Quantum Information and Computation, Vol. 23, No. 13&14 (2023) 1091–1104


# UNDERSTANDING THE MAPPING OF ENCODE DATA THROUGH AN IMPLEMENTATION OF QUANTUM TOPOLOGICAL ANALYSIS

ANDREW VLASIC
*Deloitte Consulting, LLP*
*Chicago, Illinois, 60606, US*

ANH PHAM
*Deloitte Consulting, LLP*
*Atlanta, Georgia, 30303, US*

Received September 7, 2023
Revised October 11, 2023

A potential advantage of quantum machine learning stems from the ability of encoding classical data into high dimensional complex Hilbert space using quantum circuits. Recent studies exhibit that not all encoding methods are the same when representing classical data since certain parameterized circuit structures are more expressive than the others. In this study, we show the difference in encoding techniques can be visualized by investigating the topology of the data embedded in complex Hilbert space. The technique for visualization is a hybrid quantum based topological analysis which uses simple diagonalization of the boundary operators to compute the persistent Betti numbers and the persistent homology graph. To augment the computation of Betti numbers within a NISQ framework, we suggest a simple hybrid algorithm. Through an illuminating example of a synthetic data set and the methods of angle encoding, amplitude encoding, and IQP encoding, we reveal topological differences between the encoding methods, as well as differences with the original data. Consequently, our results suggest the encoding method needs to be considered carefully within different quantum machine learning models since it can strongly affect downstream analysis like clustering or classification.

*Keywords*: Quantum Machine Learning; Data Encoding; Quantum Topological Data Analysis; Betti Numbers

## 1 Introduction

The power of quantum machine learning originates from the fact that classical data can be embedded in high-dimensional complex Hilbert space through the use of quantum circuits. Several strategies have been explored to encode classical data using different parameterized quantum circuits with different level of expressitivity to approximate the function representing classical data. For instance, Schuld et al. [1] has shown that using Fourier series analysis can reveal if a variational quantum circuit is expressive enough to represent different classical data structure. However, a question still remains if encoding classical data using quantum circuits preserves the original structure of the data or it will change this structure after embedding in complex Hilbert space.

As the applications of classical statistical modeling and machine learning modeling increase in popularity there has been a growing need of advanced exploratory data analysis to give

insight to the stability of the model through the various structures of the data. Topological data analysis (TDA) has become a stable of general techniques to understand the geometric structure of data and assist with these insights. One main technique in TDA is that of Betti numbers, which give the number of "holes" in each dimension. Classically, TDA has demonstrated to be a useful data analytic technique in revealing hidden geometrical structures of complex datasets through the analysis of Betti numbers and persistent homology [2].

The mathematical structure of calculating Betti numbers makes this method a natural candidate for a quantum analog [3], denoted as qTDA. While there have been many advancements of the original quantum algorithm [4, 5, 6, 7, 4], the majority of these advancements assumes a universal quantum processor or a hybrid method with creative circuits. Taking into consideration the shortcomings of the NISQ era to handle intricate entanglement and circuits with deep gates, this manuscript describes a hybrid qTDA method that is more scalable to augment the classical algorithm used to calculate Betti number. For instance, we suggested a hybrid algorithm known as variational quantum deflation method to calculate the eigenspectrum of the boundary operators rather than the tradition quantum phase estimation as mentioned by Lloyld at el. [3]

Motivated by the power of using topology to analyze the hidden geometrical structure in real world data, we addressed the question of how different quantum encoding techniques can affect the topology of embedded data structure within complex Hilbert space. Specifically, we explored the variation of persistent homology to understand the topological changes within different encoding techniques and with the original data. The empirical analysis displays that there are subtleties for each of the encoding techniques. In fact, our results reveal significant geometric differences between the original data and the encoding techniques, as well as geometric differences between each of the encoding techniques. Subsequently, this study implies further theoretical understanding is needed to derive the correct unitary operator to encode classical data beyond heuristic when designing a quantum variational circuit for machine learning application.

## 2 Applications of Quantum to Real-World Data

As quantum computing has become increasingly relevant as an advantage over classical computing there has been an increase in quantum analogs of machine learning algorithms. A few examples include linear regression [8], clustering [9, 10], utilizing kernels for support vector machines [11] with intuitive extensions to neural networks [12, 13], and generative modeling [14]. As one may imagine there are situations were quantum is no better than classical methods [15]. To understand how this may hold true consider modeling binary classification with a large feature space. If there is a clear delineation of the two classes within enough features a simple linear regression algorithm will model the data quite well.

Given the nature of quantum algorithms many questions arise about how to incorporate classical data into a quantum circuit. A natural method to transfer continuous valued entries in a data record is to capture each value in the array as a rotation and capture each data point as an *angle* [13, 12]. Encoding the values of the data record to a quantum state through *amplitude* is also quite natural. The authors in [16] display a divide and conquer method, similar to that of the method in the seminal Grover-Rudolph paper [17], which trades the number of qubits for the time to encode. This complexity is opposite of the more qubit less

gate intensive method of angle encoding, or that of *quantum Fourier transform* (QFT). The last method mentioned is *IQP* [18] which is noted by the authors to leverage a quantum advantage, but the series of gates yields more of a neural network mapping than a true representation of the original data. There are other encoding techniques [19, 20, 21], for simplicity, these techniques are not considered.

Mapping data to a quantum circuit has shown to be quite difficult and non-trivial [13]. Moreover, while these methods treat the data as a map to another Hilbert space the data is quite frankly mapped to a series of operators in the space of special unitary operators in dimension $2^n$, denoted as $\mathrm{SU}(2^n)$. However, there are explorations that are fairly in-depth in the analysis, which include authors in [22] that give criteria to when there is a quantum advantage in statistical modeling tasks, and the authors in [1] explore a partial Fourier series of the operator encoded with a data point. To the authors' knowledge, the paper "Effect of data encoding on the expressive power of variational quantum-machine-learning models" [1] is the first effort to consider data from this perspective. This observation about data mapped to parameters of operators points to the question about what information, if any, is lost when data is encoded into a circuit.

## 3 Encoding Approaches

The encoding methods noted in Section 2 will be discussed in detail to sample current methods and establish a deeper understanding. In a closed system there is a periodic nature of quantum mechanics. From the periodic characteristics, it may be necessary to map the data point $D_i$ to a normalized form. There are many methods, such as dividing this vector by its magnitude, a significant amount relying on local information. Defining $\mathcal{S}^n = \left\{ x \in \mathbb{R}^n : x_i > 0 \ \forall \ i \text{ and } \sum_{i=1}^{n-1} x_i = 1 \right\}$, another simplex, there exists a homeomorphic mapping of the form

$$\begin{aligned} f &: \mathbb{R}^{n-1} \to \mathcal{S}^n, \\ f(x) &= \frac{(e^{x_1}, e^{x_2}, \ldots, e^{x_{n-1}}, 1)}{1 + \sum_{i=1}^{n-1} e^{x_i}}, \\ f^{-1}(s) &= \big( \log(s_1/s_n), \log(s_2/s_n), \ldots, \log(s_{n-1}/s_n) \big). \end{aligned} \tag{1}$$

One may verify $f$ is a homeomorphism, and therefore, ensures no loss of information.

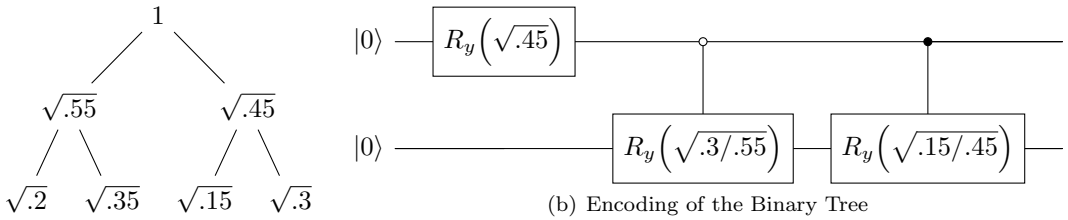
*Angle* encoding is the first technique explored [13, 12]. While this approach is intuitive has been noted to not fully leverage quantum, and in particular, each entry in the data point vector $D_i$ is mapped to a qubit, increasing the size of a typical register. The encoded operator has the mapped form

$$D_i \to \bigotimes_{l=1,2,\ldots,|D_i|} \exp(-iX_l D_i^l) \left| 0 \ldots 0 \right\rangle, \tag{2}$$

where $X_l$ is the Pauli $X$ gate acting on the $l^{th}$ qubit and $D_i^l$ is the $l^{th}$ entry.

The *amplitude* approach assumes all the data points $\tilde{D}_i \in \mathcal{S}^n$. As noted above, one may map any data point into this form without any loss of information. The authors in [16] display an efficient method to encode the amplitudes of the data, utilizing the method in the Grover-Rudolph [17] approach to load probability distributions. Given the intricate procedure of

the technique it will not be discussed in full detail. Moreover, the authors give a detailed description of their algorithm, enabling one to implement this approach. For completeness, an example of how to implement this particular amplitude encoding is given. Take the simple example of encoding the conveniently prepared data point $(\sqrt{.2}, \sqrt{.35}, \sqrt{.15}, \sqrt{.3})$ which is decomposed into a binary tree by recursively splitting each piece of the data point in half with the goal of creating the state $\sqrt{.2}\,|00\rangle + \sqrt{.35}\,|10\rangle + \sqrt{.15}\,|01\rangle + \sqrt{.3}\,|11\rangle$. Each split is then normalized and the left node is taken as the parameter of the $R_y$ gate. This binary tree and subsequent circuit is shown in Figure 1.




(a) Binary Tree Decomposition

(b) Encoding of the Binary Tree

Fig. 1. This is an example on how to amplitude encode data: (a) displays the binary tree decomposition of $(\sqrt{.2}, \sqrt{.35}, \sqrt{.15}, \sqrt{.3})$, and (b) is the respective circuit to encode the binary tree.

Lastly, the *IQP* encoding approach [18] is discussed. The authors assume $x \in (0, 2\pi]^n$ and note the approach suggests a quantum advantage. Taking the function $f$ in Equation 1 and defining the function $\tilde{f} = 2\pi \cdot f$ that maps $\tilde{f} : \mathbb{R}^{n-1} \to (0, 2\pi]^n$ does not lose information since $f$ is a homeomorphism. Denoting $Z_i$ as the Pauli $Z$ gate acting on the $i^{th}$ qubit the authors define the specific unitary operator

$$U_Z(x) = \exp\left(\sum_{i=1}^{n} x_i Z_i + \sum_{i=1}^{n}\sum_{j=1}^{n} (\pi - x_i)(\pi - x_j) Z_i Z_j\right) \tag{3}$$

for application, while giving the general operator as $U_\Phi(x) = \exp\left(\sum_{S \subset \{1,2,\ldots,n\}} \Phi_S(x) \prod_{i \in S} Z_i\right)$. One may note the coefficients in the quadratic terms are centered around 0 with standard deviation of 1, and the original data may be mapped accordingly into a different form. Denoting $H$ as the Hadamard gate, the IQP encoding is defined as

$$D_i \to U_Z(D_i) H^{\otimes n} U_Z(D_i) H^{\otimes n} \,|0 \ldots 0\rangle\,. \tag{4}$$

The authors in [19] derived this encoding by considering Ising interactions of the unitary operators in $U_Z$ and the Hadamard gates adding uniform superpositions.

## 4 Topological Data Analysis and Quantum Computation

With the explosion of statistical modeling it has become more prevalent to understand the geometric structure for further insight into how to model the data, identify weak structures of underrepresented data to anticipate failure, and compare training data and data observed after training. Topological data analysis (TDA) has been exhibited to be a powerful tool to

obtain this goal [23, 24] but is quite computationally intensive. One particular method that is quite useful in classifying the geometric structure of the data is calculating the Betti number for various topological features.

To understand what a Betti number represents more intuitively, consider a record of data points. First it is necessary to construct a *simplicial complex*, which is a collection of the data points with an $\epsilon$-sized ball around each point that creates individual points, lines, triangles, tetrahedron, and corresponding iterative higher-level simplicial objects; $\epsilon$ is a hyperparameter and is intuitively called the *grouping scale*. The collection of simplices created from parameter $\epsilon$, denoted as $S^{\epsilon}$, is known as a *Vietoris-Rips simplicial complex*. After the construction of these objects the number of connected data points, one-dimensional "circular holes", two-dimensional areas void of data points, and corresponding higher-dimensional voids. For $k \in \{0, 1, 2, \ldots\}$ the Betti number $b_k$ corresponds to the respective topological descriptions above.

Betti numbers have a deeper mathematical description with homology, which is essential to describe and incorporate in a circuit. Given a data set $D_s$, denote $H_k^{\epsilon}(D_s)$ as the $k^{th}$ homology group of $D_s$ generated from the parameter $\epsilon$. The complete simplicial complex created from $\epsilon$ is defined $H^{\epsilon} = \bigoplus_k H_k^{\epsilon}(D_s)$. To connect individual simplices, define the *boundary map* as $\delta_k : H_k^{\epsilon}(D_s) \to H_{k-1}^{\epsilon}(D_s)$, and given the derivation of the simplicial complex, one may see the natural mapping of $\delta_k$. Denoting the kernel of a function as ker and the image of a function as Im, with this structure we may define the $k^{th}$ homology as the quotient space $H_k^{\epsilon}(D_s) = \ker \delta_k / \text{Im } \delta_{k+1}$ and $b_k = \dim\big(H_k^{\epsilon}(D_s)\big)$.

This structure enables a derivation towards an generator of connectivity. Combinatorial Laplacians [25] give the exact generator and has the form $\Delta_k = (\delta_k)^{\dagger}\delta_k + \delta_{k+1}(\delta_{k+1})^{\dagger}$, and one may see that $\Delta_k$ is a Hermitian matrix. The combinatorial Laplacian may calculate the $k^{th}$ Betti number, $\beta_k$, by deriving

$$\beta_k = \dim \ker \Delta_k. \tag{5}$$

See [26] for further more in-depth information about Betti numbers.

Since the boundary maps are linear this gives makes this algorithm a candidate for a quantum analog [3], denoted as *qTDA*. In the quantum setting for the space $H_k^{\epsilon}$ is spanned by $|s_k\rangle$ where $s_k \in S_k^{\epsilon}$, where $S_k^{\epsilon}$ is the set of $k$-simplices. The boundary map applied to $|s_k\rangle$ has the form

$$\delta_k |s_k\rangle = \sum_j (-1)^j |s_{k-1}(j)\rangle \tag{6}$$

where $s_{k-1}(j)$ is the $k-1$ simplex on the boundary of $s_k$ with the $j^{th}$ vertex removed from $s_k$.

Since the derivation of qTDA there have been many extensions [4, 5, 6]. However, the general flow of the algorithm has been consistent; please see [27] for a brief overview. The outline of the algorithm is displayed in Algorithm 2.

The general flow of the circuit side-steps how to incorporate real-world data into the circuit. Currently, there are two main methods: (1) calculate the distances (or pseudo-distances), apply the $\epsilon$ hyperparameter filter, make respective connections in classical computation then incorporate this final matrix into the circuit; or (2) encode the data into the circuit and do all calculations within the circuit. The first method is oriented for the NISQ-era, and in fact, may be faster given a small enough data with the available gates. Furthermore, the

Fig. 2. General qTDA

$i \leftarrow 1$
**while** $i \leq L$ **do**
  $\frac{1}{\sqrt{|S_k^\epsilon|}} \sum_{s_k \in S_k^\epsilon} |s_k\rangle \leftarrow$ Grover's algorithm
  $\frac{1}{\sqrt{|S_k^\epsilon|}} \sum_{s_k \in S_k^\epsilon} |s_k\rangle \otimes |s_k\rangle \leftarrow$ copy states to eigenvalue registry with CNOT operations
  $\frac{1}{\sqrt{|S_k^\epsilon|}} \sum_{s_k \in S_k^\epsilon} |s_k\rangle \langle s_k| \leftarrow$ trace out the ancillary register
  $e^{i\Delta_k^{D_s}} \leftarrow$ apply unitary to eigenvalue registry
  Apply phase estimation to eigenvalue registry
  Measure the eigenvalue register to readout the approximated eigenvalue $\tilde{\lambda}$
  $i \leftarrow i + 1$
**end while**
**return** $|\{\tilde{\lambda} : \tilde{\lambda} = 0\}|/L$

first method is difficult to incorporate if one would like to calculate the Betti number of the encoded data. The second method is designed for a universal QPU as the number of gates needed and sensitivity to the noise of qubits plays a huge factor in exact calculations.

While there are hardware shortcomings of the second method it is quite interesting to explore creating such a non-hybrid circuit. The rest of this section explores how one may implement comparing two randomly selected data points from a record. The method in [28] displays how to get the encoded inner product of two data records. Since $\langle\psi|\psi\rangle = 1$ for all non-zero vectors in a circuit one may see that

$$\sqrt{\sum_{k=1}^{n} |\psi_k^i - \psi_k^j|^2} = \sqrt{2 \cdot (1 - |\langle\psi^i|\psi^j\rangle|)}. \tag{7}$$

The authors in [16] give a viable technique to implement a qRAM for the entire record of data. Coupling the qRAM and inner product with the technique in [9], which utilizes the Hadamard test yields, a sub-process to compare randomly drawn data points with a quantum advantage. See Figure 3 for an overview of such a circuit.

## 5 Hybrid Quantum Topological Data Analysis

While the NISQ-era inhibits a purely quantum solution to calculate Betti numbers, there have been efforts to create a hybrid solution [7]. In particular, the Huang et al. display a hybrid method, though the circuit given is a toy example, that calculates Betti number $b_1$ with five data points with the $L^2$-distance between each point. The authors then derive a creative circuit to calculate the $b_1$ score of the network of the boundaries. Such a circuit with pre-calculated boundaries has been previously noted [3, 5]. However, the authors go a step further and derive a matrix of the Equation 6. One may observe that with this matrix all is necessary to finish the Betti number calculation is to derive the eigenvectors and determine $|\{\tilde{\lambda} : \tilde{\lambda} = 0\}|$.

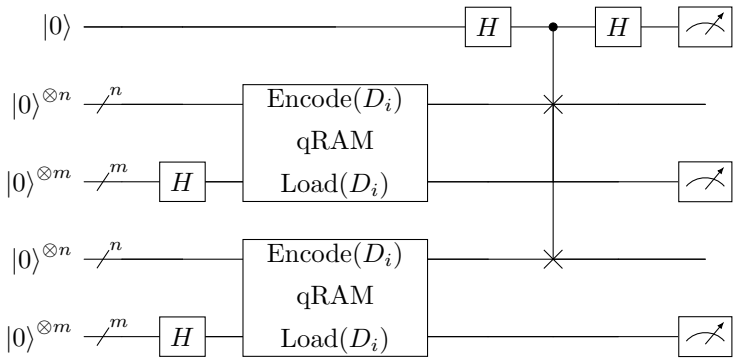



Fig. 3. This circuit displays a sub-process to calculate the inner product of two randomly chosen data points from the same uniformly distributed data points encoded into a quantum circuit and apply the *SWAP test*.

The method in [7] can be expanded by including data encoded into a circuit. However, one must be able to calculate the distance between each point in the record. A technique one may utilize is displayed in the circuit in Figure 3, which is known as the *SWAP test*. This circuit yields the calculation $|\langle E(D_j)|E(D_i)\rangle|^2$, where $E$ denotes the encoding method and is shortened notation for simplicity. One may derive this kernel method explicitly by simplifying the measurement of the circuit on the initial state $|0\dots0\rangle\langle0\dots0|$ ,

$$\begin{aligned}
&\langle 0\dots0|\, E(D_j)E(D_i)^\dagger \mathcal{M} E(D_j)^\dagger E(D_i)\,|0\dots0\rangle = \\
&\langle 0\dots0|\, E(D_j)E(D_i)^\dagger\,|0\dots0\rangle \times \langle 0\dots0|\, E(D_j)^\dagger E(D_i)\,|0\dots0\rangle \\
&= |\langle 0\dots0|\, E(D_j)^\dagger E(D_i)\,|0\dots0\rangle|^2 = |\langle E(D_j)|E(D_i)\rangle|^2.
\end{aligned} \tag{8}$$

A few kernel methods with calculation are described in the Pennylane [29] documentation.

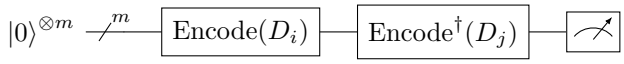



Fig. 4. A kernel method to calculate the absolute value squared of the inner product of two encoded data points, $D_i$ and $D_j$, where the data point $D_j$ is encoded with the inverse of the technique. Finally, measured in the computational basis. This is known as the *fidelity test*.

The proposed hybrid algorithm is given in Algorithm 5. While many of the steps are classical, Step 1 requires a quantum circuit for encoded data points, and the last step is to calculate the eigenvalues, recalling from Equation 5 that $\Delta_k$ is Hermitian. Higgott, Wang, and Brierley in [30] derive a circuit, noted as the *Variational Quantum Deflation* (VQD), which is a NISQ-era friendly implementation to calculate the spectrum of a Hamiltonian. While there are other circuits that may be utilized the calculate the entire set of eigenvalues, given the proliferation of the algorithm, VQD will be noted as the preferred algorithm. However, VQD will also be used as a place holder for similar quantum algorithms. Instances when there isn't a quantum advantage, for example when the circuit is too long or the matrix is small enough, one may then apply a classical eigensolver.

For completeness, the VQD algorithm is explained. VQD was derived as an extension of the variational quantum eigensolver (VQE), see [31] for an overview. Given a Hamiltonian

Fig. 5. Hybrid qTDA

**Input:** Betti number $k$

1: $B \leftarrow$ calculate inner products matrix
2: $B^\epsilon \leftarrow B$ apply $\epsilon$ filter
3: $\{\partial_k, \partial_{k+1}\} \leftarrow B^\epsilon$ calculate the boundary operators
4: $\Delta_k \leftarrow \partial_k^\dagger \partial_k + \partial_{k+1}\partial_{k+1}^\dagger$ calculate combinatorial Laplacian
5: **Either** Decompose $\Delta_k$ as hamiltonian composed of Pauli matrices using Pauli decomposition, and feed that into the VQD sub-circuit and measure $L$ times to readout the approximated eigenvalues $\tilde{\lambda}$ then
6: **return** $|\{\tilde{\lambda} : \tilde{\lambda} = 0\}|/L$
7: **Or** feed $\Delta_k$ into classical eigensolver then
8: **return** $|\{\tilde{\lambda} : \tilde{\lambda} = 0\}|$

$H = \sum c_j P_j$, VQE starts with a real $\lambda$ the ansatz state $|\psi(\lambda)\rangle$ and seeks to minimize the expectation $E(\lambda) := \sum c_j \langle\psi(\lambda)| P_j |\psi(\lambda)\rangle$, denoted as $\lambda_0$, approximating the ground state. To calculate the $k^{th}$ state Higgott, Wang, and Brierley in [30] created the cost function

$$\begin{aligned} V(\lambda_k) &:= \langle\psi(\lambda_k)| H |\psi(\lambda_k)\rangle + \sum_{i=0}^{k-1} \beta_i |\langle\psi(\lambda_k)|\psi(\lambda_i)\rangle|^2 \\ &:= E(\lambda_k) + \mathcal{B}(k, \lambda_k) \end{aligned}$$

where the $\beta_i$ values are chosen sufficiently large to ensure orthogonality, $|\langle\psi(\lambda_i)|\psi(\lambda_j)\rangle|^2 = 0$ for $i \neq j$. Denote $R(\lambda_k)$ as the procedure to prepare the circuit. The algorithm starts with an initial guess then iterates until a designated decision to stop. The schematic is given in Figure 6 and is given in generality to adjust for evolving techniques. For instance, there are a number of ways in which to calculate the expectation including the fidelity test described in Equation 8 or the "destructive SWAP test" [32].

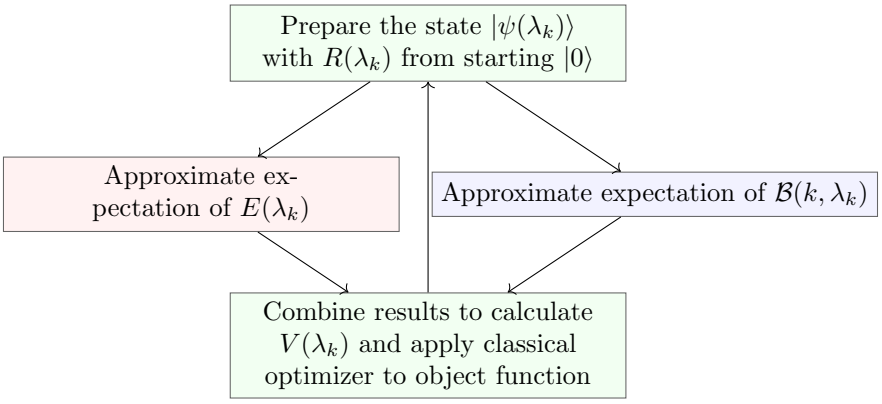



Fig. 6. Variational algorithm schematic of the VQD process.

Fig. 7. VQD for Spectrum Approximation

**Input:** Calculate the first $K+1$ eigenvalues.

1: Apply VQE to approximate $\lambda_0$
2: $i \leftarrow 1$
3: **while** $i \leq K$ **do**
4: $\quad \lambda_i \leftarrow$ apply procedure in Figure 6
5: $\quad i \leftarrow i+1$
6: **end while**
7: $\{\tilde{\lambda}\} \leftarrow$ approximate eigenvalue spectrum using $\lambda_0, \ldots, \lambda_k$ from $L$ measurements.
8: **return** $\{\tilde{\lambda}\}$

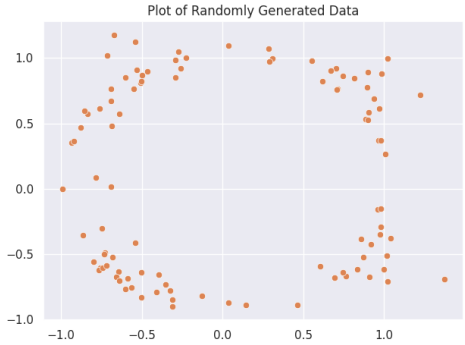



Fig. 8. Scatter plot of the randomly generated two dimensional data analyzed.

## 6 Empirical Analysis

To display the potential differences in the different encoding methods, one hundred randomly generated data points were generated and encoded with the angle method, amplitude method, and the IQP method. The Euclidean distance is applied to the original data and the data encode with each of the three approaches. After the distances are calculated the Betti number for $b_1$ is derived at different threshold levels. While there are other methods, for simplicity these three methods are considered. Given the stark contrast shown in Figure 10 between all of the data sources, it is believed other encoding methods will have a significant difference between the original data and other respective encoding techniques.

To calculate the Betti number for $b_1$ the algorithm described in Algorithm 5 is utilized, however, given the size of the Hamiltonian matrices, the eigenvalues are calculated classically and the number of eigenvalues equal to 0 are given. The Gudhi package [33] was utilized to calculate the simplices and the respective faces.

The data was generated with NumPy [34]. One thousand two dimensional from a uniform distribution in the interval $[-1, 1]$ was sampled with where each data point is normalized, one thousand two dimensional data points were then generated with a Pareto distribution with $\alpha = 10$, and finally these two sets are added together; see Figure 8 for a scatter plot of the data.

The circuits for each of the encoding methods were implemented with Qiskit [35] utilizing a simulator backend. For each pair of data points fed into the circuit in shown in Figure

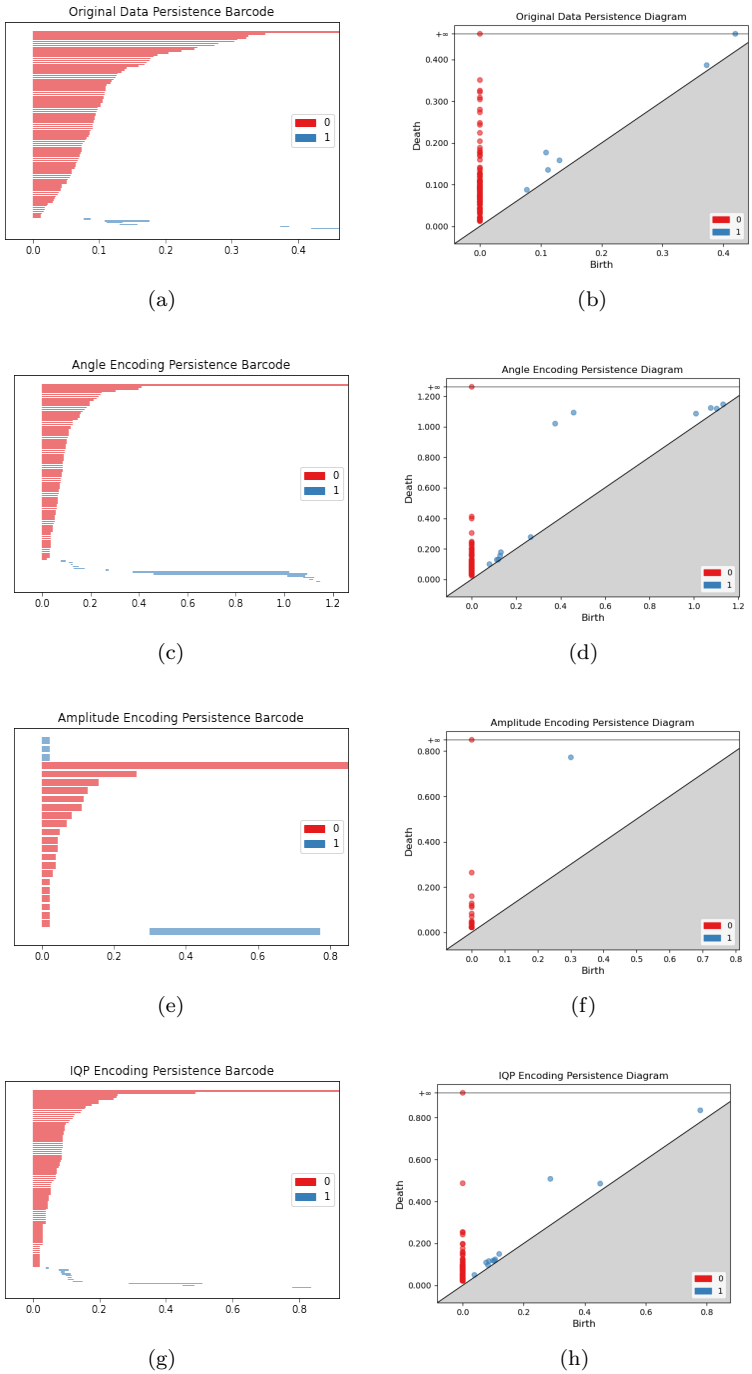
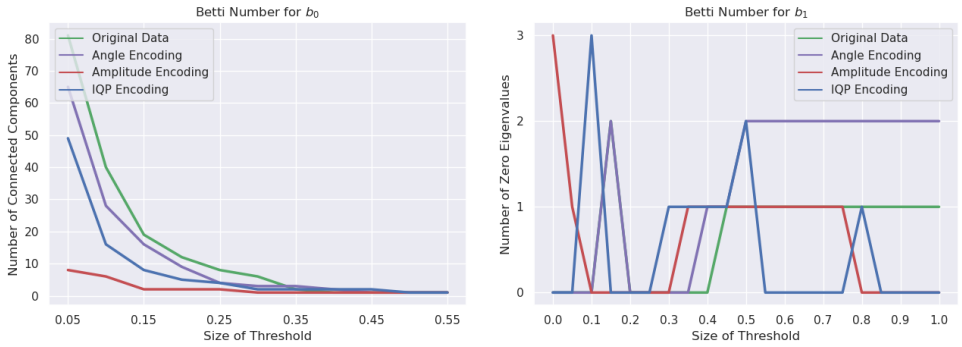



Fig. 9. The persistence barcode and respective diagram for each encoding method and original data. Each row gives insight into each respective geometry, demonstrating difference between all of the methods, as well as differences between each encoding method and the original data.




(a) (b)

Fig. 10. The Betti number for $b_0$ and $b_1$ are computed for the original data, angle encoded data, amplitude encoded data, and the IQP encoded data. Figure (a) displays Betti numbers for $b_0$ with the interval of $[.05, .55]$ with increments of .05, Figure (b) is the Betti numbers for $b_1$ with the interval of $[0.0, 1.0]$ with increments of .05.

4 and ran one thousand twenty four times. This calculation only yields the square of the inner product. Since $|\langle E(D_i)|E(D_i)\rangle| = 1$ for all data records $D_i$, to calculate the Euclidean distance Equation 7 is applied.

To examine how the geometry of the original data is altered with each encoding method, as well the difference of the geometry between each encoding methods, the Betti number of $b_0$ and $b_1$ are computed with increments of .05. Figure 10 (a) displays the evolution of $b_0$, the number of connected components. Interestingly, there is a consistent difference in each Betti number, where around when $\epsilon = .5$ there is stability between the original data and the encoding methods. This consistency shows the encoding methods map the original data rigidly into the respective geometry, however, the information is fairly limited.

Figure 10 (b) shows the Betti number density for the original data and the three encoding approaches. The $\epsilon$-thresholds start at 0.0 and go to 1.0. Unlike the Betti number for $b_0$, there are prominent differences between the original data and all of the encoding approaches. In particular, as the Betti number for the original data stabilize there is both an increase and decrease of the Betti number for the encoding methods, all of which eventually stabilize. This inconsistency displays how each technique effects the noise in the data in different ways.

The difference in the Betti numbers between the three techniques and original data are a bit surprising as one would expect the structure of the data to be intact since the each data point is mapped to a unitary operator, which keeps the underlying structure. However, since each data point is encoded into a unitary operator, it would be more applicable to compare each of these encoded data points as operators within this Lie group than to consider the output of each operator as a point in a Hilbert space.

While the Betti numbers show glaring discrepancies, it does not give insight into the respective geometries. Figure 9 gives the barcode and diagrams for each encoding method and the original data. The barcodes display both subtle and stark contrasts of the geometries, putting more context to the counter-intuitive results of Betti numbers. The diagrams exhibit the 'expressibility' mentioned in Schuld et al. [1], as the IQP method yields an intricate

| (Dim 0, Dim 1) | Original Data | Angle Encoding | Amplitude Encoding | IQP Encoding |
|---|---|---|---|---|
| Original Data Encoding | $(0, 0)$ | $(1.1222, inf)$ | $(4.5792, inf)$ | $(2.4161, inf)$ |
| Angle Encoding | $(1.1222, inf)$ | $(0, 0)$ | $(4.1322, 0.7617)$ | $(1.6849, 0.9655)$ |
| Amplitude Encoding | $(4.5792, inf)$ | $(4.1322, 0.7617)$ | $(0, 0)$ | $(2.9366, 0.443)$ |
| IQP Encoding | $(2.4161, inf)$ | $(1.6849, 0.9655)$ | $(2.9366, 0.443)$ | $(0, 0)$ |

Fig. 11. Further displaying the discrepancy between the encoding methods and the original data. The Wasserstein distance (WD) is applied to the persistence of the zeroth dimension and the first dimension of each method and original data, and the pair in each entry is of the form (WD dimension 0, WD dimension 1).

geometry. Interestingly, the amplitude method, while expressive given the intricacy of the control gates, yielded a simplistic geometry. The barcodes and diagrams were calculated using the Gudhi package [33].

Since TDA is stable against noise [36], we investigated if the difference in topology is not just statistical noise, and it is due to the different data encoding technique. Specifically, we calculated the Wasserstein distance [37] for the persistence of dimensions 0 and 1 for different encodings, we further display the discrepancies of the encoding methods shown in Figure 9 by yielding a quantitative difference in Table 11. As shown in Table 11, the encoding contains entanglement like IQP and amplitude methods show significant deviation from the angle encoding which only contains single qubit operation to encode the data. As a result, this further suggests entanglement can play an important role in encoding the classical data. Furthermore, entanglement also relates to the expressive power of the parameterized quantum circuits used to encode classical data [13].

## 7 Discussion

The technique in this manuscripts displays how to apply qTDA in a hybrid manner, with the quantum advantage in comparing the encoded data points and calculating the eigenvalues of the corresponding Hamiltonian. The results of comparing the three encoding techniques and the data in calculating the Betti number for $b_0$ and $b_1$ showed discrepancies in information retention. Since more intricate geometric structures will contain 'holes', the result holds for other examples.

It is posited that the encoded data must be considered as unitary operators in the Lie group $\mathrm{SU}(2^n)$ and compared within the respective noncommutative geometry. The code used in Section 6 is available on request.

## References

1. Maria Schuld, Ryan Sweke, and Johannes Jakob Meyer (2021), *Effect of data encoding on the expressive power of variational quantum-machine-learning models*, Physical Review A, 103(3), pp. 032430.
2. Larry Wasserman (2018), *Topological data analysis*, Annual Review of Statistics and Its Application, 5, pp. 501–532.
3. M.A. Nielsen and J. Kempe (2001), *Quantum algorithms for topological and geometric analysis of data*, Nature communications, 7(1), pp. 1-7.
4. Casper Gyurik, Chris Cade, and Vedran Dunjko (2022), *Towards quantum advantage via topological data analysis*, Quantum, 6:855.
5. George Siopsis (2018), *Quantum topological data analysis with continuous variables*, arXiv:1804.01558.

6. Ryu Hayakawa (2022), *Quantum algorithm for persistent Betti numbers and topological data analysis*, Quantum, 6:873.
7. He-Liang Huang, Xi-Lin Wang, Peter P Rohde, Yi-Han Luo, You-Wei Zhao, Chang Liu, Li Li, Nai-Le Liu, Chao- Yang Lu, and Jian-Wei Pan (2018), *Demonstration of topological data analysis on a quantum processor*, Optica, 5(2), pp. 193–198.
8. Seth Lloyd (2010), *Quantum algorithm for solving linear systems of equations*, In APS March Meeting Abstracts, 2010, pp. D4–002.
9. Afrad Basheer, A. Afham, and Sandeep K Goyal (2020), *Quantum k-nearest neighbors algorithm*, arXiv:2003.09187.
10. Yijie Dang, Nan Jiang, Hao Hu, Zhuoxiao Ji, andWenyin Zhang (2018), *Image classification based on quantum K-Nearest-Neighbor algorithm*, Quantum Information Processing, 17(9), pp. 1–18.
11. Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd (2014), *Quantum support vector machine for big data classification*, Physical Review Letters, 113(13), pp. 130503.
12. Andrea Skolik, Jarrod R McClean, Masoud Mohseni, Patrick van der Smagt, and Martin Leib (2021), *Layerwise learning for quantum neural networks*, Quantum Machine Intelligence, 3(1), pp. 1–11.
13. Maria Schuld and Nathan Killoran (2019), *Quantum machine learning in feature Hilbert spaces*, Physical Review Letters, 122(4), pp. 040504.
14. Pierre-Luc Dallaire-Demers and Nathan Killoran (2018), *Quantum generative adversarial networks*, Physical Review A, 98(1), pp. 012324.
15. Seth Lloyd, Maria Schuld, Aroosa Ijaz, Josh Izaac, and Nathan Killoran (2020), *Quantum embeddings for machine learning*, arXiv:2001.03622.
16. Israel F. Araujo, Daniel K. Park, Francesco Petruccione, and Adenilton J. da Silva (2021), *A divide-and-conquer algorithm for quantum state preparation*, Scientific reports, 11(1), pp. 1–12.
17. Lov Grover and Terry Rudolph (2002), *Creating superpositions that correspond to efficiently integrable probability distributions*, quant-ph/0208112.
18. Vojtěch Havlíček, Antonio D. Córcoles, Kristan Temme, Aram W. Harrow, Abhinav Kandala, Jerry M. Chow, and Jay M. Gambetta (2019), *Supervised learning with quantum-enhanced feature spaces*, Nature, 567(7747), pp. 209–212.
19. Long Hin Li, Dan-Bo Zhang, and Z.D. Wang (2022), *Quantum kernels with Gaussian state encoding for machine learning*, Physics Letters A, 436, pp. 128088.
20. Dave Wecker, Matthew B. Hastings, and Matthias Troyer (2015), *Progress towards practical quantum variational algorithms*, Physical Review A, 92(4), pp. 042303.
21. Sukin Sim, Peter D Johnson, and Alán Aspuru-Guzik (2019), *Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms*, Advanced Quantum Technologies, 2(12), pp. 1900070.
22. Hsin-Yuan Huang, Michael Broughton, Masoud Mohseni, Ryan Babbush, Sergio Boixo, Hartmut Neven, and Jarrod R McClean (2021), *Power of data in quantum machine learning*, Nature Communications, 12(1), pp. 1–9.
23. Joel Friedman (1998), *Computing Betti numbers via combinatorial Laplacians*, Algorithmica, 21(4), pp. 331–346.
24. Robert Ghrist (2008), *Barcodes: the persistent topology of data*, Bulletin of the American Mathematical Society, 45(1), pp. 61– 75.
25. Fan R.K. Chung and Robert P. Langlands (1996), *A combinatorial Laplacian with vertex weights*, Journal of Combinatorial Theory, Series A, 75(2), pp. 316–327.
26. Gunnar Carlsson and Mikael Vejdemo-Johansson (2021), *Topological Data Analysis with Applications*, Cambridge University Press.
27. Ankit Khandelwal and M Girish Chandra (2023), *Quantum-Enhanced Topological Data Analysis: A Peep from an Implementation Perspective*, arXiv:2302.09553.
28. Yunchao Liu, Srinivasan Arunachalam, and Kristan Temme (2021), *A rigorous and robust quantum speed-up in supervised machine learning*, Nature Physics, 17(9), pp. 1013–1017.
29. Ville Bergholm, Josh Izaac, Maria Schuld, Christian Gogolin, M Sohaib Alam, Shahnawaz Ahmed,

Juan Miguel Arrazola, Carsten Blank, Alain Delgado, Soran Jahangiri, et al. (2018), *Pennylane: Automatic differentiation of hybrid quantum-classical computations*, arXiv:1811.04968.
30. Oscar Higgott, Daochen Wang, and Stephen Brierley (2019), *Variational quantum computation of excited states*, Quantum, 3:156.
31. Jules Tilly, Hongxiang Chen, Shuxiang Cao, Dario Picozzi, Kanav Setia, Ying Li, Edward Grant, Leonard Wossnig, Ivan Rungger, George H. Booth, et al. (2021), *The variational quantum eigensolver: a review of methods and best practices*, arXiv:2111.05176.
32. Ryan LaRose, Arkin Tikku, ´Etude O'Neel-Judy, Lukasz Cincio, and Patrick J. Coles (2019), *Variational quantum state diagonalization*, NPJ Quantum Information, 5(1), pp. 57.
33. Clément Maria, Pawel Dlotko, Vincent Rouvreau, and Vincent Glisse Marc (2016), *Rips complex. In GUDHI User and Reference Manual*, GUDHI Editorial Board.
34. Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. (2020), *Array programming with NumPy*, Nature, 585(7825), pp. 357–362.
35. Gadi Aleksandrowicz, Thomas Alexander, Panagiotis Barkoutsos, Luciano Bello, Yael Ben-Haim, David Bucher, F Jose Cabrera-Hernández, Jorge Carballo-Franquis, Adrian Chen, Chun-Fu Chen, et al.(2019), *Qiskit: An open-source framework for quantum computing*, Accessed on: March 16.
36. David Cohen-Steiner, Herbert Edelsbrunner, and John Harer (2005), *Stability of persistence diagrams*, In Proceedings of the twenty-first annual symposium on Computational Geometry, pp. 263–271.
37. Franco Vazza Maksym Tsizh, Vitalii Tymchyshyn (2023), *Wasserstein distance as a new tool for discriminating cosmologies through the topology of large-scale structure*, Monthly Notices of the Royal Astronomical Society, 522, pp. 2697–2706.