BIFILTRATIONS OF ENTANGLEMENT AND 2-PARAMETER PERSISTENT HOMOLOGY

AMELIE SCHREIBER

Abstract. Quantum Mutual Information (QMI) is a measure that captures correlations between two quantum subsystems and can be employed to determine the strength of entanglement between qubits in a quantum circuit. A distance metric derived from QMI allows for the analysis of entanglement structure in a quantum circuit. By combining the QMI-based distance metric with discrete time as a second parameter, bifiltrations can be constructed to analyze the entanglement structure in quantum circuits over time and under varying (inverse) QMI thresholds. This approach provides a comprehensive view of the entanglement landscape, enabling the identification of significant topological features and their evolution throughout the circuit's execution. Two-parameter persistent homology with discrete time can be used to systematically compare the entanglement structure of two different circuit representations, identifying topological features that are preserved or altered during the transpilation process. This information can be used to inform the design of quantum algorithms, optimize circuit representations for specific hardware, or develop Application-Specific Integrated Circuit (ASIC) devices tailored to specific quantum computing problems while taking into account the topological interactions between qubits.

Contents

1. A Distance Metric from Quantum Mutual Information

Quantum Mutual Information (QMI) is a measure of the correlation between two quantum subsystems, capturing both the classical and quantum correlations. In the context of quantum computing, it can be employed to determine the strength of entanglement between qubits in a quantum circuit. A distance metric derived from QMI enables the analysis of the entanglement structure in a quantum circuit. In this section, we introduce the concept of QMI, discuss its properties, and present a method for computing a QMI-based distance metric for qubits in a quantum circuit. This distance metric, combined with discrete time as a second parameter, serves as the basis for the bifiltration

Date: April 18, 2023.
2020 *Mathematics Subject Classification.* Primary 81P40 55N31 81P68 81P70 Secondary 81P73 81P65 62R40 68T09 .

construction and the 2-parameter persistent homology analysis of quantum circuits in subsequent sections.

The Quantum Mutual Information, denoted as $M_{i,j}$, between two qubits with reduced density matrices ρ_i and ρ_j, can be defined as:

$$M_{i,j} = S(\rho_i) + S(\rho_j) - S(\rho_{i,j})$$

where $S(\rho)$ represents the von Neumann entropy of a density matrix ρ, and $\rho_{i,j}$ is the joint density matrix of qubits i and j. QMI has the following properties:

$$0 \leq M_{i,j} \leq 2\ln(2).$$

To obtain a distance metric from QMI, we define $D_{i,j}$ as the additive inverse of QMI:

$$D_{i,j} = 2\ln(2) - M_{i,j}.$$

This distance metric satisfies the following properties:

$$2\ln(2) \geq D_{i,j} \geq 0.$$

The distance metric $D_{i,j}$ brings strongly entangled qubits (with $M_{i,j} \to 2\ln 2$) close together ($D_{i,j} \to 0$), while non-entangled qubits (with $M_{i,j} \to 0$) are far apart ($D_{i,j} \to 2\ln 2$).

With this distance metric, one can construct a weighted graph, where each qubit corresponds to a node, and the weight of an edge connecting qubits i and j is given by the distance metric $D_{i,j}$. The topological structure of this graph provides valuable information about the entanglement patterns in the quantum circuit.

Using the QMI-based distance metric, we can apply techniques from topological data analysis to study the structure of entanglement in quantum circuits. In particular, we can use persistent homology, which captures the topological features of a data set at different scales, to identify and quantify entanglement patterns in the circuit. By combining the QMI-based distance metric with discrete time as a second parameter, we can construct bifiltrations and perform 2-parameter persistent homology analysis of quantum circuits. This approach enables us to study the time evolution of entanglement and its topological features, providing insights into the dynamics and complexity of quantum circuits.

In the following sections, we will detail the construction of bifiltrations using the QMI-based distance metric and discrete time, and discuss how 2-parameter persistent homology can be employed to analyze entanglement patterns in quantum circuits. This framework provides a powerful tool for understanding the topological structure of entanglement and its role in the performance of quantum algorithms, with potential implications for the design and optimization of quantum computing protocols.

2. Bifiltrations of Entanglement in Quantum Circuits

2.1. **Circuit Depth and Discrete Time in Quantum Circuits.**

Circuit depth is a measure of the complexity of a quantum or classical circuit, defined as the number of time steps (or layers of gates) required to execute the circuit. In other words, it represents the longest sequence of operations in the circuit that must be executed in series (i.e., one after another). The depth of a circuit is an important factor when evaluating the performance and computational power of a quantum or classical computer.

In a quantum circuit, gates are applied to qubits in discrete time steps, and the overall computation is carried out by applying a sequence of gates to the qubits. Gates that do not share any qubits can be applied in parallel during the same time step. However, if two gates operate on the same qubit, they must be applied sequentially, adding to the circuit depth.

Circuit depth is usually considered to be particularly important in quantum computing because quantum computers are prone to noise and errors. The longer the computation (i.e., the deeper the circuit), the more likely it is that errors will accumulate and affect the final result. In this context, it may be essential to minimize circuit depth to ensure accurate and reliable computations, particularly given the limited coherence times of qubits in current quantum hardware. If however, there are topological error correction methods being employed with sufficient one and two-qubit gate accuracy, the decoherence time can be much longer. As the authors of [Google Quantum AI] state, "Quantum error correction can exponentially suppress the operational error rates in a quantum processor, at the expense of temporal and qubit overhead." In particular, the paper shows a scaling up of a surface code improves error rates. With this in mind, it is important to begin analyzing the topological structure of errors.

2.2. **Error Correction and General Surface Codes.** Persistent homology may offer some insight in this area. In particular, we may find persistent topological features that represent errors, which can be corrected using deep learning models such as those described in [RC]. Note, using gauge group equivariance in such models, applied to surface codes, may provide a key step in solving error correction. The topological nature of not only the transformer model, but the surface code itself, implies errors and their symmetries are topological in nature.

A surface code can be derived from a cellularly embedded graph on a Riemann surface and its dual graph by constructing a specific type of quantum error-correcting code based on the graph structure. Here's a step-by-step process to obtain a general surface code from such a setup:

(1) Cellularly embedded graph: Consider a graph embedded on a Riemann surface such that it divides the surface into disjoint faces (regions or cells) without any overlaps. This graph is called a cellularly embedded graph.
(2) Dual graph: Construct the dual graph of the embedded graph by placing a vertex in each face of the original graph and connecting these vertices with edges that cross the original graph's edges. The dual graph can be thought of as a "mirror" of the original graph, with its structure closely related to the original graph's faces.
(3) Physical qubits: Associate a physical qubit with each edge in the combined graph, which consists of both the original graph and its dual. Since each edge is shared by a pair of adjacent faces (one from the original graph and one from the dual graph), the qubits represent the shared information between these faces.
(4) Check operators: For each vertex in the combined graph, define a check operator (either an X-type or Z-type operator) as the product of the Pauli operators (X or Z) acting on the qubits associated with the edges connected to that vertex. These check operators will be used to detect errors in the surface code.
(5) Logical operators: Define logical operators that act on the encoded logical qubits by selecting non-contractible loops on the Riemann surface that correspond to paths through the combined graph. These logical operators should commute with all check operators and have nontrivial overlap with other logical operators of the same type.

The resulting quantum error-correcting code, defined by the check operators and logical operators, is a general surface code. The robustness of this code comes from its topological nature, where the logical qubits are encoded in the global properties of the graph structure on the Riemann surface. Errors can be thought of as localized excitations in this structure, and the error correction process involves identifying and correcting error strings that connect these excitations.

A general surface code can be obtained from a cellularly embedded graph on a Riemann surface and its dual graph by associating physical qubits with the combined graph's edges, defining check operators based on the vertices, and constructing logical operators using non-contractible loops on

the Riemann surface. The resulting code is a topological quantum error-correcting code that is robust against local errors.

Errors can be thought of as strings propogating along a compact Riemann surface. If the strings close up into a topologically nontrivial loop on the surface, they are no longer correctable by the surface codes and represent a logical error. Such topological errors could theoretically be caught or even anticipated by a (gauge group) equivariant transformer. Such a transormer would represent errors as tokens, and these tokens would then be (information theoretically) clustered in a way that is best captured by persistent homology. In particular we note a close connection to [**?**] and [AS3]. In these paper the Jensen-Shannon distance metric is used to perform a persistent homology analysis of the attention mechanism and its information theoretic representation and clustering of text and image tokens.

2.3. **Bifiltrations.** Bifiltrations are a combinatorial structure that captures the evolution of a simplicial complex along two independent parameters. In the context of quantum circuits, we utilize bifiltrations to analyze the entanglement structure based on the QMI-derived distance metric introduced in the previous section. We outline the process of constructing bifiltrations for quantum circuits by dividing the circuit into time steps and employing the distance metric to create simplicial complexes for each time step. This construction methodology facilitates the investigation of the entanglement structure's evolution along with the time steps and under varying (inverse) QMI thresholds.

By constructing bifiltrations, we can gain insights into the topological properties of entanglement in quantum circuits. The two parameters, time and QMI-based distance, allow us to track the changes in the entanglement structure over time while considering the strength of entanglement between qubits. This approach provides a comprehensive view of the entanglement landscape, enabling the identification of significant topological features and their evolution throughout the circuit's execution.

The bifiltration construction method can be applied to a variety of quantum circuits, offering a powerful tool for analyzing entanglement patterns and their dynamics. By understanding the topological structure of entanglement in quantum circuits, researchers can develop more efficient quantum algorithms, optimize circuit representations, and design quantum hardware that takes into account the entanglement structure. Overall, bifiltrations provide a valuable framework for studying the topological properties of entanglement in quantum circuits, which can ultimately contribute to advancements in the field of quantum computing.

2.4. **Continuous Time and Rivet.** Using continuous time requires us not only to use more sophisticated mathematics, but unsurprisingly more sophisticated software as well. Continuous time indeed requires the use of [Rivet], built specifically for two-parameter persistent homology. We will discuss 2-parameter persistence as it applies to bifiltrations of quantum circuits in the next section.

3. Persistent Homology

3.1. **One Parameter Persistent Homology.** Persistent homology is a powerful mathematical tool from the field of computational topology that provides a robust, multi-scale analysis of topological features in high-dimensional data. It quantifies the persistence of topological features such as connected components, loops, and voids over a range of scales, enabling the identification of significant structures in the underlying space. In this subsection, we provide a mathematically rigorous introduction to one-parameter persistent homology.

Given a topological space X and a continuous function $f : X \to \mathbb{R}$, we aim to study the evolution of the homology groups of X as the function f varies over its domain. To achieve this, we consider the sublevel sets of the function f, defined as:

$$X_a = f^{-1}((-\infty, a]), \tag{1}$$

where $a \in \mathbb{R}$. As the parameter a increases, the sublevel sets X_a form a growing sequence of nested spaces:

$$X_{a_1} \subseteq X_{a_2} \subseteq \cdots \subseteq X_{a_n}, \tag{2}$$

for any sequence of real numbers $a_1 \leq a_2 \leq \cdots \leq a_n$.

For each a_i, we compute the homology groups $H_p(X_{a_i})$ of the corresponding sublevel set, where $p \geq 0$ denotes the dimension of the homological features we are interested in (e.g., $p = 0$ for connected components, $p = 1$ for loops, etc.). As a increases, homological features may appear, merge, or disappear, resulting in changes in the homology groups.

To track the persistence of these homological features, we define an algebraic structure known as a persistence module. A persistence module M is a collection of vector spaces $\{M_a\}_{a\in\mathbb{R}}$ indexed by the real numbers, together with linear maps $ma, b : M_a \to M_b$ for all $a \leq b$, satisfying the following conditions:

(1) $m_{a,a} = \mathrm{id}_{M_a}$ (identity maps) for all $a \in \mathbb{R}$,
(2) $m_{a,c} = m_{b,c} \circ m_{a,b}$ (composition of maps) for all $a \leq b \leq c$.

The homology groups $H_p(X_a), a \in \mathbb{R}$ and the induced homomorphisms $\{f_{a,b} : H_p(X_a) \to H_p(X_b)\}_{a\leq b}$ form a persistence module, which encapsulates the evolution of the topological features over the filtration parameter a.

The key idea in persistent homology is to summarize the information contained in the persistence module using persistence diagrams, a compact representation of the birth and death of topological features over the filtration. A persistence diagram is a multi-set of points in the extended plane $\overline{\mathbb{R}}^2 = \mathbb{R}^2 \cup \infty$, where each point (b, d) represents a homological feature born at b and dying at d. The persistence of a feature is given by the difference $d - b$, and features with high persistence are considered more significant.

To compute persistence diagrams from the persistence module, one can employ various algorithms, such as the standard reduction algorithm or the faster matrix reduction algorithms, which leverage matrix representations of the boundary operators in the filtered simplicial complex. The matrix reduction algorithms transform the boundary matrices into a canonical form, from which the persistence pairs can be directly read off.

One crucial aspect of persistent homology is its stability under perturbations of the input data. This stability is captured by the concept of interleaving distance between persistence modules, which measures the similarity between the evolution of topological features in two different spaces. More specifically, two persistence modules M and N are said to be ϵ-interleaved if there exist linear maps $m_a : M_a \to N_{a+\epsilon}$ and $n_a : N_a \to M_{a+\epsilon}$ for all $a \in \mathbb{R}$, such that $n_{a+\epsilon} \circ m_a = \mathrm{id}_{M_{a+\epsilon}}$ and $m_{a+\epsilon} \circ n_a = \mathrm{id}_{N_{a+\epsilon}}$. Intuitively, an ϵ-interleaving between two persistence modules means that the topological features in one module are matched to similar features in the other module within a tolerance of ϵ.

The interleaving distance provides a solid foundation for the stability of persistence diagrams, as it can be shown that the bottleneck distance between persistence diagrams is bounded by the interleaving distance between their corresponding persistence modules. The bottleneck distance is a popular metric for comparing persistence diagrams and is defined as the minimum value δ such that each point in one diagram can be matched to a point in the other diagram within a distance of δ in the ℓ_∞-norm.

A $\mathbb{C}[x]$-module is an algebraic structure that combines the notion of a vector space over the complex numbers, $\mathbb{C}$, with the action of polynomials in one variable, x. More formally, a $\mathbb{C}[x]$-module is a set M along with an operation $+ : M \times M \to M$ and an action of $\mathbb{C}[x]$ on M, denoted by $\cdot : \mathbb{C}[x] \times M \to M$, such that:

(1) $(M, +)$ is an abelian group.
(2) The action of $\mathbb{C}[x]$ on M is compatible with the addition in M and the multiplication in $\mathbb{C}[x]$.

At a first glance, these two concepts may seem unrelated, but there is a connection between them through the notion of a "graded module." A graded module is a module that has a direct sum decomposition indexed by a partially ordered set, which in our case is the parameter t.

Persistence modules can be seen as graded modules indexed by the parameter t. Similarly, $\mathbb{C}[x]$-modules can be considered as graded modules indexed by the degree of the polynomial. The connection between these concepts is more apparent when considering a specific subclass of persistence modules called "pointwise finite-dimensional" (PFD) persistence modules. A persistence module is PFD if all its vector spaces V_t are finite-dimensional. PFD persistence modules can be represented using a $\mathbb{C}[x]$-module structure, where the algebraic structure of the persistence module is induced by the action of the polynomial ring $\mathbb{C}[x]$ on the module. This connection helps translate topological problems involving persistence modules into algebraic problems involving $\mathbb{C}[x]$-modules, which are often easier to work with and study. Moreover, we can always decompose such modules as,

$$U \simeq \bigoplus_i x^{t_i} \cdot F[x] \oplus \left(\bigoplus_j x^{r_j} \cdot (F[x]/(x^{s_j} \cdot F[x])) \right).$$

One-parameter persistent homology provides a robust, multi-scale analysis of topological features in data by tracking the evolution of homology groups over a filtration parameter. Persistence diagrams and barcode offer a compact representation of the birth and death of topological features, and their stability under perturbations is guaranteed by the interleaving distance between persistence modules. Persistent homology has found applications in a wide range of fields, including shape analysis, data clustering, and the study of complex networks, among others.

3.2. **Two Parameter Persistent Homology.** The theory of multiparameter persistence arises in many contexts in computational topology and data science. Often a point finite cloud $P \subset \mathbb{R}^d$ is obtained from some observational data, and one would like to understand the significant features of the data cloud. It is typical that the data set will have some *"noise"*, and such noise may need to be filtered out. It is in general a very difficult problem to analyse a data set and determine what subset of the data is noise and what is a significant feature of the data. In this case, one can use topological data analysis and persistence homology. This method has proven very robust and effective in applications.

Definition 3.1. A **bifiltered space** X, is a topological space with a family of subspaces $\{X_v \subseteq\}_{v \in \mathbb{N}^2}$, with inclusion maps $X_u \hookrightarrow X_v$ whenever $u \leq v$ in the standard partial order on $\mathbb{N}^2$. We require the following diagram to commute:

$$\begin{array}{ccc} X_u & \longrightarrow & X_{v_1} \\ \downarrow & & \downarrow \\ X_{v_2} & \longrightarrow & X_w \end{array}$$

whenever $u \leq v_1, v_2 \leq w$.

Definition 3.2. Let $\mathbb{K}$ be a field. A **persistence module** M, is a family of $\mathbb{K}$-modules $\{M_u\}_{u\in\mathbb{N}^2}$ with maps

$$\phi_{u,v} : M_u \to M_v$$

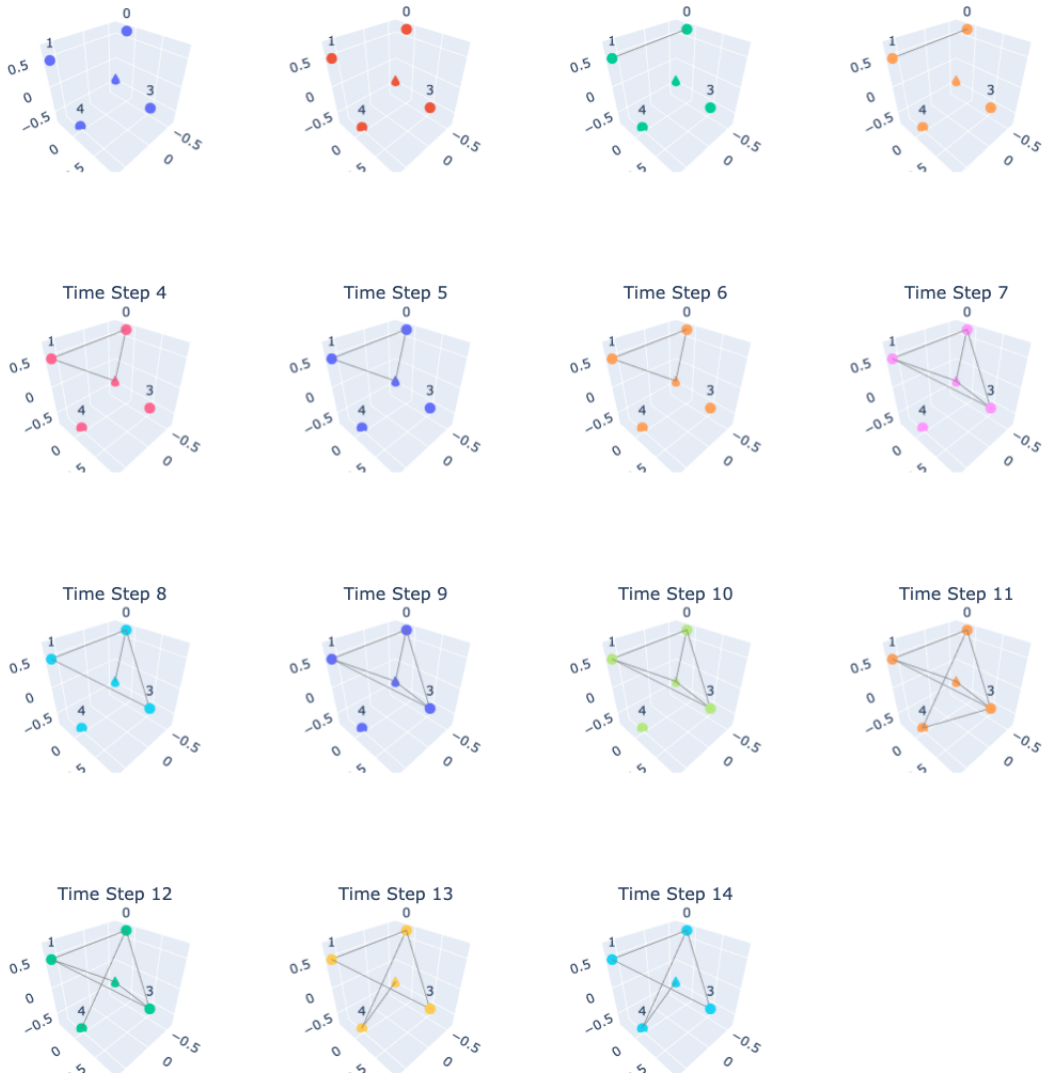
for all $u \leq v$ such that $\phi_{v,w} \circ \phi_{u,v} = \phi_{u,w}$ for all $u \leq v \leq w$. Let M be a persistence module and let $R = \mathbb{K}[x, y]$. Define

$$\alpha(M) = \bigoplus_v M_v$$

where the $\mathbb{K}$-module structure is the direct sum structure, and $x^{u-v} : M_u \to M_v$ is $\phi_{u,v}$ when $u \leq v$ in $\mathbb{N}^2$. This gives an equivalence of categories between the category of finite persistence modules over $\mathbb{K}$, and the category of $\mathbb{N}^2$-graded finitely generated modules over R.

This means we may use commutative algebra and algebraic geometry to study 2-parameter persistent homology as well. Moreover, many of the invariants that arise from this study can be efficiently computed using computer algebra software such as SageMath and Macaulay2. In the next subsection we investigate how we can reduce out study to discrete time, allowing for use of one-parameter persistent homology tools such as persistence diagrams, barcodes, and distance metrics defined on them such as the bottleneck and Wasserstein distance.

3.3. **Persistent Homology of Quantum Circuits.** In this section, we extend the concept of persistent homology to 2-parameter persistent homology, which allows us to analyze the bifiltrations constructed from quantum circuits. This analysis will provide insights into the topological structure of entanglement in quantum circuits. Please see [AS] for the relevant algebraic geometry and [LW] for a more general background on multiparameter persistent homology and some additional algebraic geometry explanations. The following is a single slice along the inverse QMI distance parameter. It shows how entanglement between any two pairs of sites can increase and decrease over time, and how the simplicial complex can "grow", and then "shrink" again because of this. Only the one-skeleton is plotted, with nodes (vertices) corresponding to the qubits.



It is interesting to notice the way the entangelement structure changes as time varies, and the inverse QMI distance threshold is held constant. In particular, we can see varying entanglement structure in some cases, where qubits become connected by edges, and then those edges disappear. This is counter intuitive as most examples of bifiltrations imply only the "growth" of the simplicial complex as each parameter increases. In this case with discrete time, where the first distance metric (QMI) is allowed to vary in time, quantum gates shift entanglement around through the qubits as time increases, causing the simplicial complex to "shrink" in some cases.

3.4. **Free-Resolutions of 2-parameter Persistence Modules.** Free resolutions offer a powerful method in commutative algebra for studying the structure of modules over a commutative ring. In the context of 2-parameter persistence modules, free resolutions can provide insights into the entanglement structure in quantum circuits by analyzing the underlying algebraic structure. In this section, we present an introduction to free resolutions from the perspective of commutative algebra

and demonstrate their application to the study of 2-parameter persistence modules arising from the bifiltrations of entanglement in quantum circuits.

A free resolution of a module M over a commutative ring R is an exact sequence of free R-modules and R-module homomorphisms, where the last term of the sequence is the module M itself. The purpose of a free resolution is to approximate the module M by a sequence of simpler, free modules. Free resolutions play a crucial role in various algebraic constructions, such as the computation of derived functors, Tor and Ext groups, and the investigation of homological properties of modules.

The study of free resolutions of $\mathbb{C}[x, y]$-modules can be applied to the analysis of 2-parameter persistent homology in the context of the quantum computing. In this setting, we are interested in understanding the topological features that persist across two parameters: the (inverse) QMI distance, and time. Here, we provide an explanation of how free resolutions of $\mathbb{C}[x, y]$-modules are connected to the 2-parameter persistent homology analysis.

(1) Bifiltered Persistence Modules: In the 2-parameter persistent homology analysis, we consider bifiltered persistence modules, which are families of $\mathbb{C}$-modules indexed by a pair of parameters $(u, v) \in \mathbb{N}^2$. We have maps $\phi_{(u_1,v_1),(u_2,v_2)} : M_{(u_1,v_1)} \to M_{(u_2,v_2)}$ whenever $(u_1, v_1) \leq (u_2, v_2)$.
(2) Algebraic Representation: We can represent bifiltered persistence modules as $\mathbb{C}[x, y]$-modules by considering the polynomial ring $R = \mathbb{C}[x, y]$. A bifiltered persistence module can be transformed into an $\mathbb{N}^2$-graded $\mathbb{C}[x, y]$-module by taking the direct sum of the $\mathbb{C}$-modules, i.e., $M' = \bigoplus_{(u,v)\in\mathbb{N}^2} M_{(u,v)}$. The maps between the original modules are replaced by multiplication by the corresponding monomials $x^{u_2-u_1}y^{v_2-v_1}$ in R.
(3) Free Resolutions: Given a finitely generated $\mathbb{C}[x, y]$-module, we can study its properties by constructing a free resolution, which is a sequence of free $\mathbb{C}[x, y]$-modules and module homomorphisms that map the free modules to the original module. This free resolution captures the algebraic structure of the module, and its properties can be used to analyze the underlying bifiltered persistence module and the associated topological features.
(4) Invariants and Topological Features: By studying the free resolutions of $\mathbb{C}[x, y]$-modules associated with bifiltered persistence modules, we can extract invariants, such as Betti numbers, that provide insights into the topological features of the attention mechanism and token probability distributions. These invariants can be used to better understand the structure of the attention mechanism across the two parameters: (inverse) QMI distance and time.

For relevant background on the algebraic geometry and free resolutions (or presentations) of 2-parameter persistence modules please refer to [LW] and [AS].

4. Examples with Perturbations of Graph States

4.1. **Graph States.** A graph state is a specific type of multi-qubit state that is associated with a graph, where each vertex represents a qubit and each edge represents a controlled-X (CNOT) gate between the corresponding qubits. They are a well studied class of quantum circuits that are used in quantum cryptography and quantum error correction and can be obtained from surface codes. Every quantum surface code given by a graph cellularly embedded in a Riemann surface has a correspodning graph state, and using the minimal genus of a graph, there always exists a (not necessarily unique) quantum surface code. Moreover, in our study, perturbations of graph states allow for expressing very complex quantum states in the form of a graph state that uses controlled random-unitary gates as entangling gates. We will discuss these perturbations of graph states in the next subsection of this section.

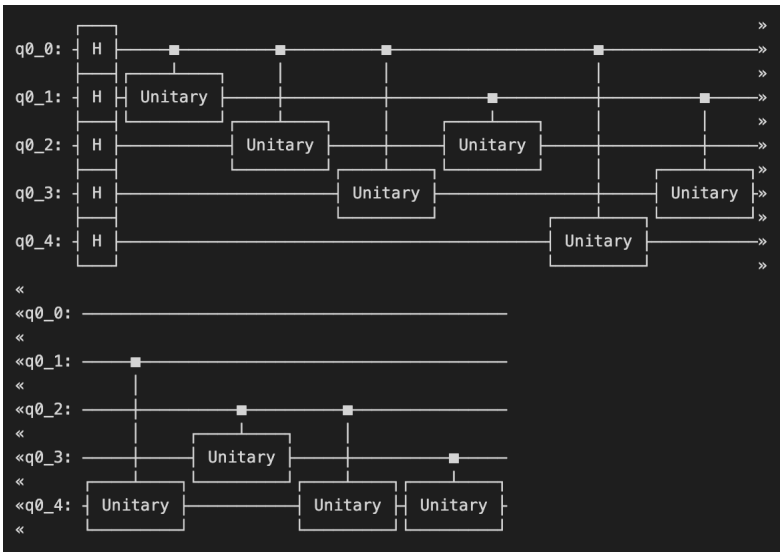
The formal definition of a graph state can be states as follows. Given a graph $G = (V, E)$, with vertices V and edges $E \subset V \times V$, and without loops or multiple edges between nodes, we construct a graph state on $|V|$-qubits by first applying the Hadamard gate to each qubit, and then applying a CNOT gate (or CZ-gate for the dual graphs of surface codes) according to the edges E. The source and target qubits typically do not matter, so long as there is a CNOT (or CZ) entangling gate between any pair of qubits with vertices connected by an edge. However, it is important to note that there are conventions on what is the source and what is the target when dealing with surface code. The conventions are irrelevant for our current purposes so we do not go into them here. As a basic example, take the complete graph K_3, with three vertices and three edges (a triangle). Then we create a graph state circuit using Qiskit as follows:

```
from qiskit import QuantumCircuit

# Initialize a 3-qubit quantum circuit
n_qubits = 3
qc = QuantumCircuit(n_qubits)

# Apply Hadamard gates to all qubits
for i in range(n_qubits):
    qc.h(i)

# Apply controlled-Z gates for each edge in the complete graph K_3
qc.cx(0, 1)  # Edge between qubit 0 and qubit 1
qc.cx(0, 2)  # Edge between qubit 0 and qubit 2
qc.cx(1, 2)  # Edge between qubit 1 and qubit 2

# Draw the quantum circuit
print(qc.draw())
```

4.2. **Perturbations of Graph States.** The perturbation of the graph state is created using a complete graph with **num_qubits** nodes. The quantum circuit is initialized with the given qubits, and Hadamard gates are applied to all qubits. Then, for each edge in the complete graph, random controlled unitary gates are applied using the qubits corresponding to the nodes of the edge. Using random controlled unitaries as opposed to CNOT or controlled-Z gates for entangling qubits has several benefits:

(1) Richer entanglement: Random controlled unitary gates can create a much richer variety of entangled states as they have more degrees of freedom compared to CNOT or controlled-Z gates. This can lead to a more diverse set of entangled states, which can be useful for quantum algorithms that rely on a broad range of entangled states for their operation.
(2) Exploration of quantum state space: By using random controlled unitaries, you can explore a wider range of quantum states and their properties. This can be useful for studying the behavior of quantum systems and developing new quantum algorithms or protocols.
(3) Benchmarking: Random controlled unitary gates can be useful for benchmarking and testing the performance of quantum computers. By generating a wide variety of quantum states and operations, you can evaluate the accuracy, fidelity, and robustness of quantum computers under different conditions.
(4) Transpiler Testing: We can test the efficacy of a transpiler using purturbations of graph states that employ controlled random unitary gates for entangling gates.

In the image below, we see a graph state quantum circuit that uses 5-qubits, and entangles every pair of qubits according to the complete graph K_5. Each unitary in the controlled unitary grates is a random matrix in $U(2)$.



5. Comparing Circuits and Transpiling Circuits

Suppose we are given the same quantum circuit, represented in two different sets of basis gates, with a transpiler mapping one representation to the other. Our goal is to compare the one and two-parameter persistent homology of these two representations of the circuit. Analyzing the evolution of entanglement at different scales over time can provide insights for finding optimal circuit representations suitable for specific hardware backends or designing quantum Application-Specific Integrated Circuit (ASIC) devices that take into account the topology of interactions between qubits.

However, comparing the persistent homology representations poses a challenge in identifying invariants such as persistence barcodes or diagrams (for one-parameter persistence) and subsequently comparing them. In the case of one-parameter persistence, Bottleneck or Wasserstein distance on persistence diagrams (or barcodes) can be employed. For two-parameter persistent homology, the task becomes more complex.

Fortunately, our second parameter, discrete time, offers a natural way to utilize barcodes and persistence diagrams. Discrete time is essential in physical theories that involve Planck time or in the realm of quantum computing, where circuits can be divided into moments according to the circuit depth.

Using two-parameter persistent homology with discrete time, we can systematically compare the entanglement structure of the two circuit representations and identify the topological features that are preserved or altered during the transpilation process. This information can be used to inform the design of quantum algorithms, optimize circuit representations for specific hardware, or develop ASIC devices tailored to specific quantum computing problems while taking into account the topological interactions between qubits.

5.1. **Code for Transpiling a Circuit.**

Let us look at an example on how to use Qiskit to transpile a circuit:

```
import qiskit
from qiskit import QuantumCircuit, transpile
from qiskit.providers.aer import AerSimulator

def transpile_circuit(circuit, basis_gates):
    simulator = AerSimulator()
    transpiled_circuit = transpile(circuit, simulator, basis_gates=basis_gates)
    return transpiled_circuit

# Example usage:
circuit = QuantumCircuit(2)
circuit.h(0)
circuit.cx(0, 1)

# Available basis gate sets
basis_gate_sets = {
    'Standard': ['u1', 'u2', 'u3', 'cx'],
    'Clifford': ['h', 's', 'sdg', 'cx'],
    'Minimal': ['rx', 'rz', 'cz'],
    'IBM Q': ['id', 'rz', 'sx', 'x', 'cx']
}

# Select a basis gate set
selected_basis_gate_set = 'Minimal'

transpiled_circuit = transpile_circuit(circuit, basis_gate_sets[selected_basis_gate_set])
print(transpiled_circuit)
```

6. Remarks and Code

The Github repo for this code can be found at:
https://github.com/Amelie-Schreiber/persistent_homology_of_entanglement.
This code constructs a perturbed random graph state using controlled random unitary matrix gates for entangling instead of CNOT and controlled-Z gates. It then calculates the density matrix of the quantum state and computes the inverse quantum mutual information (QMI) pairwise distances for each pair of qubits. The code further computes the persistent homology of the QMI distance matrix and creates an interactive 3D plot of the simplicial complex, where the nodes represent qubits and the edges represent simplices. The qubits are labeled by their corresponding numbers, and their distances in the plot are determined by the QMI distance matrix. The interactive plot also includes a slider to adjust the threshold for including edges in the simplicial complex.

Here's a step-by-step summary of what the code does:

(1) Create a perturbed random graph state using controlled random unitary matrix gates.
(2) Compute the density matrix of the quantum state.
(3) Calculate the pairwise inverse QMI distances for the qubits using the density matrix.
(4) Compute the persistent homology of the QMI distance matrix.
(5) Project the high-dimensional QMI distance matrix into a 3D space using multidimensional scaling (MDS).

(6) Visualize the simplicial complex in a 3D interactive plot, with nodes representing qubits and edges representing the one-skeleton of the simplicial complex. The nodes are labeled with qubit numbers, and their distances are determined by the QMI distance matrix.

This code can be used to study and visualize the structure of entanglement in perturbed random graph states. By analyzing the persistent homology and the simplicial complex, researchers can gain insights into the topological properties of the quantum states and explore the relationships between qubits in the system. The interactive plot allows users to adjust the threshold for including edges in the simplicial complex, enabling them to explore different levels of entanglement and their impact on the overall structure.

References

[BL] Magnus Bakke Botnan, Michael Lesnick, *An Introduction to Multiparameter Persistence*, https://arxiv.org/abs/2203.14289.

[Google Quantum AI] Google Quantum AI, *Suppressing quantum errors by scaling a surface code logical qubit*, Nature volume 614, pages 676–681 (2023), https://www.nature.com/articles/s41586-022-05434-1

[HBB] Loic Herviou, Soumya Bera, Jens H. Bardarson, *Multiscale entanglement clusters at the many-body localization phase transition*, https://arxiv.org/abs/1811.01925

[HOST] Heather A. Harrington, Nina Otter, Hal Schenck, Ulrike Tillmann. *Stratifying multiparameter persistent homology* https://arxiv.org/abs/1708.07390

[Ho] M. Hochster, *Topics in the Homological Theory of Modules over Commutative Rings*, CBMS Regional Conference Series, No. 24 (1974).

[LW] Michael Lesnick, Matthew Wright, *Computing Minimal Presentations and Bigraded Betti Numbers of 2-Parameter Persistent Homology*.

[O] Bart Olsthoorn, *Persistent homology of quantum entanglement*, https://arxiv.org/abs/2110.10214

[Rivet] Michael Lesnick, Matthew Wright, Madkour Abdel-Rahman, Anway De, Bryn Keller, Phil Nadolny, Simon Segert, David Turner, Alex Yu, Roy Zhao, *Rivet*, https://github.com/rivetTDA/rivet/

[RC] David W. Romero, Jean-Baptiste Cordonnier, *Group Equivariant Stand-Alone Self-Attention For Vision*, https://openreview.net/forum?id=JkfYjnOEo6M

[AS] Amelie Schreiber, *Multiparameter Persistent Homology: Generic Structures and Quantum Computing*, http://arxiv.org/abs/2210.11433

[AS3] Amelie Schreiber, *Persistent Homology of Language in Transformers*

[AS3] Amelie Schreiber, *Emergent Topology of Ideas in Vision Transformers*

Email address: amelie.schreiber.math@gmail.com