B and discarding the result. A *quantum operation* or *quantum channel* Φ is a linear operator between density matrices that is completely positive and trace-preserving. The unitary evolution, restriction (partial trace), and post-selection can be all described as quantum channels.

A.3. Quantum circuits

A quantum computation task can be represented by a quantum circuit. In analogy with classical circuits, we use a diagrammatic representation having one wire per qubit and apply quantum gates corresponding to unitary operations on the qubit to which they are applied. In contrast to classical circuits, each gate has the same number of fan-ins and fan-outs, as the theory of quantum information prohibits otherwise (no-cloning and no-deleting theorems). A (non-minimal) universal set of gates, i.e. any quantum circuit is efficiently decomposable into a sequence of gates from such a set, is made up of the following elements:

Identity gate	$\mathbb{I} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	Pauli X	$\mathrm{X} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$
Pauli Y	$\mathrm{Y} = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$	Pauli Z	$\mathrm{Z} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$
Hadamard	$H = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$	Controlled-NOT	$\mathrm{CNOT} = \lvert 0\rangle\langle 0\rvert \otimes \mathbb{I} + \lvert 1\rangle\langle 1\rvert \otimes \mathrm{X}$
Phase gate	$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$	Controlled-U	$\mathrm{C}\text{-}U = \lvert 0\rangle\langle 0\rvert \otimes \mathbb{I} + \lvert 1\rangle\langle 1\rvert \otimes U$
T gate	$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$	Rotation over σ	$R_\sigma(\theta) = \exp\left(-i\frac{\theta}{2}\sigma\right)$

Here, σ is one of the Pauli matrices X, Y, Z.

A.4. Matrix arithmetic on a quantum computer

An important feature of quantum computing is the ability to perform matrix arithmetic on large matrices, leading to important speedups. Let H be a complex matrix of size $2^n \times 2^n$, which for simplicity of presentation we set to be a Hermitian operator.

We can access the matrix according to different input models. A popular model is the Hamiltonian simulation, encoding the unitary $U = \exp(-itH)$ for some t. The simplest Hamiltonian simulation technique is the Trotter-Suzuki algorithm, which requires us to decompose H into a sum of tensor products of Pauli matrices, which is not always convenient as we might have up to $4^n - 1$ terms in this sum.

Alternatively, we can rely on a *block encoding*, a procedure that directly encodes H by embedding it as a block of a larger unitary matrix. This procedure is especially convenient for sparse matrices. Formally, the unitary U_H over $n + m$ qubits is an (α, m, ϵ)-block encoding for H, with $\|H\| \leq \alpha + \epsilon$, if

$$\|H - \alpha(\langle 0|^{\otimes m} \otimes \mathbb{I}_n) U_H (|0\rangle^{\otimes m} \otimes \mathbb{I}_n)\| \leq \epsilon. \tag{30}$$

In this case, U_H takes the form

$$U_H = \begin{pmatrix} H/\alpha & \cdot \\ \cdot & \cdot \end{pmatrix}. \tag{31}$$

We can perform arithmetic operations on these matrices by applying the Quantum Signal Processing (Low & Chuang, 2017) or the Quantum Singular Value Transformation (Gilyén et al., 2019) algorithms, for Hermitian and general matrices, respectively. These algorithms apply an arbitrary polynomial p (usually a low-degree one) to the input matrix, and this results in a block encoding of $p(H)$.

B. Construction of the quantum circuits

We detail the construction of the single components leading to the quantum circuit in Figure 1.

B.1. Construction of Φ_{HS}

The transformation Φ_{HS} is composed of three parts, depicted in Figure 4:

1. the sampling of the k-simplices; this is obtained via a unitary U_{SAMP} returning the coherent superposition of k-simplices in the input ASC Σ, then a measure operation modifies the coherent superposition to a classical mixture of pure states.
2. the Hamiltonian simulation of the (normalized) Laplacian $\mathbf{L}_k$, which we obtain by creating a block-encoding for $\mathbf{L}_k$ and then applying the Quantum Signal Processing algorithm (Low & Chuang, 2017) to create a block-encoding of $e^{-i/\|\mathbf{L}_k\|_2 \mathbf{L}_k}$.
3. the Quantum Phase Estimation and subsequent post-selection, to obtain a superposition of the bottom eigenvectors.

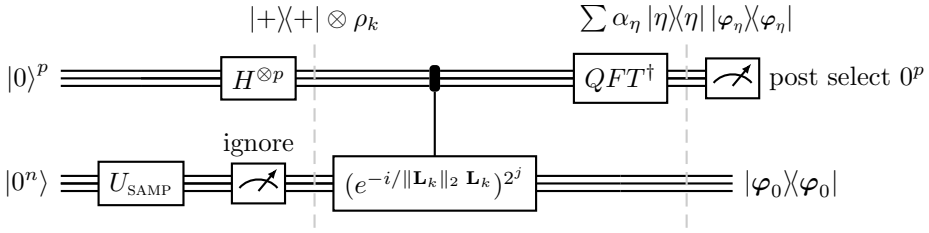


Figure 4. Quantum circuits for Φ_{HS}.

B.1.1. Efficient sampling of k-simplices

According to Figure 4, we load the classical mixture of k-simplices by first applying the unitary U_{SAMP} and then performing a measurement operation. This procedure allows us to obtain a quadratic speedup compared to the classical procedure, which simply consists of sampling the k-simplices classically. We discuss both procedures here.

A simple and classical procedure for sampling k-simplices of Σ involves randomly generating a set of $k+1$ vertices and checking if it forms a simplex in the given ASC. This checking operation is typically efficient (in time polynomial in the number of vertices n), even for clique complexes for constant and small k and for ASCs described as maximal simplicial trees. The success probability of this classical procedure is denoted by ζ_k, where $\zeta_k = |\Sigma_k| / \binom{n}{k+1}$ represents the density of k-simplices in Σ. This procedure is efficient when $\zeta_k^{-1} \in \text{poly}(n)$ or for relatively small and constant k (having the whole search space $\binom{n}{k+1}$ polynomial in n).

In contrast, Grover's search (Manenti & Motta, 2023) can achieve a quadratic speedup over the classical procedure by defining U_{SAMP} as

$$U_{\text{SAMP}} = (P(2\,|0\rangle\langle 0| - \mathbb{I})P^\dagger O_{\Sigma_k})^r P, \tag{32}$$

where O_{Σ_k} is the oracle checking whether a k-simplex belongs to Σ,

$$O_{\Sigma_k}\,|\sigma\rangle = \begin{cases} |\sigma\rangle\,, & \sigma \notin \Sigma_k \\ -\,|\sigma\rangle\,, & \sigma \in \Sigma_k \end{cases}; \tag{33}$$

P is the unitary that defines the search space of Grover's search, loading the superposition of all the items in the wanted search space (for $P = H^{\otimes n}$ we load all the possible 2^n elements); $2\,|0\rangle\langle 0| - \mathbb{I}$ is a reflection implemented via a multi-controlled-Z with negated control; and finally, r is the number of repetitions.

Note that we can restrict the search space to only the solutions having Hamming weight $k+1$; in this case, P prepares the Dicke state $|D_{k+1}\rangle$, representing the superposition of states with Hamming weight $k+1$:

$$|D_{k+1}\rangle = \sum_{\mathrm{HW}(b)=k+1} |b\rangle\,. \tag{34}$$

The search space will consist of $\binom{n}{k+1} \ll 2^n$ elements. We can follow the procedure outlined in (Aktar et al., 2022), which is deterministic and has a depth of $O(kn)$.

For $N = \binom{n}{k+1}$ possible k-simplices in the search space and $M = |\Sigma_k|$ solutions, the classical procedure has complexity $O(N/M)$. In contrast, Grover's search needs a number of repetitions r in Equation (32) equal to

$$r = \left\lfloor \frac{\pi}{4}\sqrt{\zeta_k^{-1}} \right\rfloor. \tag{35}$$

For this, the length of the circuit is $O(\sqrt{N/M})$, leading to the quadratic speedup.

Determining the number of repetitions r is challenging as $|\Sigma_k|$ is unknown. This is crucial since r is the optimal value, and both its over- and under-estimation result in a larger error. Among the possible solutions, one could employ a trial-and-error approach (Boyer et al., 1998) that allows estimating $|\Sigma_k|$ up to a multiplicative error. A more precise estimation is given if we employ quantum counting (Brassard et al., 1998), at a much larger computational cost. Alternatively, the fixed-point Grover's search (Yoder et al., 2014) does not require prior knowledge of $|\Sigma_k|$, at a slightly larger cost in terms of depth.

B.1.2. Block-encoding of $\mathbf{L}_k$

The construction of the operator $\exp(-i/\|\mathbf{L}_k\|_2 \mathbf{L}_k)$ requires us to have access to a block encoding of $\mathbf{L}_k$. Here, with $\mathbf{L}_k$, we denote $\mathbf{L}_k^\Sigma$, but we avoid explicitly referring to Σ for clarity of notation.

The procedure relies on $\mathbf{L}_k$ being a sparse and row-efficient operator. This allows us to use the scheme in (Lin, 2022). It creates a block encoding of a s-sparse Hermitian operator H, and uses $m+1$ ancillary qubits. The operator must be scaled so that $\|H\|_2 \leq 1$ for the block encoding to exist. We need to define two oracles $O_{\text{row}}, O_{\text{entry}}$ as follows:

$$O_{\text{row}} |\ell\rangle_m |j\rangle_n = |\ell\rangle_m |c_{j,\ell}\rangle_n \tag{36}$$

$$O_{\text{entry}} |0\rangle |\ell\rangle_m |j\rangle_n = (H_{c_{j,\ell},j} |0\rangle + \sqrt{1 - |H_{c_{j,\ell},j}|^2}\, |1\rangle) |\ell\rangle_m |j\rangle_n \tag{37}$$

where $c_{j,\ell}$ is the row-index of the ℓ-th nonzero element in the j-th column. The corresponding quantum circuit is depicted in Figure 5. The procedure succeeds with probability $\|H|\psi\rangle\|$, where $|\psi\rangle$ is the state input to the Hermitian operator. We also want the block encoding to be Hermitian itself, for that we can rely on the circuit in Figure 6 using one extra ancilla and two applications of the non-Hermitian block encoding.

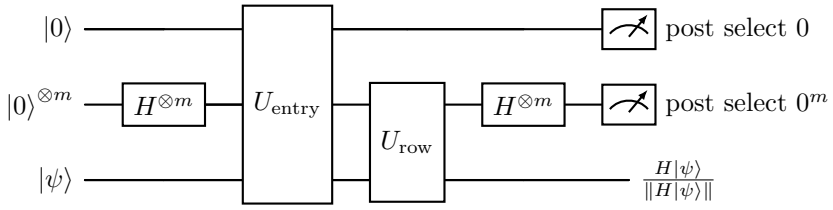


Figure 5. Block encoding of a M-sparse Hermitian operator. The procedure is especially convenient for $M = 2^m$, as that allows us to use $H^{\otimes m}$ to load the superposition of all the M indices.

Note that, although it is possible to encode $\mathbf{L}_k$ directly, which is n^2 sparse, it is much more convenient to build the Dirac operator $\mathbf{D}_k$, which is n-sparse, and apply it twice as $\mathbf{L}_k^\Sigma = (\mathbf{D}_k^\Sigma)^2$. We recall the definition of the Dirac operator:

$$\mathbf{D}_k^\Sigma = \begin{pmatrix} 0 & \mathbf{B}_k^\Sigma & 0 \\ (\mathbf{B}_k^\Sigma)^\top & 0 & \mathbf{B}_{k+1}^\Sigma \\ 0 & (\mathbf{B}_{k+1}^\Sigma)^\top & 0 \end{pmatrix}, \quad (\mathbf{D}_k^\Sigma)^2 = \begin{pmatrix} \mathbf{L}_{k-1}^{\Sigma,\uparrow} & 0 & 0 \\ 0 & \mathbf{L}_k^\Sigma & 0 \\ 0 & 0 & \mathbf{L}_{k+1}^{\Sigma,\downarrow} \end{pmatrix}. \tag{38}$$

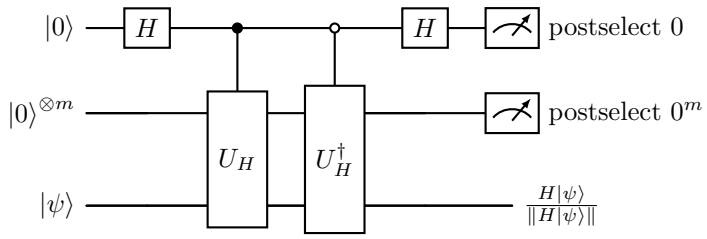


Figure 6. Given a possibly-non-Hermitian block-encoding of H, this quantum circuit builds a Hermitian block-encoding at the cost of using two queries to the original encoding and one extra ancillary qubit.

The row index oracle for $\mathbf{D}_k^\Sigma$ is defined as follows:

$$O_{\text{row}} |l\rangle_m |\sigma\rangle_n = \begin{cases} |\ell\rangle_m |\sigma \setminus \{v_\ell\}\rangle_n , & \ell < \text{HW}(\sigma) \\ |\ell\rangle_m |\sigma \cup \{\bar{v}_\ell\}\rangle_n , & \text{otherwise} \end{cases} . \tag{39}$$

For $\ell < \text{HW}(\sigma)$ (the valid inputs), the row index corresponds to the ℓ-th element in the column σ and corresponds to the row of the face σ without the vertex v_ℓ. For $\ell \geq \text{HW}(\sigma)$, it will denote an invalid input and be erased by having a coefficient zero by O_{entry}. However, the output must be chosen to make the oracle unitary: when ℓ exceeds the Hamming weight, we add the ℓ-th vertex of the complement simplex of σ.

The oracle O_{row} works by applying the unitary U_{SIEF} $2n$ times. U_{SIEF} stands for 'if Set then Inc, if Equals then Flip'. It acts on the register $|\ell\rangle_m$, a counter register of the same size $|\text{cnt}\rangle_m$, and one qubit $|v_i\rangle$ associated with the i-th vertex in the simplex σ:

$$U_{\text{SIEF}} |\ell\rangle_m |\text{cnt}\rangle_m |b\rangle = \begin{cases} |\ell\rangle_m |\text{cnt}+1\rangle_m |1 \oplus b\rangle , & b = 1 \text{ and } \ell = \text{cnt}+1 \\ |\ell\rangle_m |\text{cnt}+1\rangle_m |b\rangle , & b = 1 \text{ and } \ell \neq \text{cnt}+1 \\ |\ell\rangle_m |\text{cnt}\rangle_m |b\rangle , & b = 0 \end{cases} . \tag{40}$$

The unitary U_{SIEF} is shown in Figure 7.

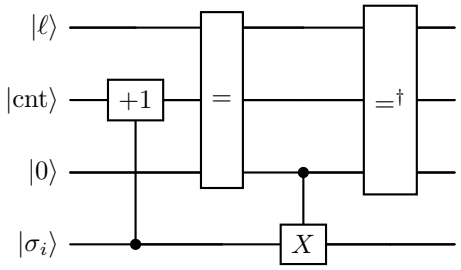


Figure 7. Quantum circuits for U_{SIEF}. If the register σ_i the counter is incremented. The unitary $(=)$ and its inverse $(=)^\dagger$ compare the two registers and set the output flag if the registers are equal. If the flag is set, the value of σ_i is flipped. At the end, the intermediate register $|0\rangle$ has been uncomputed.

To implement U_{row}, we assume the counter register is initialized to -1, so we start decrementing it. Then, for each vertex, we apply the unitary U_{sief}, which will flip (unset) the ℓ-th nonzero vertex and increment the counter register by $\text{HW}(\sigma)$. We are not done, as we cannot uncompute the counter register yet. For that, we negate the σ register and apply U_{sief} again, which will not unset any vertex but will increment by $n - \text{HW}(\sigma)$ the counter register, which can now be uncomputed with a decrement operation. Finally, the σ register is restored. The quantum circuit is shown in Figure 8.

The entry oracle U_{entry} is defined by

$$O_{\text{entry}} |0\rangle |\ell\rangle_m |\sigma\rangle_n = \begin{cases} (-1)^\ell |0\rangle |\ell\rangle |\sigma\rangle , & \sigma \in \Sigma \text{ and } \text{HW}(\sigma) = k+1 \text{ and } 0 \leq \ell \leq k \\ (-1)^\ell |0\rangle |\ell\rangle |\sigma\rangle , & \sigma \in \Sigma \text{ and } \text{HW}(\sigma) = k+2 \text{ and } 0 \leq \ell \leq k+1 \\ |1\rangle |\ell\rangle |\sigma\rangle , & \text{otherwise} \end{cases} \tag{41}$$

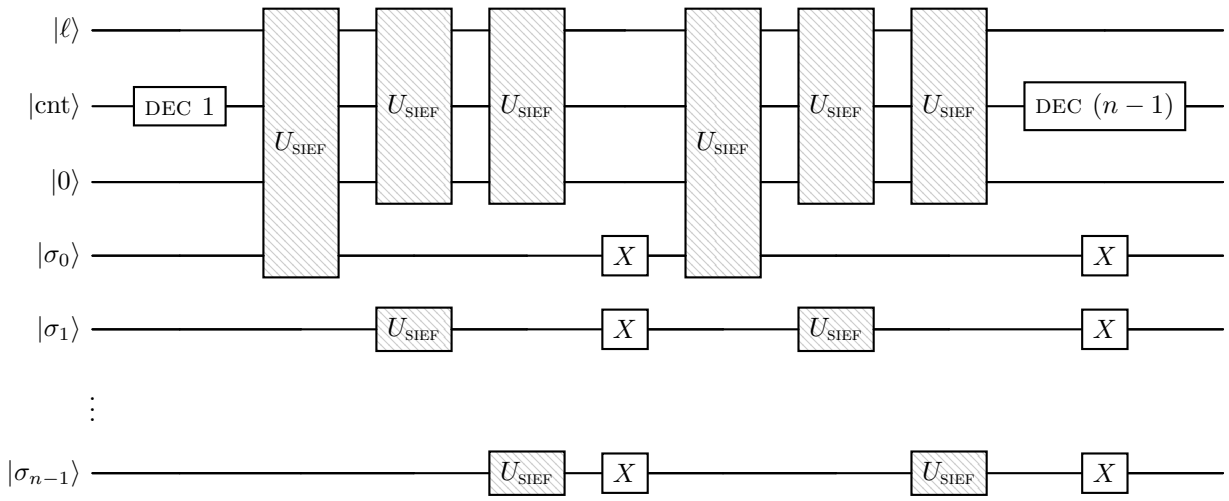


Figure 8. Quantum circuits for U_{ROW}.

The ancillary qubit mark to non-zero elements only for $\sigma \in \Sigma_k$ ($\text{HW}(\sigma) = k+1$ and $\sigma \in \Sigma$), and corresponds to filling the block $\mathbf{B}_k^\Sigma$ and adjoint, and for $\sigma \in \Sigma_{k+1}$ (block $\mathbf{B}_{k+1}^\Sigma$ and adjoint). The parity of ℓ decide if the sign is $+1$ (rotation $R_y(-\pi)$) or -1 (rotation $R_y(\pi)$). Its quantum circuit is shown in Figure 9.

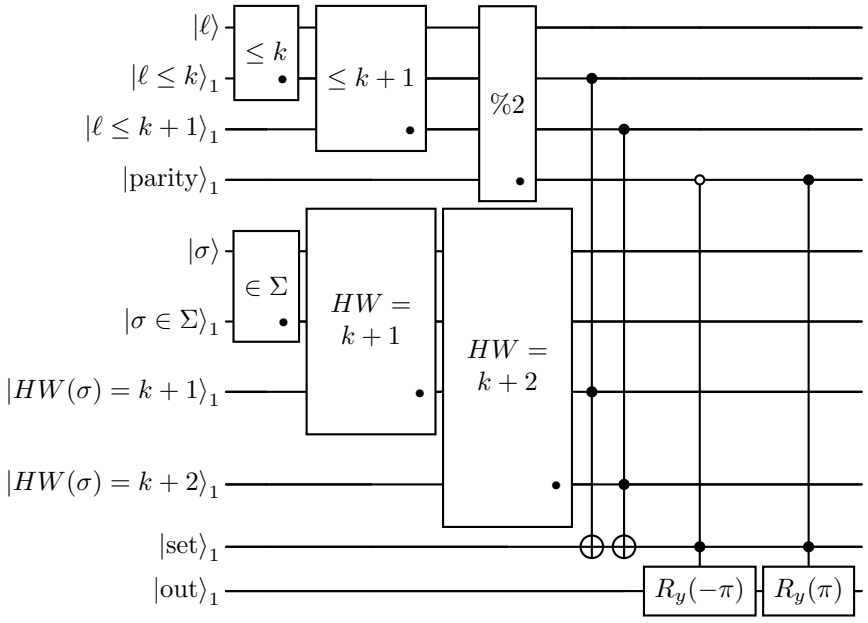


Figure 9. Quantum circuits for U_{ENTRY}. All the auxiliary registers have to be uncomputed at the end by applying each gate in the reverse order (not explicitly shown in the picture).

B.1.3. Hamiltonian simulation

Once we have access to the (normalized) block encoding of $\mathbf{L}_k$, we can construct $e^{-i\mathbf{L}_k}$. This task is known as Hamiltonian simulation, and we can use the Quantum Signal Processing (QSP) algorithm, as this problem has a complexity matching its lower bound (Low & Chuang, 2017).

Before delving into the details of this algorithm, let's introduce a few unitaries that will be used in QSP. Consider the

definition of the CNOT gate,

$$\text{CNOT} = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes X, \tag{42}$$

where the action of this operator applies the bitflip operation X whenever the control qubit is in the image of the projector $\Pi = |1\rangle\langle 1|$. We can generalize this operation by defining the *projector-controlled NOT*, denoted as C_ΠNOT, as

$$\text{C}_\Pi\text{NOT} = (I - \Pi) \otimes I + \Pi \otimes X. \tag{43}$$

We will particularly need the case for $\Pi = |0^m\rangle\langle 0^m|$, for which C_ΠNOT is an $m+1$ qubit operator. In this case, the m control qubits are controlled by the value 0, and the bitflip operation is applied to the target qubit. Furthermore, note that a reflection over Π is obtained by applying a Hadamard operation on the target qubit before and after the C_ΠNOT,

$$U_\Pi = 2\Pi - I = (I \otimes H)\text{C}_\Pi\text{NOT}(I \otimes H). \tag{44}$$

The quantum circuit of U_π requires $m+1$ qubits, although the last qubit (the target of the C_ΠNOT) can be ignored as it will remain untouched after the operation, and the gate can act on m qubits only. Furthermore, we will need a *projector-controlled phase* gate, denoted as $e^{i\phi U_\pi}$ and acting on $m+1$ qubits, defined by

$$e^{i\phi U_\Pi} = \text{C}_\Pi\text{NOT}(e^{i\phi Z} \otimes I_m)\text{C}_\Pi\text{NOT}. \tag{45}$$

Finally, note that

$$U_\Pi = (-i)e^{i\frac{\pi}{2}U_\pi}. \tag{46}$$

The QSP revolves around the following theorem (Gilyén et al., 2019; Lin, 2022). Let U_H be a (α, m)-block encoding of H. Let P, Q be complex polynomials in x, and $d > 0$ positive integer. Then, if the following conditions are satisfied: (1) $\deg(P) \leq d$ and P has even parity; (2) $\deg(Q) \leq d$ and Q has odd parity; (3) $|P(x)|^2 + (1-x^2)|Q(x)|^2$ for $x \in [-1, 1]$. Then, there exists a set of phases $\Phi = (\phi_0, \ldots, \phi_d) \in \mathbb{R}^{d+1}, \phi_j \in [-\pi, \pi)$ such that the unitary (cf. Figure 10)

$$U_\Phi = \left[\prod_{j=0}^{d-1} e^{i(\frac{\pi}{2}+\phi_j)U_\Pi} U_H\right] e^{i\phi_d U_\Pi}, \tag{47}$$

where U_H is a Hermitian block encoding of H, takes the form

$$U_\Phi = \begin{pmatrix} P(H) & -iQ(H)\sqrt{1-H^2} \\ iQ^\dagger(H)\sqrt{1-H^2} & P^\dagger(H) \end{pmatrix}, \tag{48}$$

meaning U_Φ is a $(\alpha, m+1)$-block encoding of $P(H)$.

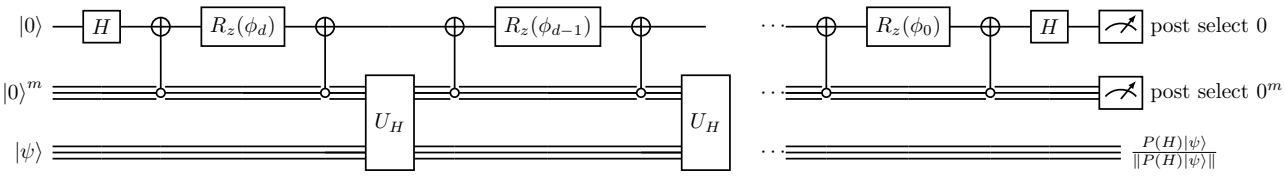


Figure 10. Quantum circuit for quantum signal processing, meaning the construction of the block encoding of the polynomial $P(H)$.

The phase factors ϕ_i can be found numerically and efficiently via the approach proposed by Dong et al. (2021). The same authors made available the software QSPPACK (`https://github.com/qsppack/QSPPACK`).

The function $P(x) = e^{-ix}$ has no defined parity, thus the theorem does not apply. However, we can use Euler's formula to divide into the sum of $\cos(x)$ and $\sin(x)$ and proceed via a linear combination of unitaries, as in Figure 11.

Note that the advancement of quantum algorithms for Betti number estimation could also impact our quantum spectral clustering algorithm. Some of the alternative schemes we could have used to define Φ_{HS} include the approach in McArdle et al. (2022), which has introduced a more compact encoding for the k-simplices (compared to the one in Equation (10), that uses $O(k \log n)$ instead of $O(n)$ qubits. Ubaru et al. (2021) has introduced an alternative encoding that does not rely on the row and entry oracles. Furthermore, the work by Berry et al. (2023) has introduced several major improvements, including the use of a Kaiser window in the quantum phase estimation.

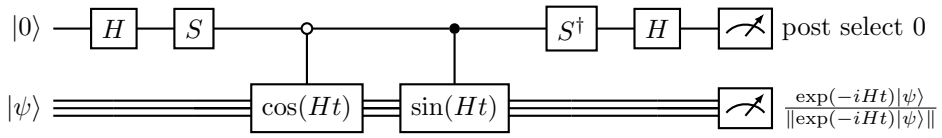


Figure 11. Quantum circuits for Hamiltonian simulation using the block encodings of $\cos(Ht)$ and $\sin(Ht)$.

B.2. Construction of U_{RND}

We can efficiently create an approximate unitary 2-design using local random circuits. Among the various ansatz options, one of the most popular is the one presented by McClean et al. (2018), known to converge to an exact 2-design beyond a certain depth. The unitary (over n qubits) is defined as:

$$U_{\text{RND}} = \left[\prod_{i=1}^{\ell}\prod_{j=1}^{n-1} \text{CZ}^{(j,j+1)} \prod_{j=1}^{n} R_{P_{i,j}}(\theta_{i,j})\right]\left[\prod_{i=1}^{n} R_{\text{Y}}^{(i)}(\frac{\pi}{4})\right], \tag{49}$$

as depicted in Figure 12. Here, ℓ is the number of layers (proportional to the depth), $P_{i,j}$ are Pauli matrices uniformly sampled in $\{\text{X}, \text{Y}, \text{Z}\}$, and $\theta_{i,j}$ are real values uniformly sampled in $[0, 2\pi)$.

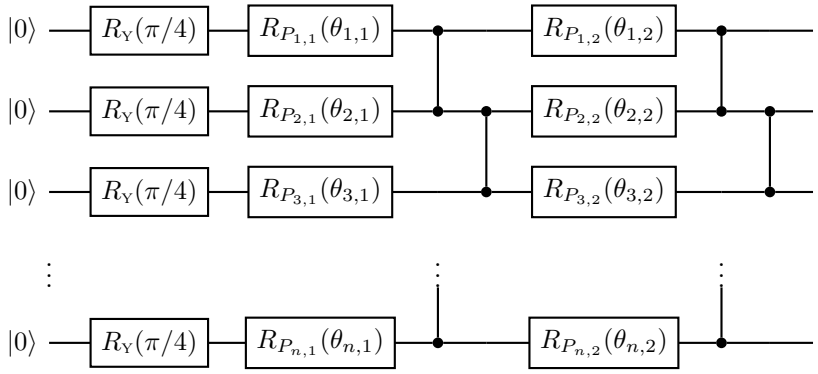


Figure 12. Quantum circuits U_{RND} with $\ell = 2$ layers. To obtain an approximate unitary 2-design, it is sufficient to set $\ell = O(n)$ where n is the number of qubits.

It has been proven by Kumaran et al. (2024) that random quantum circuits with a depth of $O(n)$ are sufficient to obtain the expected results, specifically random projections satisfying the Johnson-Lindenstrauss lemma.

B.3. Construction of Π_i

A projection can be implemented via a Hadamard test together with a post-selection operation. Consider the simple example in Figure 13(a) implementing the single-qubit projector $|0\rangle\langle 0|$ or $|1\rangle\langle 1|$ depending on the post-selection value. The system evolves as follows:

$$\begin{aligned}
&(H \otimes \mathbb{I}) \cdot \text{CZ} \cdot (H \otimes \mathbb{I}) \cdot |0\rangle\,|\psi\rangle \\
=&(H \otimes \mathbb{I}) \cdot \text{CZ} \cdot \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right)|\psi\rangle \\
=&(H \otimes \mathbb{I}) \cdot \left(\frac{1}{\sqrt{2}}|0\rangle\,\mathbb{I}\,|\psi\rangle + \frac{1}{\sqrt{2}}|1\rangle\,\text{Z}\,|\psi\rangle\right) \\
=&|0\rangle \underbrace{\frac{(\mathbb{I}+\text{Z})}{2}}_{|0\rangle\langle 0|}|\psi\rangle + |1\rangle \underbrace{\frac{(\mathbb{I}-\text{Z})}{2}}_{|1\rangle\langle 1|}|\psi\rangle\,.
\end{aligned}$$

Then, post-selection on the ancillary qubit of the value 0 leads to the projector $\Pi = |0\rangle\langle 0|$, while a post-selection on the same qubit of the value 1 leads to $\Pi = |1\rangle\langle 1|$.

The process can be easily extended to multiple qubits, as for the two-qubit projector in Figure 13(b).

$$\begin{aligned}
&(H \otimes \mathbb{I} \otimes \mathbb{I}) \cdot \mathrm{CCZ} \cdot (H \otimes \mathbb{I} \otimes \mathbb{I}) \cdot |0\rangle_1 |\psi\rangle_2 \\
=&(H \otimes \mathbb{I} \otimes \mathbb{I}) \cdot \mathrm{CCZ} \cdot \left(\frac{1}{\sqrt{2}} |0\rangle_1 + \frac{1}{\sqrt{2}} |1\rangle_1 \right) |\psi\rangle_2 \\
=&(H \otimes \mathbb{I} \otimes \mathbb{I}) \cdot \left(\frac{1}{\sqrt{2}} |0\rangle \mathbb{I} |\psi\rangle + \frac{1}{\sqrt{2}} |1\rangle \mathrm{CZ} |\psi\rangle \right) \\
=&|0\rangle \underbrace{\frac{(\mathbb{I} + \mathrm{CZ})}{2}}_{\mathbb{I} - |11\rangle\langle 11|} |\psi\rangle + |1\rangle \underbrace{\frac{(\mathbb{I} - \mathrm{CZ})}{2}}_{|11\rangle\langle 11|} |\psi\rangle .
\end{aligned}$$

A simple modification of this scheme allows us to implement any projector. For example, negating before and after the control and Z implement the projector $|00\rangle\langle 00|$, while, in general, for $\Pi = |t\rangle\langle t|$ and $P |0\rangle^{\otimes n} = |t\rangle$, we can use the scheme in Figure 13(c).

Consider the implementation of the projector Π_i, $i = 1, \ldots, q$, over n qubits, where let $q = 2^m$ for simplicity. Each projector can be defined to be the sum of $N = 2^n/2^m$ states of the computational basis. This results in

$$\Pi_i = \sum_{j=1}^{N} |iN + j\rangle\langle iN + j| , \tag{50}$$

which can be easily obtained with the scheme in Figure 13(c) using as P the binary representation of i in the first m qubits and placing a Hadamard gate on the remaining ones.

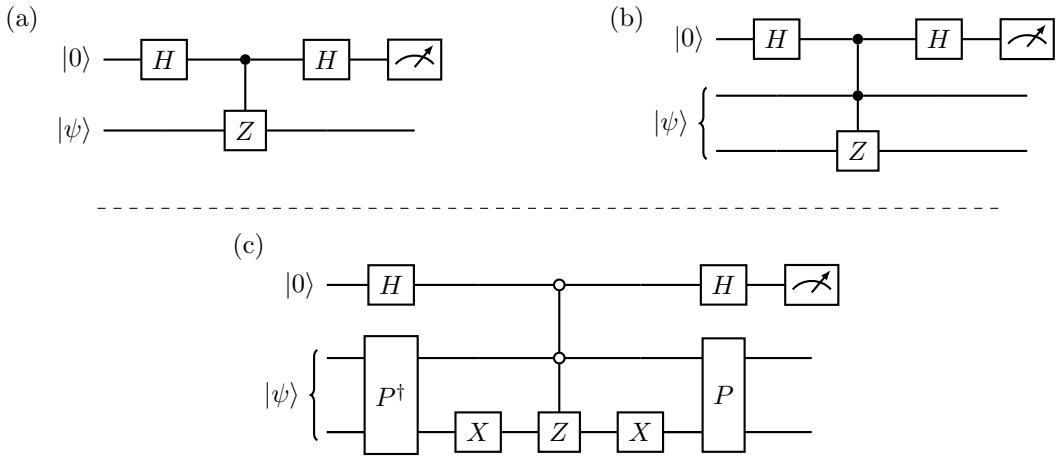


Figure 13. Quantum circuit to create projections. (a) Quantum circuit whose postselection on the ancillary qubit of value 1 leads to the single qubit projection $\Pi = |1\rangle\langle 1|$. (b) Quantum circuit whose postselection on the ancillary qubit of value 1 leads to the two-qubit projection $\Pi = |11\rangle\langle 11|$. (c) Quantum circuit whose postselection on the ancillary qubit of value 0 leads to the arbitrary qubits projection $\Pi = |s\rangle\langle s|$ for $P |0\rangle = |s\rangle$.

B.4. Overlap test

The overlap between two states $|\psi\rangle , |\phi\rangle$, once we are able to prepare them via the unitaries

$$U_\psi |0\rangle = |\psi\rangle \tag{51}$$

$$U_\phi |0\rangle = |\phi\rangle \tag{52}$$

is obtained via the Hadamard test, repeated twice to estimate separately the real and imaginary parts of the inner product (cf. Figure 14 and Figure 15). The controlled transformation in both quantum circuits is

$$U = U_\phi^\dagger U_\psi. \tag{53}$$

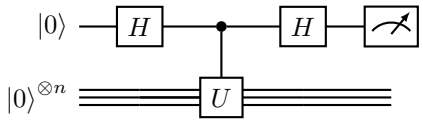


Figure 14. Hadamard test whose expectation value $\langle \mathrm{Z} \otimes \mathbb{I}^{\otimes n} \rangle$ (probability of obtaining output 0 minus the probability of obtaining 1) corresponds to $\mathrm{Re}(\langle 0| U |0\rangle)$.

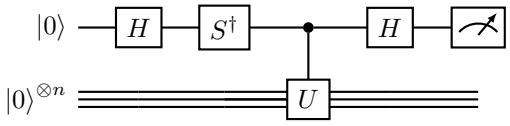


Figure 15. Hadamard test whose expectation value $\langle \mathrm{Z} \otimes \mathbb{I}^{\otimes n} \rangle$ (probability of obtaining output 0 minus the probability of obtaining 1) corresponds to $\mathrm{Im}(\langle 0| U |0\rangle)$.

C. Extended description of the numerical experiment

C.1. Extended setup

C.1.1. Data generation

The ASCs are generated using the `generate_abstract_simplicial_complex` function. This function takes the number of vertices n, the number of clusters m, the interaction size to analyze k, and a density parameter $0 < p < 1$ as inputs.

We create m $\lfloor n/m \rfloor$-simplices, each acting on independent vertices, then define the ASCs with maximal simplices corresponding to the determined $\lfloor n/m \rfloor$-simplices. This process generates many k- and $(k+1)$-simplices, which are the primary objects of our analysis. To loosely connect initially disconnected clusters, we introduce additional k-simplices. The number of bridge simplices is randomly sampled from a range of 0 to $600 \times m$. Finally, to simulate the presence of holes in the ASC, a certain percentage (determined by the parameter p) of $(k+1)$-simplices is removed.

The ground truth for clustering is defined such that any k-simplex belongs to cluster $i \in \{1, \ldots, m\}$ if the majority of its vertices belong to the i-th maximal simplex. It is important to note that this clustering task is highly artificial and may not necessarily represent real-world use cases. However, its flexibility allows us to simulate various scenarios, including denser or sparser situations with few or many holes.

In our experiments, we set the number of vertices to $n = 20$, the number of clusters to $m = 2$, and the interaction size to $k = 3$. The density ranges in $p \in \{0.25, 0.33, 0.50, 0.66, 0.75\}$. For each different value of p, we generate 50 ASCs.

C.1.2. Spectral clustering

The spectral embedding is calculated using the `spectral_embedding` function, which takes the ASC as input, along with the target dimensionality of the Euclidean space q, the eigenvalue threshold τ, and the chosen dimensionality reduction technique (no reduction, Haar random unitaries, or PCA).

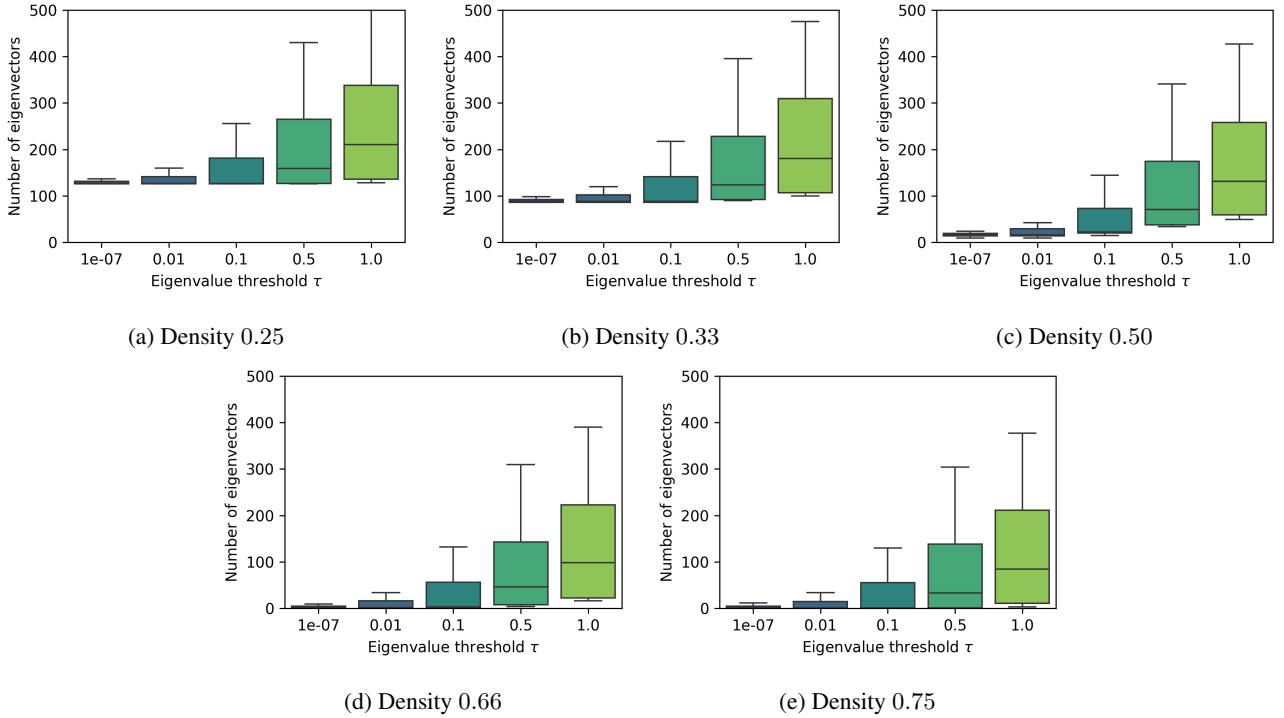
The process begins by computing the Laplacian matrix $\mathbf{L}_k$ and its eigendecomposition. Subsequently, components with eigenvalues $< \tau$ are filtered out. In cases where all components are deleted (which can rarely happen for simplices with no holes and $\tau \approx 0$), the bottom two components are selected to proceed. If no dimensionality reduction is applied, only the bottom q components are considered, and the rest are disregarded. If there are fewer than q elements, the process continues with a lower target dimensionality of the Euclidean space.

Alternatively, one of the supported dimensionality reduction techniques can be applied. The Haar-random unitary technique, which is more easily implementable on quantum devices, is one option. PCA is also included for a more extensive comparison.

Spectral clustering is tested with various configurations: the dimensionality of the embedding Euclidean space is set to $q \in \{3, 5, 7, 9\}$, the eigenvalue threshold is varied across $\tau \in \{1e-7, 1e-2, 1e-1, 5e-1, 1e0\}$, and all the described dimensionality reduction techniques are considered, including no reduction.

The performance of the clustering is assessed using the normalized mutual information score, where accuracy serves as a pertinent measure in this context. Take, for instance, the ground truth $[0, 1]$ and the clustering result $[1, 0]$. The correct normalized mutual information score is 1.0 because a permutation of the predicted labels aligns with the ground truth. However, the accuracy would be 0.0 as it does not consider the permutation of labels.

C.2. Extended results



(a) Density 0.25 (b) Density 0.33 (c) Density 0.50

(d) Density 0.66 (e) Density 0.75

Figure 16. Number of eigenvalues below the threshold τ of the operator $\mathbf{L}_3$ calculated on ASC with different density of 4-simplices kept.

In Figure 16, we show the expected number of "small" eigenvalues of the Laplacian $\mathbf{L}_3$ when creating the ASCs on 20 vertices using the technique detailed in Appendix C.1.1. For small values of the density p, such as $p = 0.25$ and $p = 0.33$, we observe that the number of zero eigenvectors ($\tau \approx 0$ plus a small numerical error) is very large—on average, beyond 100. This implies that, in this case, it is challenging to directly apply the spectral clustering approach in Ebli & Spreemann (2019). In contrast, with higher values of p, e.g. $p = 0.66$ and $p = 0.75$, we encounter fewer holes, and Ebli & Spreemann (2019) can be employed. However, at this point, it is unclear whether this is convenient in terms of performance.

In Figure 17, the normalized mutual information score in various configurations is presented. For lower densities, e.g., $p = 0.25, p = 0.33$, where a large number of zero eigenvectors is present, the introduction of even more components is not beneficial. In this situation, the best performance is obtained by applying no dimensionality reduction. Surprisingly, maintaining a very low target dimensionality of the Euclidean space ($q = 3$) yields comparable performance to higher values ($q = 9$).

For higher densities, e.g., $p = 0.50$, which has a few tenths of zero eigenvalues, adding more components is beneficial up to the threshold $t = 1e - 2$ for Haar dimensionality reduction. However, adding even more components degrades performance.

This already suggests the trend that the fewer zero eigenvalues there are in the spectrum of the Laplacian, the more we need to include in the embedding to compensate for the limited topological information. Furthermore, PCA has proven to be a more robust approach; including components beyond $\tau = 1e - 2$ does not degrade performance, as they likely do not significantly affect the principal components. From a quantum perspective, though implementing PCA is feasible, it is not as efficient and immediate as an approximate unitary 2-design.

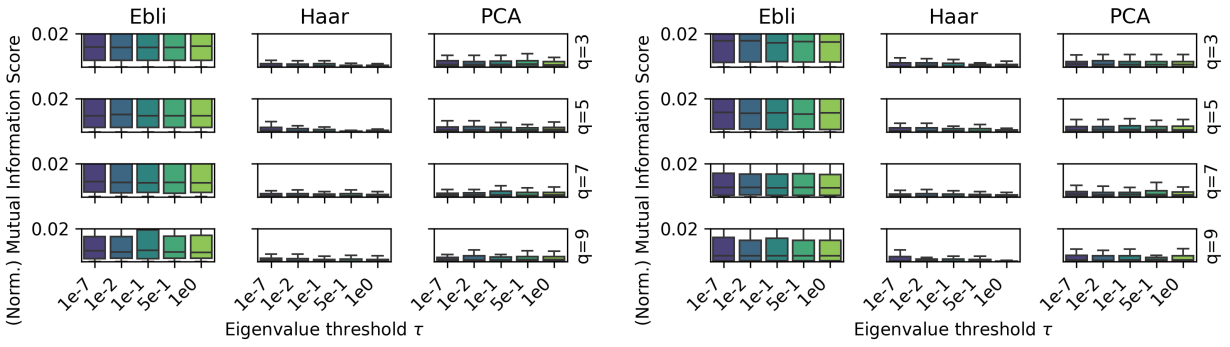
For even higher densities, e.g., $p = 0.66, p = 0.75$, we confirm the trend that adding more components is necessary to increase performance. The optimal value is $\tau = 5 \times 10e - 1$. Increasing the dimensionality of the Euclidean space has the desired effect of improving performance, as expected. In these latter experiments, however, the use of PCA is comparable to Haar dimensionality reduction, and it is not necessarily true that the approach is robust in this case.

C.3. Code

The code used to perform the experiments has been written in Python 3. The experiments can be reproduced by either uploading the Python notebook file `quantum_spectral_clustering.ipynb` to Google Colab, or by setting up an environment that includes Python 3.10.12 with PIP version 23.3.2. In the latter case, the required libraries can be installed using the attached file `requirements.txt`. This configuration has been tested on Ubuntu 22.04.3 LTS.

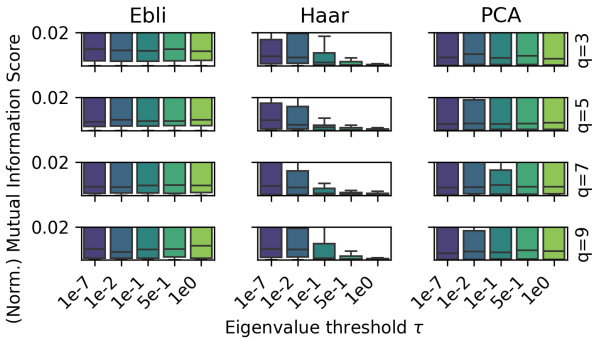
C.4. Data

To recreate the plots without rerunning the experiments from scratch, you can load the file `df_densityXX.pickle`. Here, `XX` can be $25, 33, 50, 66, 75$, and corresponds to the percentage of 4-simplices removed from the ASC; lower values of `XX` lead to simplices with more holes.

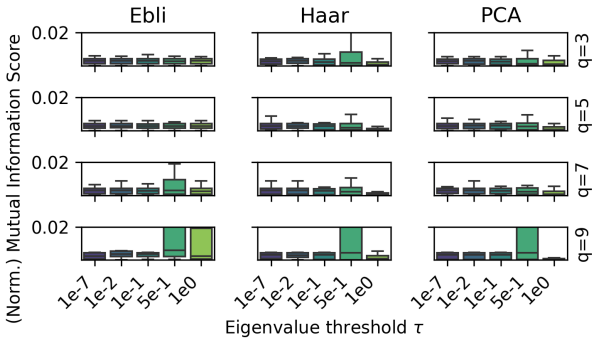


(a) Density 0.25

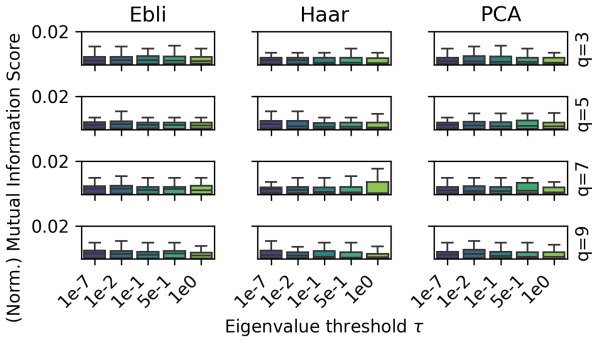
(b) Density 0.33



(c) Density 0.50



(d) Density 0.66



(e) Density 0.75

Figure 17. Normalized mutual information with respect to the eigenvalue threshold τ (on the x-axis), target dimensionality of the Euclidean space (on the rows), dimensionality reduction technique (on the columns; Ebli: no reduction, Haar: Haar-random unitary, PCA: principal component analysis) and density of 4-simplices. The score is calculated via the spectral embedding followed by k-means.