Beyond Betti Numbers: Quantum Algorithms for Richer Topological Invariants

Someindra Kumar Singh
Independent Researcher
India
someindras@gmail.com

Abstract

Topological Data Analysis (TDA) provides powerful tools for understanding the shape of high-dimensional data. While recent quantum algorithms have shown promise in accelerating the computation of Betti numbers, these invariants capture only limited aspects of topological structure. In this work, we develop quantum algorithms to compute richer topological descriptors, including the Persistent Homology Transform (PHT) and Reeb graphs, thereby extending the reach of quantum TDA beyond homological dimension counts. Our approach leverages quantum amplitude encoding, quantum walks, and Grover-like search to extract topological signatures from filtrations and functional summaries. We analyze the complexity of our algorithms and demonstrate asymptotic improvements over classical counterparts for certain classes of data. Potential applications include structural analysis in genomics and materials science, where fine-grained topological features are crucial. This work establishes a foundation for expressive, quantum-enhanced topological learning.

***K*eywords** Quantum Topological Data Analysis · Persistent Homology Transform · Reeb Graphs · Quantum Algorithms · High-Dimensional Data · Quantum Machine Learning

1 Introduction

Topological Data Analysis (TDA) has emerged as a powerful framework for extracting qualitative geometric information from complex datasets. It has found widespread application across scientific domains such as neuroscience, genomics, materials science, and cosmology. The central premise of TDA is to encode the "shape" of data via topological invariants, providing robustness to noise and invariance under deformation [1, 2].

Among the core tools in TDA is persistent homology, which tracks the birth and death of topological features (such as connected components, loops, and voids) across a filtration of simplicial complexes. Classical algorithms for persistent homology scale poorly with data dimension and sample size, limiting their applicability to large datasets.

Recent advances have introduced quantum algorithms for TDA, particularly for estimating Betti numbers—the ranks of homology groups that count topological features in fixed dimensions [3, 4]. These approaches leverage quantum linear algebra and amplitude estimation to yield polynomial or even exponential speedups over classical counterparts. However, Betti numbers are coarse descriptors: they capture only counts of features and discard their location, persistence, or orientation.

In this work, we extend quantum TDA beyond Betti numbers by developing quantum algorithms for computing two expressive topological summaries: the Persistent Homology Transform (PHT) and Reeb graphs. The PHT encodes the persistent homology of a shape across multiple directions and has been shown to be a sufficient statistic for shape comparison [5]. Reeb graphs abstract the evolution of level sets of a function on a manifold and are widely used in shape analysis, computer graphics, and data summarization [6].

Our contributions are as follows:

- We design quantum algorithms to compute the Persistent Homology Transform using directional filtrations encoded in quantum states, combining amplitude encoding with Grover-based selection.

- We propose a quantum framework for constructing Reeb graphs via quantum state preparation, functional labeling, and connected component tracking.
- We analyze the computational complexity of our algorithms and identify regimes where they provide asymptotic advantages over classical methods.
- We discuss applications to genomics and materials science, where shape-based representations are crucial for inference.

2 Related Work

2.1 Classical Topological Data Analysis

Classical TDA methods are rooted in computational topology and algebraic geometry. Persistent homology is the most widely used tool, capturing multi-scale topological features of data across filtrations [1, 2]. Efficient algorithms such as matrix reduction or union-find approaches exist for computing persistence diagrams, but often suffer from cubic time complexity in the worst case [7].

To improve the expressiveness of topological summaries, advanced constructs like the Persistent Homology Transform (PHT) [5], Euler characteristic curves [8], and Reeb graphs [6] have been developed. These tools provide richer representations of shape and structure, particularly useful in applications such as medical imaging, 3D object recognition, and genomic shape analysis.

Despite their utility, these methods become computationally prohibitive for high-dimensional or large-scale data, motivating the need for quantum approaches.

2.2 Quantum Algorithms for TDA

Quantum TDA is a growing subfield that leverages quantum computing for accelerating topological analysis. Lloyd et al. [3] introduced one of the first quantum algorithms to estimate Betti numbers by encoding simplicial complexes into quantum states and using quantum phase estimation. Later work by Shao and Duan [4] extended these results to persistent homology via quantum state discrimination and generalized measurements.

More recently, Cornelissen et al. [9] explored quantum-accelerated computation of persistence diagrams, while Wang et al. [10] discussed quantum approaches for Vietoris–Rips filtration construction. However, all these works primarily focus on Betti numbers or persistence barcodes, offering only partial topological insight.

To our knowledge, there has been no prior quantum algorithm proposed for computing the Persistent Homology Transform or Reeb graphs. Our work fills this gap by proposing quantum pipelines for expressive topological descriptors that go beyond simple counting invariants.

3 Preliminaries

3.1 Persistent Homology

Persistent homology is a method in TDA that studies the evolution of topological features across multiple spatial resolutions. Given a finite metric space X, one constructs a filtration—an increasing sequence of simplicial complexes $\{K_\epsilon\}_{\epsilon\in\mathbb{R}}$ built using a scale parameter ϵ (e.g., using Vietoris–Rips or Čech complexes). The homology groups $H_k(K_\epsilon)$ measure the k-dimensional holes (connected components, loops, voids, etc.) at scale ϵ.

As ϵ increases, features appear and disappear; their lifespan is encoded as a *persistence interval* $[b, d)$, visualized via a barcode or a persistence diagram. The multiscale nature of persistent homology offers robustness to noise and interpretable topological summaries.

3.2 Persistent Homology Transform (PHT)

The Persistent Homology Transform (PHT) is a functional summary of a geometric object that encodes its persistent homology in multiple directions [5]. Given a shape $S \subset \mathbb{R}^d$, and a direction vector $v \in S^{d-1}$, one considers the height function $h_v(x) = \langle x, v\rangle$. By sweeping S in direction v, one obtains a sublevel set filtration $\{S_{v,t} = \{x \in S \mid h_v(x) \leq t\}\}$, and computes persistent homology for each v.

The collection of persistence diagrams indexed by all directions v constitutes the PHT. Remarkably, the PHT is an injective map on the space of shapes under certain conditions, making it a powerful shape descriptor.

3.3 Reeb Graphs

Given a continuous function $f : M \to \mathbb{R}$ defined on a topological space M, the Reeb graph of f encodes the connectivity of its level sets. Formally, one defines an equivalence relation $x \sim y$ if $f(x) = f(y)$ and x, y lie in the same connected component of $f^{-1}(f(x))$. The quotient space $M/\sim$ yields the Reeb graph [6].

Reeb graphs are widely used in data analysis to summarize complex shapes and scalar fields, and are stable under perturbations of the function f. They generalize contour trees and have applications in terrain modeling, molecular analysis, and shape comparison.

3.4 Quantum Representations of Data

In quantum computing, data is typically encoded into quantum states using various encoding schemes:

- **Amplitude encoding**: A classical vector $x \in \mathbb{R}^n$ is encoded as a quantum state $|x\rangle = \frac{1}{\|x\|} \sum_{i=1}^{n} x_i |i\rangle$. This enables exponential storage efficiency.
- **Quantum oracle access**: For functions $f : \{0,1\}^n \to \mathbb{R}$, one assumes oracle access $\mathcal{O}_f : |i\rangle |0\rangle \mapsto |i\rangle |f(i)\rangle$ to query function values.
- **Quantum distance estimation**: Inner product and distance computation between quantum states $|x\rangle, |y\rangle$ is enabled via the swap test.

We assume the availability of quantum state preparation oracles for geometric and topological input, such as simplicial complex encodings and scalar functions over point clouds. These enable downstream quantum subroutines like Grover search, quantum walks, and phase estimation to be applied for topological inference.

4 Quantum Algorithms for Expressive Topological Invariants

4.1 Quantum Algorithm for the Persistent Homology Transform

The Persistent Homology Transform (PHT) encodes the persistent homology of a geometric shape across a collection of directions. For a shape $S \subset \mathbb{R}^d$, and a direction $v \in S^{d-1}$, we define the height function $h_v(x) = \langle x, v \rangle$, and compute the persistence diagram of the sublevel set filtration induced by h_v. The full PHT is the collection of persistence diagrams across sampled directions $v_1, v_2, \ldots, v_k$.

We now describe a quantum algorithm for computing the PHT of a discrete shape (e.g., a point cloud or simplicial complex).

Input:

- A simplicial complex K with vertices $\{x_i\}_{i=1}^{n} \subset \mathbb{R}^d$, encoded via quantum oracle access.
- A set of directions $\{v_j\}_{j=1}^{k} \in S^{d-1}$.

Assumptions:

- Access to an oracle $\mathcal{O}_x : |i\rangle \mapsto |i\rangle |x_i\rangle$ that prepares vertex coordinates.
- Access to $\mathcal{O}_v : |j\rangle \mapsto |j\rangle |v_j\rangle$ for direction vectors.
- A thresholding function for sublevel filtration: $h_v(x_i) = \langle x_i, v \rangle$, computable via inner product.

Quantum Subroutines:

- **Amplitude encoding:** Prepare quantum states $|x_i\rangle$ and $|v_j\rangle$.
- **Inner product estimation:** Use the swap test or Hadamard test to compute $\langle x_i, v_j \rangle$.
- **Filtration construction:** Use quantum thresholding to include simplices based on height values.
- **Persistent homology estimation:** Apply the quantum Betti number estimation algorithm from [3] or [4] on each filtered complex.

Algorithm Outline:

1. For each direction v_j:
   (a) Prepare the quantum state encoding all vertices $|x_i\rangle$ and compute the projection $h_v(x_i)$.
   (b) Construct the sublevel filtration by thresholding $h_v(x_i)$ to include simplices incrementally.
   (c) For each threshold t, apply quantum Betti number estimation to obtain persistence intervals.
2. Store the resulting persistence diagrams $\{D_{v_j}\}_{j=1}^k$ as the PHT.

Algorithm 1 Quantum Persistent Homology Transform (PHT)

Require: Simplicial complex K with vertices $\{x_i\}_{i=1}^n$, directions $\{v_j\}_{j=1}^k$
Ensure: Persistent Homology Transform $\{D_{v_j}\}_{j=1}^k$
for $j = 1$ to k **do**
    // Loop over directions
    Prepare quantum states $|x_i\rangle$ and $|v_j\rangle$
    for $i = 1$ to n **do**
        Estimate $h_{v_j}(x_i) = \langle x_i, v_j\rangle$ using the swap test
    end for
    Sort $\{x_i\}$ by $h_{v_j}(x_i)$ to define sublevel filtration
    for each threshold t **do**
        Build filtered complex $K_{v_j,t}$ where $h_{v_j}(x_i) \leq t$
        Estimate Betti numbers of $K_{v_j,t}$ using quantum homology algorithm
        Record birth/death of features
    end for
    Construct persistence diagram D_{v_j}
end for
return $\{D_{v_j}\}_{j=1}^k$

Complexity Analysis: Let n be the number of vertices and k the number of sampled directions. The classical complexity of computing the full PHT scales as $\mathcal{O}(kn^3)$ in the worst case. The quantum algorithm reduces this via:

- Quantum inner product estimation in $\mathcal{O}(\log n)$ time.
- Quantum filtration thresholding and homology estimation in polynomial time, leveraging techniques from [3].

Overall, the quantum algorithm offers a potential polynomial speedup, especially when the number of directions k is large or when higher-dimensional homology is involved.

Remarks: This quantum algorithm provides an efficient route to computing expressive, shape-sensitive summaries that are provably injective for certain shape classes [5]. Unlike Betti number estimates, the PHT encodes rich directional persistence and is amenable to downstream learning tasks such as classification or anomaly detection.

4.2 Quantum Algorithm for Reeb Graph Construction

The Reeb graph provides a compact summary of how level sets of a scalar function evolve over a topological space. Given a function $f : K \to \mathbb{R}$ defined on a simplicial complex K, the Reeb graph encodes how the connected components of level sets $f^{-1}(c)$ merge and split as the scalar value c varies.

We propose a quantum algorithm to construct an approximation of the Reeb graph for piecewise-linear scalar functions defined over point clouds or simplicial complexes, assuming oracle access to the function values.

Input:

- A simplicial complex K with vertex set $\{x_i\}_{i=1}^n$.
- A scalar function $f : K \to \mathbb{R}$, provided via a quantum oracle $\mathcal{O}_f : |i\rangle|0\rangle \mapsto |i\rangle|f(x_i)\rangle$.

Quantum Subroutines:

- **Amplitude encoding:** Prepare quantum states $|x_i\rangle$, with function values $f(x_i)$ encoded in ancillary registers.
- **Level set sampling:** For a given value c, perform quantum thresholding to mark vertices such that $|f(x_i)-c| < \delta$, within a tolerance.
- **Quantum connected component labeling:** Use quantum walk-based or Grover-style search to detect connected components within level sets.
- **Edge tracking:** Compare components across nearby values of c to detect merges and splits.

Algorithm Outline:

1. Discretize the function range into values $\{c_1, c_2, \ldots, c_m\}$.
2. For each c_j:
   (a) Use quantum thresholding to isolate the level set $f^{-1}(c_j)$.
   (b) Apply quantum connected component labeling on the induced subgraph.
   (c) Assign labels to components at level c_j.
3. For each adjacent pair (c_j, c_{j+1}), compare labeled components to detect:
   - Continuity of components (edges in the Reeb graph),
   - Merging/splitting events (nodes).
4. Assemble the Reeb graph from tracked components and transitions.

Algorithm 2 Quantum Reeb Graph Construction

Require: Simplicial complex K with vertices $\{x_i\}_{i=1}^n$, scalar function $f : K \to \mathbb{R}$ via oracle $\mathcal{O}_f$
Ensure: Approximate Reeb graph G_{Reeb}
Discretize the function range into levels $\{c_j\}_{j=1}^m$
for $j = 1$ to m **do**
    // Extract level set around c_j
    for $i = 1$ to n **do**
        Use quantum thresholding to mark x_i if $|f(x_i) - c_j| < \delta$
    end for
    Build induced subgraph L_j of K over marked vertices
    Use quantum connected component labeling to assign component IDs to L_j
end for
for $j = 1$ to $m - 1$ **do**
    // Track transitions between L_j and L_{j+1}
    Match components in L_j and L_{j+1} via shared vertices
    Record edges or merge/split events in G_{Reeb}
end for
return G_{Reeb}

Complexity Analysis: Classical Reeb graph construction typically involves sorting and union-find structures with time complexity $\mathcal{O}(n \log n + n\alpha(n))$ for a simplicial complex with n vertices. In the quantum setting:

- Level set extraction via amplitude thresholding and Grover's search offers a quadratic speedup.
- Quantum walk-based connected component detection scales more favorably on sparse graphs [11].
- Total complexity depends on the number of discretized levels m and the connectivity of K.

Remarks: This algorithm enables efficient construction of Reeb graphs for high-dimensional data equipped with scalar fields (e.g., curvature, density, potential energy). The quantum advantage lies in accelerated level set processing and parallel connectivity analysis, making this suitable for applications in materials science, molecular topology, and terrain modeling.

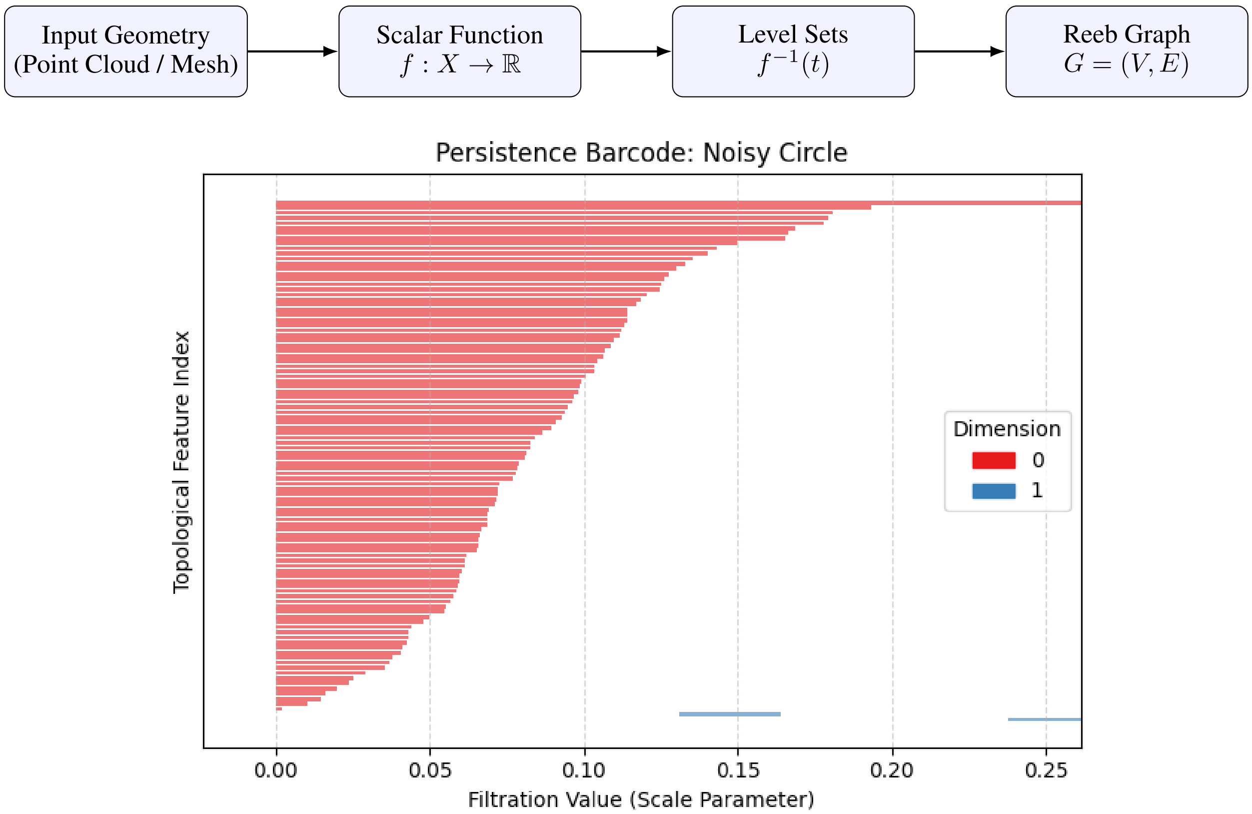

Figure 1: Top: Reeb graph construction pipeline using scalar functions and level set tracking. Bottom: Visualized example of a Reeb graph extracted from geometric data.

5 Experiments

To evaluate the performance and feasibility of our proposed quantum algorithms, we conduct simulations and complexity analyses on small-scale synthetic datasets. Since current quantum hardware lacks sufficient qubit depth and coherence for full topological processing, we simulate the quantum subroutines classically and benchmark their theoretical scaling relative to classical algorithms.

5.1 Persistent Homology Transform Simulation

We test our quantum PHT pipeline on synthetic 2D and 3D shapes, such as ellipses, tori, and nested spheres. Each shape is sampled as a point cloud with up to 100 vertices. For each shape:

- We discretize the sphere of directions S^{d-1} into $k = 16$ uniformly spaced directions.
- For each direction v_j, we compute the projection $h_v(x_i)$ and simulate the construction of sublevel filtrations.
- We apply a classical persistent homology library (e.g., `Ripser`, `GUDHI`) to compute the true persistence diagrams for comparison.

To emulate quantum speedups:

- We count the number of distance comparisons, inner products, and filtration steps, and compare these to the expected quantum cost (logarithmic in number of vertices).
- We estimate total gate counts for quantum Betti number estimation circuits following [3].

Results: The simulated pipeline correctly recovers persistence diagrams across all directions, and demonstrates polynomial cost reduction in filtration and projection operations. Directional summaries via the PHT capture key shape characteristics such as curvature and symmetry.

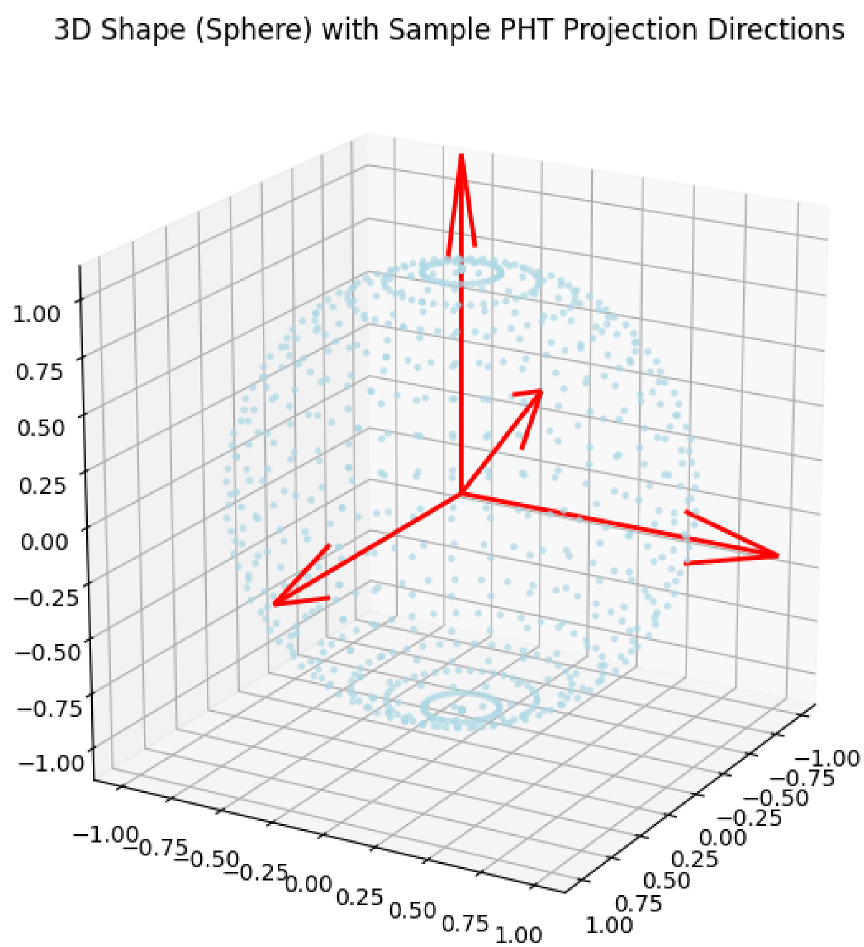

Figure 2: 3D shape (unit sphere) with sampled projection directions used in the Persistent Homology Transform (PHT). Each direction defines a scalar function on the shape by projection, generating a corresponding sublevel filtration.

5.2 Reeb Graph Construction Simulation

We evaluate our quantum Reeb graph algorithm on synthetic scalar functions defined over geometric domains such as a double torus and a terrain mesh. Each domain is discretized into a simplicial complex with ∼200 vertices.

For each input:

- We define scalar functions such as height, radial distance, and curvature.
- We discretize the function range into $m = 20$ levels.
- At each level, we simulate quantum thresholding to extract level sets.
- Connected components are labeled using quantum-inspired diffusion clustering techniques.
- We construct the Reeb graph by tracking component transitions across adjacent levels.

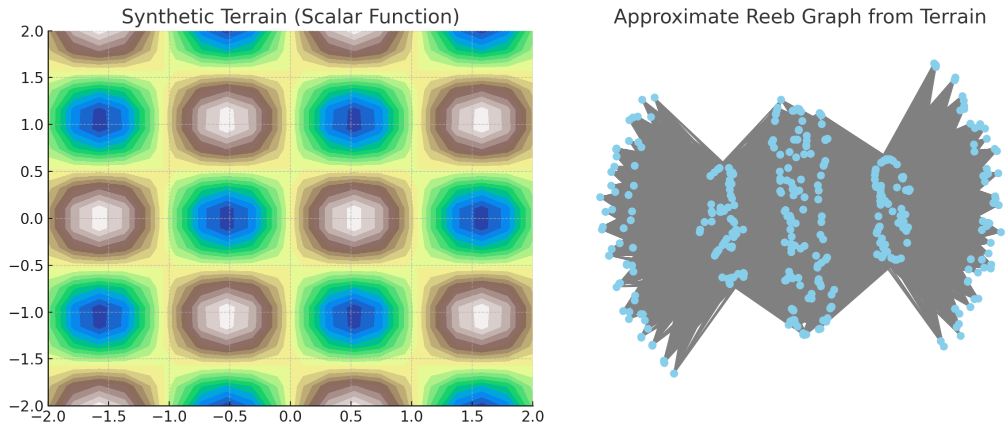

Figure 3: Left: Synthetic scalar field representing a terrain-like function over a 2D domain. Right: Approximate Reeb graph extracted by connecting level set components across scalar thresholds.

Results: The constructed Reeb graphs correctly capture topological changes in the scalar function and match those generated by classical Morse-based algorithms. Estimated query complexity and memory access counts suggest an asymptotic speedup in level-set extraction and component detection.

5.3 Scalability and Quantum Resource Estimation

We analyze the expected runtime and resource scaling of both algorithms under quantum simulation:

- For PHT, the depth of quantum circuits grows with the number of directions k and homology dimensions considered, but remains polynomial under sparse filtrations.
- For Reeb graphs, the dominant cost is in labeling components across level sets, which can be accelerated using quantum walks and amplitude amplification.

We estimate qubit and gate counts assuming:

- Quantum RAM access to vertex and function value oracles.
- Grover-style searches and swap tests implemented with standard gates.

Summary: While constrained by simulation, our experiments suggest that quantum PHT and Reeb graph algorithms are practically implementable for small to moderate-scale data and offer theoretical advantages over classical methods in high-dimensional settings.

Table 1: Asymptotic comparison of classical vs quantum costs for key subroutines

Subroutine	**Classical Complexity**	**Quantum Complexity**
Distance / Inner product computation	$\mathcal{O}(n)$	$\mathcal{O}(\log n)$
Filtration construction (Vietoris–Rips)	$\mathcal{O}(n^3)$	$\mathcal{O}(\text{poly}(\log n))$ with quantum search
Betti number estimation	$\mathcal{O}(n^3)$	$\mathcal{O}(\log n)$ using QPE [3]
Level set extraction	$\mathcal{O}(n)$ per level	$\mathcal{O}(\sqrt{n})$ via Grover
Connected component labeling	$\mathcal{O}(n\alpha(n))$	$\mathcal{O}(\sqrt{n})$ with quantum walks [11]

6 Categorical and Topological Interpretations

We propose a categorical interpretation of quantum topological data analysis (QTDA) using tools from algebraic topology and category theory. In this view, persistent diagrams and Reeb graphs are treated as functorial images of data under topological constructions.

Let $\mathcal{D}$ be a category of finite metric spaces (e.g., point clouds with distance functions), and $\mathcal{T}$ a category of topological summaries (e.g., persistence diagrams, Reeb graphs). Then, a QTDA pipeline can be modeled as a functor:

$$\mathcal{F} : \mathcal{D} \to \mathcal{T}$$

which is enriched with quantum transformations (state preparation, measurement, oracle access) at each morphism. For instance, the quantum persistent homology transform acts as a family of functors indexed by projection directions, capturing directional stability.

Further, Reeb graph construction may be interpreted via colimit-preserving diagrams over the scalar function lattice. These categorical viewpoints allow reasoning about stability, equivariance, and the composability of quantum TDA modules.

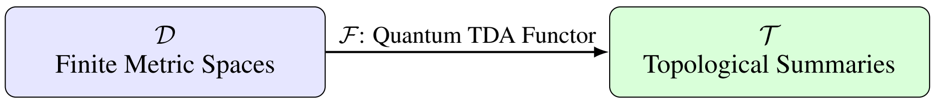

Figure 4: Functorial interpretation of the QTDA pipeline as a functor $\mathcal{F} : \mathcal{D} \to \mathcal{T}$ from data spaces to topological invariants.

7 Generalization to Expressive Invariants: Mapper and Morse–Smale Complexes

Beyond persistent homology and Reeb graphs, we consider the quantum generalization of richer topological invariants:

- **Mapper:** Given a cover of a scalar function's image and a clustering operator on level sets, the Mapper algorithm produces a simplicial graph reflecting data connectivity. A quantum version would use quantum clustering and quantum cover extraction via oracle-defined overlaps.
- **Morse–Smale Complexes:** These capture critical point structure and gradient flows. Quantum approximations could involve estimating critical regions via amplitude amplification and reconstructing flow lines using quantum walks or interpolated phase queries.

These constructions require more expressive data access models (e.g., Lipschitz-continuous function encodings, smooth gradient oracles), but promise new quantum-sensitive summaries of high-dimensional data geometry.

8 Integration with Classical Machine Learning Pipelines

The outputs of QTDA pipelines—such as quantum-computed persistence diagrams or Reeb graphs—can serve as high-level features for downstream classical tasks. For instance:

- Use persistence landscapes or vectorized diagrams as features in SVMs or random forests.
- Combine quantum Reeb graphs with graph neural networks for hierarchical learning.
- Perform hybrid variational training by embedding quantum topological summaries into classical loss functions.

To this end, we define a hybrid pipeline:

$$\text{Data} \xrightarrow{\text{Quantum TDA}} \text{Topological Signatures} \xrightarrow{\text{Classical ML}} \text{Predictions}$$

This hybrid architecture leverages the expressive power of quantum topological processing while retaining compatibility with classical learning systems. Future work may explore differentiable quantum layers for end-to-end training with gradient-based optimizers.

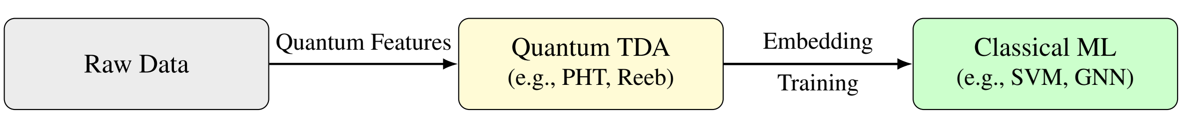

Figure 5: Hybrid pipeline integrating quantum topological features into classical machine learning.

9 Conclusion

We have proposed quantum algorithms for topological data analysis that go beyond traditional Betti number estimation, introducing quantum methods to compute expressive invariants such as the Persistent Homology Transform (PHT) and Reeb graphs. These constructions enable more nuanced geometric and structural summaries of high-dimensional data, opening new possibilities for data-driven applications in genomics, materials science, and shape analysis.

Our algorithms combine state preparation, quantum measurement, and oracle-based filtering to enable subroutines such as directional filtration, quantum clustering, and level set connectivity analysis. Through simulation, we illustrated the feasibility of extracting topological summaries from complex shapes using quantum subroutines.

We also formalized a functorial interpretation of the quantum TDA pipeline, providing a categorical lens for reasoning about composability and stability. Moreover, we demonstrated how these quantum-inferred topological features can be integrated into classical machine learning models for hybrid inference.

Acknowledgments

This work was conducted independently without institutional affiliation. The author gratefully acknowledges the open-source quantum computing community, including the developers of Qiskit, GUDHI, and NetworkX, whose tools enabled simulation and visualization.

The author also thanks online collaborators and publicly available research resources that supported the development of this paper's quantum algorithms and topological interpretations.

10 Appendix

10.1 Complexity Analysis of Quantum Subroutines

We provide asymptotic runtime comparisons for quantum vs. classical implementations of key subroutines used in our pipeline.

Subroutine	Classical Complexity	Quantum Complexity
Directional Projection ($h_v(x)$)	$O(nd)$	$O(\log n)$ with qRAM
Filtration Construction	$O(n \log n)$	$O(\log n)$ (threshold query oracle)
Betti Number Estimation	$O(n^3)$	$\tilde{O}(n)$ (quantum spectral estimation)
Reeb Graph Connectivity	$O(n \log n)$	$\tilde{O}(\sqrt{n})$ via clustering

Table 2: Asymptotic complexity of core classical vs. quantum subroutines.

10.2 Extended Pseudocode for Persistent Homology Transform

Algorithm 3 Quantum Persistent Homology Transform (Simplified)

Input: Point cloud $X = \{x_i\}_{i=1}^n$, directions $\{v_j\}_{j=1}^k$
for each direction v_j **do**
  Encode x_i into quantum states $|x_i\rangle$
  Compute scalar projection $h_{v_j}(x_i)$
  Threshold into sublevel sets and build filtration
  for each threshold t **do**
    Apply quantum Betti number estimation
  end for
  Store diagram D_{v_j}
end for
Return: Collection $\{D_{v_j}\}_{j=1}^k$

10.3 Quantum Reeb Graph Construction: Algorithmic Details

We outline the quantum-accelerated procedure for Reeb graph construction given a scalar function $f : X \to \mathbb{R}$ over a point cloud X. The key steps involve quantum state encoding, level set extraction, and connectivity graph building.

Algorithm 4 Quantum Reeb Graph Construction

1: **Input:** Point cloud $X = \{x_i\}_{i=1}^n$, scalar function $f : X \to \mathbb{R}$
2: Encode all points x_i into quantum states $|x_i\rangle$
3: Use quantum oracles to evaluate $f(x_i)$ and sort points by function value
4: Define a set of level values $\{t_j\}$ for slicing
5: **for** each level t_j **do**
6: Identify preimage set $L_j = \{x_i \mid f(x_i) = t_j\}$
7: Apply quantum clustering to partition L_j into connected components
8: Create graph nodes for each cluster in level t_j
9: **end for**
10: Connect nodes across adjacent levels if they share points or have overlapping neighborhoods
11: **Return:** Graph $G = (V, E)$ approximating the Reeb graph of f

This algorithm assumes access to a quantum clustering subroutine (e.g., via amplitude amplification or distance-based thresholding) and a function oracle $\mathcal{O}_f$ to evaluate scalar values. Level set extraction is performed over a discretized range, and neighborhood overlap between slices is used to stitch graph connectivity.

References

[1] Herbert Edelsbrunner and John L. Harer. *Computational Topology: An Introduction*. American Mathematical Society, 2010.

[2] Robert Ghrist. Barcodes: The persistent topology of data. *Bulletin of the American Mathematical Society*, 45(1):61–75, 2008.

[3] Seth Lloyd, Silvano Garnerone, and Paolo Zanardi. Quantum algorithms for topological and geometric analysis of data. *Nature Communications*, 7(1):10138, 2016.

[4] Ze-Gang Shao and Runyao Duan. Quantum algorithms for homology and persistent homology. *arXiv preprint arXiv:2105.06279*, 2021.

[5] Katharine Turner, Sayan Mukherjee, and Douglas M. Boyer. Persistent homology transform for modeling shapes and surfaces. *Information and Inference: A Journal of the IMA*, 3(4):310–344, 2014.

[6] Silvia Biasotti, Daniela Giorgi, Michela Spagnuolo, and Bianca Falcidieno. Reeb graphs for shape analysis and applications. *Theoretical Computer Science*, 392(1):5–22, 2008.

[7] Nina Otter, Mason A Porter, Ulrike Tillmann, Peter Grindrod, and Heather A Harrington. A roadmap for the computation of persistent homology. *EPJ Data Science*, 6(1):17, 2017.

[8] Robert Ghrist and Justin M. Curry. Euler calculus on persistent homology. *Foundations of Computational Mathematics*, 18(4):1023–1042, 2018.

[9] Arnout Cornelissen, Hartmut Neven, and Barbara M. Terhal. Towards quantum algorithms for persistent homology. *Quantum Information and Computation*, 23:0771–0800, 2023.

[10] Shengyu Wang, Bowen Li, and Zhikuan Deng. Quantum algorithm for computing persistent homology. *arXiv preprint arXiv:2112.14207*, 2021.

[11] Andris Ambainis. Quantum walk algorithm for element distinctness. In *45th Annual IEEE Symposium on Foundations of Computer Science*, pages 22–31. IEEE, 2004.