



**HAL**  
open science

## A (simple) classical algorithm for estimating Betti numbers

Simon Apers, Sayantan Sen, Dániel Szabó

► **To cite this version:**

Simon Apers, Sayantan Sen, Dániel Szabó. A (simple) classical algorithm for estimating Betti numbers. *Quantum*, 2023, 7, pp.1202. <10.22331/q-2023-12-06-1202>. <hal-03882485>

**HAL Id: hal-03882485**

**<https://hal.science/hal-03882485v1>**

Submitted on 13 Nov 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY 4.0 - Attribution - International License

# A (simple) classical algorithm for estimating Betti numbers

Simon Apers<sup>1</sup>, Sander Gribling<sup>2</sup>, Sayantan Sen<sup>3</sup>, and Dániel Szabó<sup>1</sup>

<sup>1</sup>Université Paris Cité, CNRS, IRIF, Paris, France

<sup>2</sup>Tilburg University, The Netherlands

<sup>3</sup>National University of Singapore, Singapore

We describe a simple algorithm for estimating the  $k$ -th normalized Betti number of a simplicial complex over  $n$  elements using the path integral Monte Carlo method. For a general simplicial complex, the running time of our algorithm is  $n^{\mathcal{O}\left(\frac{1}{\sqrt{\gamma}} \log \frac{1}{\varepsilon}\right)}$  with  $\gamma$  measuring the spectral gap of the combinatorial Laplacian and  $\varepsilon \in (0, 1)$  the additive precision. In the case of a clique complex, the running time of our algorithm improves to  $(n/\lambda_{\max})^{\mathcal{O}\left(\frac{1}{\sqrt{\gamma}} \log \frac{1}{\varepsilon}\right)}$  with  $\lambda_{\max} \geq k$ , where  $\lambda_{\max}$  is the maximum eigenvalue of the combinatorial Laplacian. Our algorithm provides a classical benchmark for a line of quantum algorithms for estimating Betti numbers. On clique complexes it matches their running time when, for example,  $\gamma \in \Omega(1)$  and  $k \in \Omega(n)$ .

## 1 Introduction

An abstract *simplicial complex*  $K$  is a family of sets or *faces* that is downward closed under the subset relation. A canonical example of a simplicial complex is the *clique complex* of a graph, where the sets correspond to the cliques in the graph.

There is a recent surge of interest in using simplicial complexes to model higher-order relations in data sets – a technique often referred to as “*topological data analysis*” [6]. In these studies, an object of particular interest is the homology or Betti number of a simplicial complex, which characterizes the number of high-dimensional holes in the complex. Unfortunately, efficiently computing the Betti number seems like a challenging task. Indeed, it was recently shown that even multiplicatively approximating the Betti number of a simplicial complex or clique complex is hard even for quantum computers<sup>1</sup>.

Thus the next natural question is whether we can *additively* approximate the Betti number. More formally, given a parameter  $\varepsilon \in (0, 1)$ , can we efficiently output an estimate  $\tilde{\nu}_k$  of the  $k$ -th (normalized) Betti number of the complex satisfying

$$\tilde{\nu}_k = \frac{\beta_k}{d_k} \pm \varepsilon,$$

where  $d_k$  denotes the number of  $k$ -faces in the complex. To the best of our knowledge, Elek [11] was the first to study this question. In [11], it was proved that if the complex has constant degree, that is, every 0-face (vertex) of the complex is contained in a constant

---

<sup>1</sup>Multiplicative approximation of Betti numbers is QMA1-hard [7], where QMA1 is the 1-sided error version of QMA, and QMA is the quantum analog of NP.

number of 1-faces (edges), then there is an algorithm whose running time depends only on the parameter  $\varepsilon$ . The constant degree assumption however implies that the complex has constant dimension as well (that is, it contains no  $k$ -faces for  $k \in \omega(1)$ ), and thus all Betti numbers  $\beta_k$  for non-constant  $k$  are zero.

The problem was later reconsidered by Lloyd, Garnerone, and Zanardi [19], who proposed a *quantum* algorithm for estimating the Betti number<sup>2</sup>. Their algorithm combines quantum techniques such as Hamiltonian simulation and quantum phase estimation. Assuming that we can efficiently sample a uniformly random  $k$ -face from the complex, the algorithm outputs an  $\varepsilon$ -additive approximation of  $\beta_k$  in time

$$\text{poly}(n, 1/\gamma, 1/\varepsilon),$$

where  $n = d_0$  denotes the number of 0-faces in the complex, and  $\gamma$  is the (normalized) spectral gap of the  $k$ -th combinatorial Laplacian. As current classical algorithms for calculating Betti numbers seem to run in time  $\text{poly}(d_k)$  [13], which can be  $\text{poly}(n^k)$ , this suggests an exponential speedup of quantum algorithms over classical ones for this problem. This explains the surge of interest in the quantum algorithm, and in particular in its application to clique complexes, which have a concise  $\text{poly}(n)$ -size description [1, 2, 4, 7, 17, 20, 23].

**Contributions:** In this work, we describe a simple classical algorithm for approximating Betti numbers. We refer to Theorem 1.1 for a formal statement about the complexity of our algorithm, here we discuss its implications. Our algorithm provides a natural benchmark for the aforementioned quantum algorithms. Similarly to these quantum algorithms, our algorithm is efficient if the gap and the precision are “*large*”. However, while the quantum algorithms can afford a gap  $\gamma$  and precision  $\varepsilon$  as small as  $1/\text{poly}(n)$ , our algorithm requires these to be *constant* for general complexes. This is similar to the dequantization results from [14]. In the case of clique complexes, we can go further. For example, if  $k \in \Omega(n)$  then we can afford precision  $\varepsilon = 1/\text{poly}(n)$  (if the gap is constant) or gap  $\gamma = \Omega(1/\log^2 n)$  (if the precision is constant). Overall, this further pins down the region where we can expect a quantum advantage for the problem of estimating Betti numbers.

In a nutshell, we base our algorithm on a random variable whose expectation is close to  $\beta_k/d_k$  and whose variance is (sufficiently) small. Crucially, we show that we can efficiently generate samples from this random variable. Standard concentration bounds can be used to bound the required number of samples and hence establish the complexity of our algorithm. More precisely, the algorithm is based on the technique of path integral Monte Carlo [3], akin to the Ulam-von Neumann algorithm for solving linear systems [12]. Our result is formally stated below. By  $\Gamma(\Delta_k)$  we denote the *spectral gap* of the combinatorial Laplacian  $\Delta_k$ , which is equal to its smallest nonzero eigenvalue.

**Theorem 1.1.** *Let  $\Delta_k$  denote the  $k$ -th combinatorial Laplacian of the complex. Assume that in time  $\text{poly}(n)$  we can (i) draw a  $k$ -face uniformly at random, and (ii) check whether a set is in the complex. Given an estimate  $\hat{\lambda} \in \Theta(\lambda_{\max}(\Delta_k))$  and a lower bound  $\gamma$  such that  $\Delta_k$  has spectral gap  $\Gamma(\Delta_k) \geq \gamma\hat{\lambda}$ , there exists a classical algorithm that, for any  $\varepsilon > 0$ , outputs with high probability an estimate  $\tilde{\nu}_k = \beta_k/d_k \pm \varepsilon$  of the  $k$ -th (normalized) Betti number of a general simplicial complex in time*

$$n^{\mathcal{O}\left(\frac{1}{\sqrt{\gamma}} \log \frac{1}{\varepsilon}\right)},$$

---

<sup>2</sup>More correctly, [19] and follow-up works output an approximation  $\tilde{\beta}_k = \beta_k \pm \varepsilon d_k$  of the actual Betti number, rather than the normalized version that we consider. This however requires a (multiplicative) estimate of  $d_k$ . While such an estimate can be obtained efficiently if e.g. the complex is “dense” (i.e.,  $d_k/\binom{n}{k+1} = 1/\text{poly}(n)$ ), we avoid this restriction.

| Algorithm                 | Complexes                 | Complexity is $\text{poly}(n)$ if  |
|---------------------------|---------------------------|--|
| quantum algorithm of [19] | general                   | $\gamma, \varepsilon \in \Omega(1/\text{poly}(n))$   |
| this work                 | general                   | $\gamma, \varepsilon \in \Omega(1)$  |
| this work                 | clique, $k \in \Omega(n)$ | $\gamma \in \Omega(1), \varepsilon \in \Omega(1/\text{poly}(n))$<br>or $\gamma \in \Omega(1/\log^2(n)), \varepsilon \in \Omega(1)$ |

Table 1: Comparison of the parameter settings of quantum and classical algorithms for the Betti number estimation problem under which their running time is polynomial.

and of a clique complex in time

$$\binom{n}{\frac{n}{\hat{\lambda}}}^{\mathcal{O}\left(\frac{1}{\sqrt{\gamma}} \log \frac{1}{\varepsilon}\right)} \cdot \text{poly}(n).$$

The algorithm has space complexity  $\text{poly}(n, 1/\gamma, \log(1/\varepsilon))$ .

For general complexes, our algorithm improves upon the aforementioned classical algorithms if  $k \in \Omega(1/\sqrt{\gamma})$ . Since  $n \geq \lambda_{\max}(\Delta_k) \geq k + \delta_k + 1$  (see Lemma 2.4), with  $\delta_k$  being the maximum up-degree<sup>3</sup> over all  $k$ -faces, we can simply set  $\hat{\lambda} = n$  if  $k \in \Omega(n)$  or if we know that  $\delta_k \in \Omega(n)$ . In such case, the algorithm for clique complexes runs in time  $2^{\mathcal{O}\left(\frac{1}{\sqrt{\gamma}} \log \frac{1}{\varepsilon}\right)} \cdot \text{poly}(n)$ . This is polynomial if either  $\gamma \in \Omega(1)$  and  $\varepsilon = 1/\text{poly}(n)$ , or  $\gamma \in \Omega(1/\log^2 n)$  and  $\varepsilon \in \Omega(1)$ . The algorithm provides a classical counterpart to the aforementioned line of quantum algorithms for estimating Betti numbers which, under similar assumptions, have a runtime scaling as  $\text{poly}(n, 1/\gamma, 1/\varepsilon)$ . We summarize these findings in Table 1.

The complexity of our algorithm for general simplicial complexes can alternatively be obtained using the singular value transformation (SVT) algorithm<sup>4</sup> from Gharibian and Le Gall [14]. The main difference is that we use a path integral Monte Carlo approach for computing matrix powers, instead of computing them explicitly as in [14, Lemma 3]. This approach provides us with an exponential improvement in the space complexity since the SVT algorithm has space complexity  $n^{\mathcal{O}\left(\frac{1}{\gamma} \log \frac{1}{\varepsilon}\right)}$ . The more significant benefit is that we get an improved algorithm for clique complexes, which is the main case of interest for the aforementioned quantum algorithms. We show that the  $k$ -th combinatorial Laplacian is  $n$ -sparse for clique complexes, as compared to general complexes which are  $\mathcal{O}(kn)$ -sparse. This implies that it is closer to a diagonally dominant matrix, and we can exploit this for designing better algorithms using the path integral Monte Carlo technique.

**Open problems** A natural question is to what extent we can improve our results. The most stringent barrier seems to come from [5, Theorem 6], who proved that Betti number estimation for general complexes is DQC1-hard when  $\varepsilon, \gamma = 1/\text{poly}(n)$ , where DQC1 is a complexity class that is expected to be hard to simulate classically<sup>5</sup>. This safeguards a quantum speedup for the case of general complexes, yet it leaves open the case of clique complexes. Our work shows that we can get additional leverage for clique complexes. We leave it as our main open question whether the classical complexity for clique complexes can be improved to  $\text{poly}(n, 1/\gamma, 1/\varepsilon)$ .

<sup>3</sup>The up-degree of a  $k$ -face is the number of  $(k+1)$ -faces that contain the  $k$ -face (see Definition 2.1).

<sup>4</sup>More precisely, the “dequantized quantum singular value transformation algorithm” in Section 4 of [14].

<sup>5</sup>In fact, they consider a slight generalization of the problem called “quasi-Betti number estimation”.

The task of estimating persistent Betti numbers has been getting an increasing amount of attention, see e.g. [17, 21, 24]. It seems like our method can be used for solving this problem too (if we have membership query access to both complexes  $K$  and  $L$  with  $K \subseteq L$ ), but we leave the formal proof of this for future work. However, note that we lose the speedup in the clique complex case, because for the persistent Laplacian it is no longer true that it can only have less than  $n$  nonzero entries per row (see our Lemma 4.3). For example, if  $K$  only consists of  $n$  isolated vertices, but  $L$  is the clique complex corresponding to the complete graph, then for any  $k$  the  $k$ -th persistent Laplacian of the pair  $(K, L)$  has  $\Omega(nk)$  nonzero entries in each row.

A final open question, as was already mentioned in earlier works [4], is characterizing which complexes admit a large spectral gap. The advantage of our algorithm, as well as the aforementioned quantum algorithms, hinges on this assumption. We note that [4] discussed the complete  $k$ -partite graph  $K(m, k)$  as an example where our spectral gap assumption on the combinatorial Laplacian holds.  $K(m, k)$  consists of  $k$  clusters, where each cluster contains  $m$  vertices, and two vertices are adjacent iff they are in different clusters. The  $(k - 1)$ -th combinatorial Laplacian of the clique complex defined by  $K(m, k)$  has spectral gap  $m$  and the  $(k - 1)$ -th Betti number of the complex is  $(m - 1)^k$  (see [4, Proposition 1 & 2]). Observe that for  $K(n/k, k)$ , the spectral gap is  $n/k$ , and the normalized spectral gap is  $\gamma = 1/k$ . Thus, for this kind of complexes our algorithm runs in polynomial time if for example  $\varepsilon \in \Omega(1)$  and  $k \in O(\log^2(n))$ .

**Related work.** Upon completion of this work, we noticed that a similar classical path integral Monte Carlo algorithm for estimating Betti numbers was proposed in [4]. The authors use a Trotterization approach to implement an imaginary time evolution of the combinatorial Laplacian, and use a more complex distribution over paths to minimize the variance of the Monte Carlo estimator. These techniques seem more flexible than ours, and might eventually lead to a better algorithm. However, unless some additional conditions are imposed, the current runtime of their algorithm still shows an exponential dependency on both  $k$  and  $1/\varepsilon$ , which our algorithm avoids.

As a reviewer has pointed out, a similar usage of the path integral Monte Carlo method to estimate elements of a matrix power is present in [8] (in particular, see their Lemma 2.5).

## 2 Notations and Preliminaries

### 2.1 Homology and Laplacians

In this work, we consider a simplicial complex  $K$  over the set  $[n]$ , which corresponds to a downward closed set system over  $[n]$ , where  $[n] = \{1, \dots, n\}$ . We denote by  $C_k \subseteq K$  the set of subsets  $A = \{i_1, \dots, i_{k+1}\} \subseteq [n]$  of cardinality  $k + 1$ , also called the  $k$ -faces of  $K$ , and  $d_k = |C_k|$ . The *dimension* of  $K$  is the largest integer  $k$  such that  $K$  contains at least one  $k$ -face, that is,  $\dim(K) = \max\{k \text{ s.t. } d_k > 0\}$ . We *orient* the faces by ordering the vertices in increasing order (that is, we assume  $i_\ell < i_{\ell+1}$ ). Once we have defined an orientation of the faces, we associate to each face a basis vector  $|A\rangle := |i_1 \dots i_{k+1}\rangle$ . The *boundary operator*  $\partial_k : C_k \mapsto C_{k-1}$  is defined by

$$\partial_k |A\rangle = \sum_{\ell=1}^{k+1} (-1)^{\ell-1} |A \setminus \{i_\ell\}\rangle.$$

From the boundary operator we can define the *combinatorial Laplacian*  $\Delta_k : C_k \mapsto C_k$  as

$$\Delta_k = \Delta_k^\downarrow + \Delta_k^\uparrow,$$

with  $\Delta_k^\downarrow = \partial_k^\dagger \partial_k$  and  $\Delta_k^\uparrow = \partial_{k+1} \partial_{k+1}^\dagger$ .

$\Delta_k$  is a symmetric, positive semidefinite matrix of size  $d_k \times d_k$ . Next we define notions of degree and neighborhood of a face.

**Definition 2.1.** In a simplicial complex, the *up-degree* of a  $k$ -face  $i$  is the number of  $(k+1)$ -faces that contain  $i$ . It is denoted as  $d_i^{\text{up}} := |\{j \in C_{k+1} \text{ s.t. } i \subseteq j\}|$ . The maximum up-degree among all the  $k$ -faces is denoted as  $\delta_k = \max_{i \in C_k} d_i^{\text{up}}$ .

**Definition 2.2** (Down-up and up-down neighbors). Let  $i, j \in C_k$  be two  $k$ -faces of a simplicial complex  $K$ .  $i$  and  $j$  are said to be *down-up neighbors* if their symmetric difference  $|i \Delta j| = 2$ . Additionally, if  $i \cup j$  is a  $(k+1)$ -face of  $K$ , then  $i$  and  $j$  are also said to be *up-down neighbors*.

The following lemma from [15] uses these notions to characterize the entries of  $\Delta_k$ .

**Lemma 2.3** (Restatement of Laplacian Matrix Theorem, [15, Theorem 3.3.4]). *Let  $K$  be a finite oriented simplicial complex,  $k$  be an integer with  $0 < k \leq \dim(K)$ , and  $\{1, 2, \dots, d_k\}$  denote the  $k$ -faces of  $K$ . Let  $i, j \in [d_k]$ . Then we have the following:*

- $(\Delta_k)_{ii} = d_i^{\text{up}} + k + 1$ .
- $(\Delta_k)_{ij} = \pm 1$  if  $i \neq j$ , and  $i$  and  $j$  are down-up neighbors but they are not up-down neighbors.
- $(\Delta_k)_{ij} = 0$  otherwise (i.e. if  $i \neq j$ , and either  $i$  and  $j$  are up-down neighbors or they are not down-up neighbors).

The following lemma gathers some useful facts about  $\Delta_k$  that will be used in our proofs.

**Lemma 2.4.** *Let us consider a simplicial complex  $K$  with  $\Delta_k$  being its combinatorial Laplacian and  $\delta_k$  being the maximum up-degree among all  $k$ -faces of  $K$ . Then the following results hold:*

- $\delta_k + k + 1 \leq \lambda_{\max}(\Delta_k) \leq n$ .
- $(\Delta_k)_{ii} \leq n$ .
- $\Delta_k$  has at most  $(n - k - 1)(k + 1)$  nonzero off-diagonal entries in each row, all equal to  $\pm 1$ <sup>6</sup>.

*Proof.* The second and third bullets follow from Lemma 2.3. The second inequality of the first bullet follows from [10, Proposition 6.2], who prove that  $\lambda_{\max}(\Delta_k^\uparrow), \lambda_{\max}(\Delta_k^\downarrow) \leq n$ . This gives the claimed bound if we use that  $\lambda_{\max}(\Delta_k) = \max\{\lambda_{\max}(\Delta_k^\uparrow), \lambda_{\max}(\Delta_k^\downarrow)\}$ , which follows from  $\Delta_k^\uparrow \Delta_k^\downarrow = \Delta_k^\downarrow \Delta_k^\uparrow = 0$ .

---

<sup>6</sup>This is tight up to a constant for general simplicial complexes, in contrast to some earlier papers [16, 20] that mention  $\mathcal{O}(n)$  nonzero off-diagonal entries. Later in the paper (see Lemma 4.3) we show that for clique complexes it is actually  $\mathcal{O}(n)$ , and we exploit this to design a more efficient algorithm.

For proving the first inequality of the first bullet, we write the largest eigenvalue using the Rayleigh quotient:

$$\begin{aligned}\lambda_{\max}(\Delta_k) &= \max_{\|x\|_2=1} x^T \Delta_k x = \max_{\|x\|_2=1} (x^T \Delta_k^\dagger x + x^T \Delta_k^\downarrow x) \\ &= \max_{\|x\|_2=1} (\|\partial_{k+1}^\dagger x\|_2^2 + \|\partial_k x\|_2^2).\end{aligned}$$

We can lower bound this by taking a particular  $x$ : the one that is all zero except for a position where it is one, and the latter position corresponds to a  $k$ -face with up-degree  $\delta_k$ . For this vector,  $\partial_{k+1}^\dagger x$  contains  $\delta_k$  ones and the other elements are zero (because it has up-degree  $\delta_k$ ). And it is also true that  $\partial_k x$  contains  $k+1$  ones and the other elements are zero (because every  $k$ -face contains  $k+1$  many  $(k-1)$ -faces). This concludes that  $\delta_k + k + 1 \leq \lambda_{\max}(\Delta_k)$ .  $\square$

Now let us define the central objects of our interest, namely the *Betti numbers* of a simplicial complex.

**Definition 2.5** (Betti number). Let  $\Delta_k$  be the  $k$ -th combinatorial Laplacian of a simplicial complex  $K$  with  $\partial_k$  denoting the boundary operator. The  $k$ -th Betti number  $\beta_k$  equals the dimension of the  $k$ -th homology group:  $\beta_k = \dim(\ker(\partial_k)/\text{Im}(\partial_{k+1}))$ . Equivalently,  $\beta_k = \dim \ker(\Delta_k)$ , thus it is equal to the number of zero eigenvalues of  $\Delta_k$ .

We will be particularly interested in the Betti numbers of *clique complexes*. The clique complex  $K$  associated to a graph  $G = (V, E)$ , with vertices  $V = [n]$  and (undirected) edges  $E$ , is the family of subsets of  $V$  that are cliques in  $G$ , that is, a subset  $A \subseteq V$  is in  $K$  if and only if  $A$  induces a clique in  $G$ .

## 2.2 Hoeffding's inequality

The following concentration bound will be crucial in our proofs.

**Lemma 2.6** (Hoeffding's Inequality [18], see also [9]). *Let  $X_1, \dots, X_p$  be independent random variables such that  $a \leq X_i \leq b$  and let  $X = \frac{1}{p} \sum_{i=1}^p X_i$ . Then, for all  $\delta > 0$ ,*

$$\mathbb{P}(|X - \mathbb{E}[X]| \geq \delta) \leq 2 \exp\left(\frac{-2p\delta^2}{(b-a)^2}\right).$$

## 3 Algorithm for general simplicial complexes

Consider a general simplicial complex  $K$  over  $[n]$ , with  $d_k$  being the number of its  $k$ -faces,  $\Delta_k$  its  $k$ -th combinatorial Laplacian, and with  $k$ -th Betti number  $\beta_k$ . We wish to obtain an estimate  $\tilde{\nu}_k$  that satisfies  $\tilde{\nu}_k = \beta_k/d_k \pm \varepsilon$  for some parameter  $\varepsilon \in (0, 1)$ . In this section, we make the following assumptions:

1. In time polynomial in  $n$ , (a) we can check whether a set is in the complex, and (b) we can sample a  $k$ -face from the simplicial complex  $K$  uniformly at random <sup>7</sup>.

---

<sup>7</sup>We do not need assumption (b) if the complex is dense in  $k$ -faces.

2. We have estimates  $\hat{\lambda}$  and  $\gamma$  on the largest eigenvalue and the spectral gap of  $\Delta_k$ , respectively, satisfying

$$\lambda_{\max}(\Delta_k) \leq \hat{\lambda} \leq c\lambda_{\max}(\Delta_k) \quad \text{and} \quad \lambda_2(\Delta_k) \geq \gamma\hat{\lambda},$$

for some constant  $c > 0$ <sup>8</sup>.

Now consider the matrix

$$H = I - \Delta_k/\hat{\lambda},$$

which, as we discuss below, satisfies  $0 \preceq H \preceq I$ . From Lemma 2.4, we know that  $0 \leq (\Delta_k)_{ii} \leq n$  and  $\Delta_k$  has  $\mathcal{O}(nk)$  nonzero off-diagonal entries in every row, each of absolute magnitude 1. This implies that

$$\|H\|_1 = \max_j \sum_i |H_{ij}| \in \mathcal{O}(nk).$$

By construction, the  $k$ -th combinatorial Laplacian  $\Delta_k$  is positive semidefinite, hence all eigenvalues are non-negative,  $\beta_k$  of them are equal to 0, the second smallest distinct eigenvalue is  $\lambda_2(\Delta_k)$ , and the maximum eigenvalue is  $\lambda_{\max}(\Delta_k)$ . Thus, by linearity, the eigenvalues of  $H$  lie between 0 and 1,  $\beta_k$  of them equal to 1, and all other eigenvalues lie below  $1 - \lambda_2(\Delta_k)/\hat{\lambda} \leq 1 - \gamma$ . The following lemma shows how to relate the trace of  $H^r$ , for sufficiently large  $r$ , to the Betti number  $\beta_k$ .

**Lemma 3.1.** *If  $r \geq \frac{1}{\gamma} \log \frac{1}{\varepsilon}$  then  $\beta_k \leq \text{Tr}(H^r) \leq \beta_k + \varepsilon d_k$ .*

*Proof.* On the one hand, we have that

$$\text{Tr}(H^r) = \sum_{i=1}^{d_k} \lambda_i(H)^r \geq \beta_k.$$

On the other hand, we have that

$$\text{Tr}(H^r) = \sum_{i=1}^{d_k} \lambda_i(H)^r \leq \beta_k + \sum_{i: \lambda_i(H) < 1} (1 - \gamma)^r \leq \beta_k + \varepsilon d_k,$$

where we used that  $(1 - \gamma)^r \leq \varepsilon$  for  $r \geq \frac{1}{\gamma} \log \frac{1}{\varepsilon}$ . □

Using this observation, we can obtain a  $2\varepsilon$ -additive estimate of  $\beta_k/d_k$  from an  $\varepsilon$ -additive estimate of  $\text{Tr}(H^r)/d_k$ . To obtain the latter we use another observation, that holds not only for large  $r$  as above, but for a general nonnegative  $z$ -th power of  $H$ .

**Observation 3.2.**

$$\frac{1}{d_k} \text{Tr}(H^z) = \frac{1}{d_k} \sum_{i=1}^{d_k} \langle i | H^z | i \rangle = \mathbb{E}_i \left[ X_z^{(i)} \right],$$

where  $i \in [d_k]$  is sampled uniformly at random and  $X_z^{(i)} = \langle i | H^z | i \rangle$ .

---

<sup>8</sup>If we do not have such a bound on the spectral gap, an alternative is to approximate the “quasi-Betti number”, which is the number of small eigenvalues (below  $\gamma\hat{\lambda}$ ) of the combinatorial Laplacian, as in [5].

Since  $H$  is  $\mathcal{O}(nk) \in \mathcal{O}(n^2)$ -sparse, we can evaluate  $X_z^{(i)}$  exactly in time  $\mathcal{O}(n^{2z})$ . Indeed this is the approach in [14]. Here we use another approach based on the path integral Monte Carlo method, which has two advantages. First, it improves the space complexity from  $n^{\mathcal{O}(z)}$ , as in [14], to  $\tilde{\mathcal{O}}(nz)$ . Second, it will lead to a faster algorithm for clique complexes (see next section).

Let us denote the sign of  $H_{i,j}$  by  $(-1)^{s(i,j)}$ , with  $s(i,j) \in \{0,1\}$ . We can rewrite  $X_z^{(i)}$  as follows:

$$\begin{aligned} X_z^{(i)} &= \langle i | H^z | i \rangle \\ &= \sum_{j_1, \dots, j_z} \langle i | j_z \rangle \langle j_z | H | j_{z-1} \rangle \dots \langle j_1 | H | i \rangle \\ &= \sum_{j_1, \dots, j_z} Y_z(i, j_1, \dots, j_z) \frac{|H_{j_z, j_{z-1}}|}{\|H_{\cdot, j_{z-1}}\|_1} \dots \frac{|H_{j_1, i}|}{\|H_{\cdot, i}\|_1}, \end{aligned}$$

with

$$Y_z(j_0, j_1, \dots, j_z) = \langle j_0 | j_z \rangle \prod_{\ell=0}^{z-1} (-1)^{s(j_{\ell+1}, j_\ell)} \|H_{\cdot, j_\ell}\|_1.$$

By Lemma 2.4 it holds that  $|Y_z| \leq \|H\|_1^z \leq (n + nk)^z \in \mathcal{O}(n^{2z})$ . By our choice of normalization, we can interpret  $|H_{j,i}|/\|H_{\cdot, i}\|_1 =: P(i, j)$  as a transition probability from face  $i$  to face  $j$ . We can then say that

$$X_z^{(i)} = \mathbb{E}_{(j_0=i, j_1, \dots, j_z)} [Y_z(j_0, j_1, \dots, j_z)],$$

where the path  $(j_0, j_1, \dots, j_z)$  is drawn with probability  $P(j_0, j_1) \dots P(j_{z-1}, j_z)$  from the resulting Markov chain with transition matrix  $P$ . Moreover, if we choose the initial  $k$ -face  $j_0 \in [d_k]$  uniformly at random, then

$$\begin{aligned} \mathbb{E}[Y_z] &= \mathbb{E}_{(j_0, j_1, \dots, j_z)} [Y_z(j_0, j_1, \dots, j_z)] \\ &= \mathbb{E}_{j_0} [X_z^{(j_0)}] = \frac{1}{d_k} \text{Tr}(H^z). \end{aligned}$$

This gives us an unbiased estimator  $Y_z$  for the normalized trace of  $H^z$ . Moreover, as proven in the following lemma, we can sample  $Y_z$  efficiently.

**Lemma 3.3.** *We can sample from  $Y_z$ , as defined above, in time  $z \cdot \text{poly}(n)$ .*

*Proof.* We can evaluate  $Y_z$  by sampling  $z$  steps of the Markov chain over  $k$ -faces. The initial  $k$ -face  $j_0$  is drawn uniformly at random. By our assumptions, we can do this in time  $\text{poly}(n)$ . Subsequent steps are sampled as follows.

Let  $j_i$  be the current  $k$ -face. First we learn the up-degree  $d_{j_i}^{up}$ , and hence  $(\Delta_k)_{j_i j_i}$ . We do this by, for all potential up-neighbors (obtained by adding one element to the face), querying whether they are in the complex. This takes  $n - k - 1$  queries, and hence time  $\text{poly}(n)$ . Then we learn all down-up neighbors by querying all  $\mathcal{O}(n^2)$  subsets with symmetric difference 2. This again takes time  $\text{poly}(n)$ . By Lemma 2.3 we can now derive all  $\mathcal{O}(n^2)$  nonzero entries of the  $j_i$ -th row  $H_{\cdot, j_i}$ , and hence sample  $j_{i+1}$  according to the probability  $P(j_{i+1}, j_i) = |H_{j_{i+1}, j_i}|/\|H_{\cdot, j_i}\|_1$ . This yields time  $\text{poly}(n)$  per step of the Markov chain, and so  $z \cdot \text{poly}(n)$  time overall.  $\square$

It remains to bound the complexity of estimating  $\mathbb{E}[Y_z]$ , given samples of  $Y_z$ . For this, we use Hoeffding's inequality (Lemma 2.6), which yields the following lemma.

**Lemma 3.4.** For any  $\delta > 0$  and integer  $z \geq 0$ , we can obtain a  $\delta$ -additive estimate of  $\mathbb{E}[Y_z] = \text{Tr}(H^z)/d_k$  by taking the average of  $\mathcal{O}(n^{4z})/\delta^2$  many independent samples of  $Y_z$ .

*Proof.* We know that  $|Y_z| \leq \|H\|_1^z \leq (n + nk)^z \in \mathcal{O}(n^{2z})$  (Lemma 2.4). Consider  $p$  independent samples  $Y_{z,1}, \dots, Y_{z,p}$  distributed according to  $Y_z$ . For any  $\delta > 0$ , Hoeffding's inequality (Lemma 2.6) states that

$$\Pr \left( \left| \frac{1}{p} \sum_{i=1}^p Y_{z,i} - \mathbb{E}[Y_z] \right| \geq \delta \right) \leq 2 \exp \left( \frac{-2p\delta^2}{\mathcal{O}(n^{4z})} \right).$$

If we choose  $p = \mathcal{O}(n^{4z})/\delta^2$  then  $\frac{1}{p} \sum Y_{z,i}$  will be  $\delta$ -close to its expectation  $\mathbb{E}[Y_z] = \frac{1}{d_k} \text{Tr}(H^z)$  with probability at least  $1 - 1/2^{\text{poly}(n)}$ .  $\square$

This leads to the following algorithm, which has time complexity  $\mathcal{O}(n^{4z}/\delta^2)$ .

---

**Algorithm 1:** Algorithm for  $\delta$ -estimating  $\text{Tr}(H^z)/d_k = \text{Tr} \left( (I - \Delta_k/\hat{\lambda})^z \right) / d_k$

---

**Input:** Query and sample access to complex  $K$ , integer  $k$ , parameters  $\hat{\lambda}$  and  $z$ , precision parameter  $\delta \in (0, 1)$ .

**Output:** Estimate  $\text{est}_{k,z}$  such that  $\text{est}_{k,z} = \text{Tr}(H^z)/d_k \pm \delta$  with high probability.

- 1 Set  $p = \mathcal{O}(n^{4z})/\delta^2$ .
  - 2 **for**  $t = 1, \dots, p$  **do**
  - 3     Sample a  $k$ -face  $j_0$  of  $K$  uniformly at random.
  - 4     Sample  $z$  steps  $(j_0, j_1, \dots, j_z)$  of the Markov chain  $P$  with initial face  $j_0$ .
  - 5     Set  $Y_{z,t} = \langle j_0 | j_z \rangle \prod_{q=0}^{z-1} (-1)^{s(j_{q+1}, j_q)} \|H_{\cdot, j_q}\|_1$ .
  - 6 **Return**  $\text{est}_{k,z} = \frac{1}{p} \sum_{t=1}^p Y_{z,t}$ .
- 

For  $r \geq \frac{1}{\gamma} \log \frac{2}{\varepsilon}$ , we know from Lemma 3.1 that  $\text{Tr}(H^r)/d_k = \beta_k/d_k \pm \varepsilon/2$ . Hence, setting  $\delta = \varepsilon/2$  and  $z = r$  in the algorithm above we get an  $\varepsilon$ -additive estimate of  $\beta_k/d_k$ . The algorithm requires  $p = \mathcal{O}(n^{4r}/\delta^2) = n^{\mathcal{O}(\frac{1}{\gamma} \log \frac{1}{\varepsilon})}$  samples of  $Y_r$ , each of which can be obtained in time  $r \cdot \text{poly}(n)$  by Lemma 3.3. The overall time complexity of Algorithm 1 is hence  $n^{\mathcal{O}(\frac{1}{\gamma} \log \frac{1}{\varepsilon})}$ .

### 3.1 Improvement using Chebyshev polynomials

We can slightly improve this result by approximating  $H^r$  with a polynomial of degree roughly  $\sqrt{r}$  using Chebyshev polynomials, and then estimating the monomials using Algorithm 1. We use the following approximation lemma.

**Lemma 3.5** (Follows from [22, Theorem 3.3]). For any  $\delta > 0$  and  $d \geq \sqrt{2r \log(2/\delta)}$ , the monomial  $x^r$  can be approximated by a polynomial  $p_{r,d}(x)$  of degree  $d$  such that  $|p_{r,d}(x) - x^r| \leq \delta$  for all  $x \in [-1, 1]$ .

Now we bound the size of the monomial coefficients in  $p_{r,d}(x)$ , as these will govern the precision with which we need to estimate the trace of each monomial. Following [22, Chapter 3], the polynomial  $p_{r,d}(x)$  is obtained by first approximating  $x^r$  in the Chebyshev basis by

$$p_{r,d}(x) = \alpha_0^{(r)} + \sum_{i=1}^d 2\alpha_i^{(r)} T_i(x)$$

where  $T_i(x)$  is the  $i$ 'th Chebyshev polynomial (of the first kind), and  $\alpha_i^{(r)} = \binom{r}{(r-i)/2}/2^r$  if  $i$  has the same parity as  $r$ , and  $\alpha_i^{(r)} = 0$  otherwise. We then obtain the desired coefficients of  $p_{r,d}$  in the monomial basis by expressing each of the Chebyshev polynomials in the monomial basis. Concretely, if  $T_i(x) = \sum_{\ell=0}^i c_\ell^{(i)} x^\ell$  then

$$\begin{aligned} p_{r,d}(x) &= \alpha_0^{(r)} + \sum_{i=1}^d 2\alpha_i^{(r)} T_i(x) \\ &= \alpha_0^{(r)} + \sum_{\ell=0}^d \left[ \sum_{i=\ell}^d 2\alpha_i^{(r)} c_\ell^{(i)} \right] x^\ell =: \sum_{\ell=0}^d b_\ell^{(r,d)} x^\ell. \end{aligned}$$

To bound the coefficients  $b_\ell^{(r,d)}$ , we first bound the coefficients  $c_\ell^{(i)}$  in Lemma 3.6 below. Combined with the bounds  $|\alpha_i^{(r)}| \leq 1$  this lemma yields the upper bound  $|b_\ell| \leq (d+1) \cdot 2 \cdot 2^{2d} \leq 2^{3d}$ .

**Lemma 3.6.** *For all  $i \in \mathbb{N}$  and  $\ell \leq i$ , we have  $|c_\ell^{(i)}| \leq 2^{2i}$ .*

*Proof.* We prove the lemma using induction on  $i$ . The Chebyshev polynomials can be defined via the following recurrence

$$T_i(x) = 2xT_{i-1}(x) - T_{i-2}(x),$$

where  $T_0(x) = 1$ ,  $T_1(x) = x$ . This immediately shows that the bounds  $|c_\ell^{(i)}| \leq 2^{2i}$  hold for  $i = 0$  and  $i = 1$ . Now assume  $i > 1$  and that for all  $i' < i$  and  $\ell \leq i'$ , we have  $|c_\ell^{(i')}| \leq 2^{2i'}$ . Then from the recursion  $T_i(x) = 2xT_{i-1}(x) - T_{i-2}(x)$ , we obtain  $c_\ell^{(i)} = 2c_{\ell-1}^{(i-1)} - c_\ell^{(i-2)}$  and hence  $|c_\ell^{(i)}| \leq 2^{2(i-1)+1} + 2^{2(i-2)} \leq 2 \cdot 2^{2(i-1)+1} = 2^{2i}$ .  $\square$

Now we can describe an efficient algorithm that, for any  $\varepsilon > 0$ , outputs an additive  $\varepsilon$ -estimate of  $\beta_k/d_k$  with high probability. The correctness and complexity of the algorithm are proven in Theorem 3.7.

---

**Algorithm 2:** Algorithm for  $\varepsilon$ -estimating  $\beta_k/d_k$

---

**Input:** Query and sample access to complex  $K$ , integer  $k$ , estimates of  $\hat{\lambda}$  and  $\gamma$ , precision parameter  $\varepsilon \in (0, 1)$ .

**Output:** Estimate  $\tilde{v}_k$  such that  $\tilde{v}_k = \beta_k/d_k \pm \varepsilon$  with high probability.

- 1 Set  $r = \lceil \frac{1}{\gamma} \log \frac{3}{\varepsilon} \rceil$  and  $d = \lceil \sqrt{\frac{2}{\gamma}} \log \frac{6}{\varepsilon} \rceil$ .
  - 2 **for**  $\ell = 0, \dots, d$  **do**
  - 3     Estimate  $\text{Tr}(H^\ell)/d_k$  to additive precision  $\delta = \varepsilon / (3(d+1)2^{3d})$  with high probability using Algorithm 1. Let  $\text{est}_{k,\ell}$  denote the output.
  - 4 Return  $\tilde{v}_k = \sum_{\ell=0}^d b_\ell^{(r,d)} \text{est}_{k,\ell}$ .
- 

**Theorem 3.7.** *Algorithm 2 returns with high probability an estimate of  $\beta_k/d_k$  with additive error  $\varepsilon$  in time  $n^{\mathcal{O}\left(\frac{1}{\sqrt{\gamma}} \log \frac{1}{\varepsilon}\right)}$ .*

*Proof.* First we prove the correctness. By our choice of  $r$ , we know from Lemma 3.1 that  $\text{Tr}(H^r)/d_k = \beta_k/d_k \pm \varepsilon/3$ , so it suffices to return an  $(2\varepsilon/3)$ -additive estimate of  $\text{Tr}(H^r)/d_k$ . By Lemma 3.5 we can use the approximation

$$\frac{1}{d_k} \text{Tr}(H^r) = \frac{1}{d_k} \sum_{\ell=0}^d b_\ell^{(r,d)} \text{Tr}(H^\ell) \pm \varepsilon/3$$

for  $d = \left\lceil \sqrt{2r \log \frac{6}{\varepsilon}} \right\rceil \leq \left\lceil \sqrt{\frac{2}{\gamma}} \log \frac{6}{\varepsilon} \right\rceil$ . We estimate each term  $\text{Tr}(H^\ell)/d_k$  to precision  $\delta = \frac{\varepsilon}{3(d+1)2^{3d}}$  with high probability, so that the final estimator has a total error

$$\begin{aligned} \tilde{v}_k &= \sum_{\ell=0}^d b_\ell^{(r,d)} \text{est}_{k,\ell} = \sum_{\ell=0}^d b_\ell^{(r,d)} \left( \text{Tr}(H^\ell)/d_k \pm \delta \right) \\ &= \left( \frac{1}{d_k} \sum_{\ell=0}^d b_\ell^{(r,d)} \text{Tr}(H^\ell) \right) \pm \varepsilon/3, \end{aligned}$$

using that  $\left| \sum_{\ell=0}^d b_\ell^{(r,d)} \delta \right| \leq \delta(d+1)2^{3d} \leq \varepsilon/3$ . Combined with the previous error bounds this shows that  $\tilde{v}_k = \beta_k/d_k \pm \varepsilon$  with high probability.

To bound the time complexity, recall that the time complexity of Algorithm 1 in Line 3 is  $\mathcal{O}(n^{4\ell}/\delta^2) \in n^{\mathcal{O}(d)}/\varepsilon^2$ . Summing over the  $d+1$  loops, and using the expression for  $d$ , this yields a total time complexity that is  $n^{\mathcal{O}\left(\frac{1}{\sqrt{r}} \log \frac{1}{\varepsilon}\right)}$ .  $\square$

This completes the proof of the first item of Theorem 1.1.

## 4 Algorithm for clique complexes

The complexity of our path integral Monte Carlo algorithm is dominated by the sample complexity that follows from Hoeffding's inequality (Lemma 2.6), which we bound using the fact that  $|Y_z| \leq \|H\|_1^z$  and  $\|H\|_1 = \text{poly}(n)$ . Here we prove a tighter bound on  $\|H\|_1$  for the special case of clique complexes, and exploit this to improve the algorithm.

We will use the following characterization of the off-diagonal elements of the combinatorial Laplacian  $\Delta_k$ .

**Lemma 4.1** (Follows from Lemma 2.3). *Let  $\Delta_k$  denote the  $k$ -th combinatorial Laplacian of a simplicial complex  $K$ . Then  $(\Delta_k)_{ij}$  for  $i \neq j$  is nonzero if and only if the corresponding two  $k$ -faces are down-up neighbors but not up-down neighbors.*

The following claim is going to be useful for the proof of the next lemma.

**Claim 4.2.** *In a clique complex, every  $k$ -face has at most  $n - k - 1$  down-up neighbors that are not its up-down neighbors.*

*Proof.* Since we are in the clique complex case, a  $k$ -face is exactly a  $(k+1)$ -clique, so we will use the two expressions interchangeably. We will prove a slightly stronger statement: every vertex that is not in a  $(k+1)$ -clique  $C$  can appear in at most one down-up neighbor of  $C$  that is not its up-down neighbor. Let us consider a  $(k+1)$ -clique  $C$  and suppose there is a vertex  $v$  among the  $n - k - 1$  vertices that are not in  $C$  such that  $v$  belongs to two distinct down-up neighbors  $C_1$  and  $C_2$  of  $C$ . That is, suppose there are two distinct vertices  $u_1, u_2 \in C$  such that  $C_1 = C \setminus \{u_1\} \cup \{v\}$  and  $C_2 = C \setminus \{u_2\} \cup \{v\}$  are  $(k+1)$ -cliques. We show that  $C_1$  and  $C_2$  are up-down neighbors of  $C$ . Indeed,  $v$  must be adjacent to every

vertex of  $C$ : from  $C_1 \in K$  it is adjacent to all vertices in  $C$  other than the vertex  $u_1$ , and from  $C_2 \in K$  it is adjacent all vertices except for  $u_2$ . So  $C \cup \{v\}$  forms a  $(k+2)$ -clique and hence  $C_1$  and  $C_2$  are up-down neighbors of  $C$ .  $\square$

This section's main observation is the following.

**Lemma 4.3.** *The  $k$ -th combinatorial Laplacian of a clique complex has at most  $n - k - d_i^{\text{up}}$  nonzero entries in every row, that is,*

$$|\{j : (\Delta_k)_{ij} \neq 0\}| \leq n - k - d_i^{\text{up}} \quad \forall i \in d_k$$

where  $d_i^{\text{up}}$  is the up-degree of the  $k$ -face corresponding to the  $i$ -th row of the combinatorial Laplacian.

*Proof.* As a consequence of Claim 4.2, every  $k$ -face has at most  $n - k - 1$  down-up neighbors that are not its up-down neighbors. Following Lemma 4.1, these elements correspond exactly to the nonzero off-diagonal entries in  $\Delta_k$ . Adding the diagonal element  $(\Delta_k)_{ii}$ , we obtain that the total number of nonzero entries in a row of the  $k$ -th combinatorial Laplacian is at most  $n - k$ .

To improve this bound, notice that if a vertex  $v$  is adjacent to all the vertices of a  $k$ -face (i.e, we have an up-neighboring  $(k+1)$ -face), then  $v$  cannot be in any down-up neighbor that is not an up-down neighbor as well. Thus, using Lemma 4.1 again, we can say that every up-neighbor ‘‘cancels’’ the corresponding down-up neighbors in  $\Delta_k$ .

Hence, if the  $k$ -face corresponding to the  $i$ -th row of the combinatorial Laplacian has up-degree  $d_i^{\text{up}}$ , then the number of nonzero entries in the  $i$ -th row of  $\Delta_k$  is at most  $n - k - d_i^{\text{up}}$ .  $\square$

From this, we get the following corollary.

**Corollary 4.4.**  $\|H\|_1 \leq 2n/\hat{\lambda}$ .

*Proof.* Recall that  $H = I - \Delta_k/\hat{\lambda}$ . Thus,

$$\begin{aligned} \|H\|_1 &= \max_j \sum_i |H_{ij}| \\ &\leq \max_j \left| (I)_{jj} - \frac{(\Delta_k)_{jj}}{\hat{\lambda}} \right| + n \cdot \frac{1}{\hat{\lambda}} \leq 2\frac{n}{\hat{\lambda}}, \end{aligned}$$

where for the first inequality we used the fact that  $|(\Delta_k)_{ij}|$  is either 1 or 0 if  $i \neq j$ , and by Lemma 4.1 it is 1 at most  $n$  times in every row or column. In the second inequality we used the fact that  $0 \leq (\Delta_k)_{ii} \leq n$ . Combining, we have the result.  $\square$

Now let us recall the path integral estimator  $Y_z$  as defined in the previous section:

$$Y_z(j_0, j_1, \dots, j_z) = \langle j_0 | j_z \rangle \prod_{\ell=0}^{z-1} (-1)^{s(j_{\ell+1}, j_\ell)} \|H_{\cdot, j_\ell}\|_1.$$

As a consequence of Corollary 4.4, it satisfies

$$|Y_z| \leq \|H\|_1^z \leq \left( \frac{2n}{\hat{\lambda}} \right)^z.$$

This improves the sample complexity in Lemma 3.4 from  $\mathcal{O}(n^{4z})/\delta^2$  to  $\mathcal{O}\left(\left(\frac{2n}{\lambda}\right)^{2z}\right) \cdot \frac{1}{\delta^2}$  that is  $\left(\frac{n}{\lambda}\right)^{\mathcal{O}(z)} \cdot \frac{1}{\delta^2}$ . This directly propagates to Algorithm 2, improving its time complexity from

$$n^{\mathcal{O}\left(\frac{1}{\sqrt{\gamma}} \log \frac{1}{\varepsilon}\right)} \text{ to } \left(\frac{n}{\lambda}\right)^{\mathcal{O}\left(\frac{1}{\sqrt{\gamma}} \log \frac{1}{\varepsilon}\right)} \cdot \text{poly}(n).$$

The  $\text{poly}(n)$  term comes from the time required for sampling (Lemma 3.3). This completes the proof of the second item of Theorem 1.1.

## Acknowledgements

This work benefited from discussions with Chris Cade, Marcos Crichigno, and Sevag Gharibian. Sayantan would like to thank Sophie Laplante for hosting him at IRIF, Paris, for an academic visit where this work was initiated. Finally, we thank anonymous reviewers for pointing out useful references and ways to improve our presentation.

## References

- [1] Ismail Yunus Akhalwaya, Shashanka Ubaru, Kenneth L Clarkson, Mark S Squillante, Vishnu Jejjala, Yang-Hui He, Kugendran Naidoo, Vasileios Kalantzis, and Lior Horesh. Exponential advantage on noisy quantum computers. *arXiv:2209.09371*, 2022. DOI: [10.48550/arXiv.2209.09371](https://doi.org/10.48550/arXiv.2209.09371).
- [2] Bernardo Amenyro, Vasileios Maroulas, and George Siopsis. Quantum persistent homology. *arXiv:2202.12965*, 2022. DOI: [10.48550/arXiv.2202.12965](https://doi.org/10.48550/arXiv.2202.12965).
- [3] J. A. Barker. A quantum-statistical Monte Carlo method; path integrals with boundary conditions. *The Journal of Chemical Physics*, 70(6):2914–2918, 1979. DOI: [10.1063/1.437829](https://doi.org/10.1063/1.437829).
- [4] Dominic W Berry, Yuan Su, Casper Gyurik, Robbie King, Joao Basso, Alexander Del Toro Barba, Abhishek Rajput, Nathan Wiebe, Vedran Dunjko, and Ryan Babbush. Analyzing prospects for quantum advantage in topological data analysis. *arXiv:2209.13581*, 2022. DOI: [10.48550/arXiv.2209.13581](https://doi.org/10.48550/arXiv.2209.13581).
- [5] Chris Cade and P Marcos Crichigno. Complexity of supersymmetric systems and the cohomology problem. *arXiv:2107.00011*, 2021. DOI: [10.48550/arXiv.2107.00011](https://doi.org/10.48550/arXiv.2107.00011).
- [6] Gunnar Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46(2):255–308, 2009. DOI: [10.1090/S0273-0979-09-01249-X](https://doi.org/10.1090/S0273-0979-09-01249-X).
- [7] Marcos Crichigno and Tamara Kohler. Clique homology is QMA1-hard. *arXiv:2209.11793*, 2022. DOI: [10.48550/arXiv.2209.11793](https://doi.org/10.48550/arXiv.2209.11793).
- [8] Dean Doron, Amir Sarid, and Amnon Ta-Shma. On approximating the eigenvalues of stochastic matrices in probabilistic logspace. *computational complexity*, 26:393–420, 2017. DOI: [10.1007/s00037-016-0150-y](https://doi.org/10.1007/s00037-016-0150-y).
- [9] Devdatt P Dubhashi and Alessandro Panconesi. *Concentration of measure for the analysis of randomized algorithms*. Cambridge University Press, 2009. DOI: [10.1017/CBO9780511581274](https://doi.org/10.1017/CBO9780511581274).
- [10] Art Duval and Victor Reiner. Shifted simplicial complexes are Laplacian integral. *Transactions of the American Mathematical Society*, 354(11):4313–4344, 2002. DOI: [10.1090/S0002-9947-02-03082-9](https://doi.org/10.1090/S0002-9947-02-03082-9).
- [11] Gábor Elek. Betti numbers are testable. In *Fete of combinatorics and computer science*, pages 139–149. Springer, 2010. DOI: [10.1007/978-3-642-13580-4\\_6](https://doi.org/10.1007/978-3-642-13580-4_6).

- [12] George E Forsythe and Richard A Leibler. Matrix inversion by a Monte Carlo method. *Mathematics of Computation*, 4(31):127–129, 1950. DOI: [10.1090/S0025-5718-1950-0038138-X](https://doi.org/10.1090/S0025-5718-1950-0038138-X).
- [13] Joel Friedman. Computing Betti numbers via combinatorial Laplacians. *Algorithmica*, 21(4):331–346, 1998. DOI: [10.1007/PL00009218](https://doi.org/10.1007/PL00009218).
- [14] Sevag Gharibian and François Le Gall. Dequantizing the quantum singular value transformation: Hardness and applications to quantum chemistry and the quantum PCP conjecture. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 19–32, 2022. DOI: [10.1145/3519935.3519991](https://doi.org/10.1145/3519935.3519991).
- [15] Timothy E Goldberg. Combinatorial Laplacians of simplicial complexes. *Senior Thesis, Bard College*, 2002.
- [16] Casper Gyurik, Chris Cade, and Vedran Dunjko. Towards quantum advantage via topological data analysis. *Quantum*, 6:855, 2022. DOI: [10.22331/q-2022-11-10-855](https://doi.org/10.22331/q-2022-11-10-855).
- [17] Ryu Hayakawa. Quantum algorithm for persistent Betti numbers and topological data analysis. *Quantum*, 6:873, 2022. DOI: [10.22331/q-2022-12-07-873](https://doi.org/10.22331/q-2022-12-07-873).
- [18] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. DOI: [10.1080/01621459.1963.10500830](https://doi.org/10.1080/01621459.1963.10500830).
- [19] Seth Lloyd, Silvano Garnerone, and Paolo Zanardi. Quantum algorithms for topological and geometric analysis of data. *Nature communications*, 7(1):1–7, 2016. DOI: [10.1038/ncomms10138](https://doi.org/10.1038/ncomms10138).
- [20] Sam McArdle, András Gilyén, and Mario Berta. A streamlined quantum algorithm for topological data analysis with exponentially fewer qubits. *arXiv:2209.12887*, 2022. DOI: [10.48550/arXiv.2209.12887](https://doi.org/10.48550/arXiv.2209.12887).
- [21] Facundo Mémoli, Zhengchao Wan, and Yusu Wang. Persistent laplacians: Properties, algorithms and implications. *SIAM Journal on Mathematics of Data Science*, 4(2): 858–884, 2022. DOI: [10.1137/21M1435471](https://doi.org/10.1137/21M1435471).
- [22] Sushant Sachdeva and Nisheeth K. Vishnoi. Faster algorithms via approximation theory. *Foundations and Trends® in Theoretical Computer Science*, 9(2):125–210, 2014. ISSN 1551-305X. DOI: [10.1561/04000000065](https://doi.org/10.1561/04000000065).
- [23] Alexander Schmidhuber and Seth Lloyd. Complexity-theoretic limitations on quantum algorithms for topological data analysis. *arXiv:2209.14286*, 2022. DOI: [10.48550/arXiv.2209.14286](https://doi.org/10.48550/arXiv.2209.14286).
- [24] Rui Wang, Duc Duy Nguyen, and Guo-Wei Wei. Persistent spectral graph. *International Journal for Numerical Methods in Biomedical Engineering*, 36(9):e3376, 2020. DOI: [10.1002/cnm.3376](https://doi.org/10.1002/cnm.3376).