EMERGENT TOPOLOGY OF LANGUAGE FROM ATTENTION

AMELIE SCHREIBER

Abstract. We conduct a persistent homology analysis of the attention mechanism in transformers. Using a distance metric based on the KL-divergence we show how language has an emergent simplicial complex structure. An analysis of how this simplicial complex changes over time, that is, as new tokens are added is an open problem and may provide a way to improve our understanding of both persistent homology and of language as it is modeled in transformers. Analyzing linguistic structures captured by the simplicial complex of persistent homology provides new avenues of research for linguists and those working on transformers for language tasks. We also give some hints at possible applications of the analysis from improving temporal coherence in generative tasks, and improving language translation. For temporal coherence an extension to two-parameter persistent homology is required, but we show how this can be reduced to a one-parameter persistent homology with discrete time slices, making it possible for persistence diagrams and barcodes to be computed. This in turn is key in the potential applications to generative tasks and translation.

Contents

1. Introduction

The emergence of transformer-based models has significantly impacted the field of natural language processing (NLP), establishing new standards in a variety of tasks such as machine translation, sentiment analysis, and text summarization. A critical component of transformer architectures is the attention mechanism, which enables the model to dynamically assess the significance of different tokens in the input sequence. Although these models exhibit remarkable performance, the inner workings of transformers and their attention mechanisms are still not fully understood. A deeper comprehension of their underlying structure may result in further progress in NLP tasks.

In this paper, we examine the attention mechanism in transformer-based models using persistent homology, a computational topology tool adept at analyzing high-dimensional data and detecting topological features that persist across multiple scales. By utilizing a distance metric based on

Date: April 17, 2023.
2010 *Mathematics Subject Classification.* Primary 81P40 55N31 81P68 81P70 Secondary 81P73 81P65 62R40 68T09 .

Kullback-Leibler (KL) divergence, we unveil the presence of a simplicial complex structure in language, providing a fresh outlook on the linguistic structures captured by transformers.

We initially offer an in-depth analysis of the simplicial complex derived from the attention mechanism, discussing its implications for both persistent homology and language modeling in transformers. We also identify several open challenges, such as understanding the simplicial complex's evolution as new tokens are integrated. Addressing these challenges can result in valuable insights and advancements in transformer-based language tasks.

Besides the theoretical analysis, we investigate potential applications of our findings in improving temporal coherence in generative tasks and refining language translation. To tackle temporal coherence, we suggest an extension to two-parameter persistent homology, which can be simplified to a one-parameter approach with discrete time slices. This simplification enables the computation of persistence diagrams and barcodes, essential for the proposed applications in generative tasks and translation.

The contributions of this paper are as follows:

(1) We conduct a thorough examination of the attention mechanism in transformer-based models using persistent homology, uncovering the emergence of a simplicial complex structure in language.
(2) We emphasize open challenges and potential directions for future research in understanding and enhancing transformer models through the study of persistent homology.
(3) We introduce potential applications of our analysis in improving temporal coherence in generative tasks and refining language translation, focusing on extending persistent homology to address temporal aspects.

2. Jensen-Shannon Distance Metric

The Jensen-Shannon distance (JSD) is a metric derived from the Jensen-Shannon divergence (JSDiv), a symmetrized and smoothed version of the Kullback-Leibler (KL) divergence. KL-divergence is a popular measure of dissimilarity between probability distributions and has found widespread application in deep learning.

To define the Jensen-Shannon divergence, we first recall the KL-divergence between two probability distributions P and Q:

$$\mathrm{KL}(P\|Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}, \tag{1}$$

where i ranges over the elements in the support of the distributions. Note that KL-divergence is not symmetric, i.e., $\mathrm{KL}(P\|Q) \neq \mathrm{KL}(Q\|P)$. In order to derive a symmetrized measure, we introduce the Jensen-Shannon divergence:

$$\mathrm{JSDiv}(P, Q) = \frac{1}{2}\mathrm{KL}(P\|M) + \frac{1}{2}\mathrm{KL}(Q\|M), \tag{2}$$

where $M = \frac{1}{2}(P + Q)$ is the average of the two probability distributions. The Jensen-Shannon divergence satisfies the properties of being symmetric, i.e., $\mathrm{JSDiv}(P, Q) = \mathrm{JSDiv}(Q, P)$, and non-negative, i.e., $\mathrm{JSDiv}(P, Q) \geq 0$.

However, the Jensen-Shannon divergence is not a metric, as it does not satisfy the triangle inequality. To obtain a metric, we define the Jensen-Shannon distance as the square root of the Jensen-Shannon divergence:

$$\mathrm{JSD}(P, Q) = \sqrt{\mathrm{JSDiv}(P, Q)}. \tag{3}$$

The Jensen-Shannon distance satisfies the properties of a metric, including non-negativity, symmetry, the identity of indiscernibles, and the triangle inequality. This makes it a useful measure of dissimilarity between probability distributions for various applications, including deep learning.

3. Multi-head Self-Attention

3.1. **Attention and Attention Blocks.** The self-attention mechanism is a critical component of transformer-based models, enabling the model to weigh the importance of different tokens in the input sequence dynamically. In this section, we provide a technical explanation of the self-attention matrix used in these models, which can be defined as:

$$\mathrm{Attn}(X) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right), \tag{4}$$

where Q, K, and V represent the query, key, and value matrices, respectively, and d_k is the dimensionality of the key vectors.

Given an input sequence $X = x_1, x_2, \ldots, x_n$, where each x_i is a token in the sequence, the self-attention mechanism aims to compute a weighted sum of the input representations based on their contextual relevance. To achieve this, the input sequence is first linearly transformed into query, key, and value matrices using separate weight matrices W^Q, W^K, and W^V:

$$Q = XW^Q, \tag{5}$$

$$K = XW^K, \tag{6}$$

$$V = XW^V. \tag{7}$$

The query matrix Q represents the transformed input sequence, and each row in the key matrix K corresponds to the contextual importance of a token with respect to another token. By computing the dot product between the query and key matrices, we obtain an affinity matrix that measures the contextual relevance between each pair of tokens:

$$S = QK^T. \tag{8}$$

To ensure that the self-attention mechanism is invariant to the scale of the input, we normalize the affinity matrix by dividing each element by the square root of the key vector dimensionality, i.e., $\sqrt{d_k}$:

$$\widetilde{S} = \frac{S}{\sqrt{d_k}}. \tag{9}$$

Finally, we apply the softmax function to the normalized affinity matrix row-wise, obtaining the self-attention matrix $\mathrm{Attn}(X)$:

$$\mathrm{Attn}(X) = \mathrm{softmax}(\widetilde{S}). \tag{10}$$

Each row of the self-attention matrix represents the attention weights for a specific token in the input sequence, and these weights are used to compute a weighted sum of the value matrix V, resulting in the final output of the self-attention mechanism.

3.2. **Probability Distributions from Tokens.** Now, taking X_i the i^{th} row of the matrix of token embedding vectors X, we denote by

$$\langle q_i, k_j \rangle = (X_i W^Q)(X_j W^K)^T. \tag{11}$$

Then we have,

$$P(X_i) = \left(\text{softmax}_j\left(\frac{\langle q_i, k_j\rangle}{\sqrt{d}}\right)\right)_{j=1}^{n} = \left(\frac{e^{\frac{\langle q_i, k_j\rangle}{\sqrt{d}}}}{\sum_{l=1}^{n} e^{\frac{\langle q_i, k_l\rangle}{\sqrt{d}}}}\right)_{j=1}^{n} \tag{12}$$

To measure the dissimilarity between the attending behaviors of tokens X_i and X_j, we can compute the Kullback-Leibler (KL) divergence, denoted as $D_{KL}(P(X_i)\|P(X_j))$. The formula for KL divergence is:

$$KL(P(X_i)\|P(X_j)) = \sum_{k=1}^{n} P(X_i)_k \log_2 \frac{P(X_i)_k}{P(X_j)_k} \tag{13}$$

$$= \sum_{k=1}^{n} \frac{e^{\frac{\langle q_i, k_k\rangle}{\sqrt{d}}}}{\sum_{l=1}^{n} e^{\frac{\langle q_i, k_l\rangle}{\sqrt{d}}}} \log_2 \left(\frac{\frac{e^{\frac{\langle q_i, k_k\rangle}{\sqrt{d}}}}{\sum l=1^n e^{\frac{\langle q_i, k_l\rangle}{\sqrt{d}}}}}{\frac{e^{\frac{\langle q_j, k_k\rangle}{\sqrt{d}}}}{\sum_{l=1}^{n} e^{\frac{\langle q_j, k_l\rangle}{\sqrt{d}}}}}\right) \tag{14}$$

$$= \sum_{k=1}^{n} \frac{e^{\frac{\langle q_i, k_k\rangle}{\sqrt{d}}}}{\sum_{l=1}^{n} e^{\frac{\langle q_i, k_l\rangle}{\sqrt{d}}}} \left(\log_2\left(\frac{e^{\frac{\langle q_i, k_k\rangle}{\sqrt{d}}}}{\sum_{l=1}^{n} e^{\frac{\langle q_i, k_l\rangle}{\sqrt{d}}}}\right) - \log_2\left(\frac{e^{\frac{\langle q_j, k_k\rangle}{\sqrt{d}}}}{\sum_{l=1}^{n} e^{\frac{\langle q_j, k_l\rangle}{\sqrt{d}}}}\right)\right) \tag{15}$$

3.3. **Multihead Attention.** Multi-head self-attention is an extension of the self-attention mechanism, which allows the model to capture multiple contextual relationships among tokens in the input sequence simultaneously. The concept of multi-head self-attention involves dividing the original attention mechanism into multiple parallel sub-components, referred to as "heads." Each head computes its own attention matrix and output, which are then combined to form the final output.

Mathematically, the multi-head self-attention can be defined as follows:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h) W^O, \tag{16}$$

where h is the number of heads, and each head_i is the output of the self-attention mechanism for the i-th head:

$$\text{head}_i = \text{Attn}(QW_i^Q, KW_i^K, VW_i^V), \tag{17}$$

with W_i^Q, W_i^K, and W_i^V being the weight matrices for the query, key, and value transformations for the i-th head, respectively. The final output of the multi-head self-attention is obtained by concatenating the outputs of all heads and applying a linear transformation using the weight matrix W^O.

The following transformer-based models are explored in the code for this paper and multi-head self-attention is employed with different configurations of layers and heads in each:

(1) GPT-2 (gpt2): GPT-2 consists of 12 layers in its base configuration, with each layer utilizing 12 attention heads.

(2) DistilBERT (distilbert-base-uncased): DistilBERT is a smaller, distilled version of BERT, comprising 6 layers, with each layer employing 12 attention heads.
(3) BERT (bert-base-uncased): BERT's base configuration consists of 12 layers, with each layer utilizing 12 attention heads.
(4) RoBERTa (roberta-base): RoBERTa is a variant of BERT and has a similar architecture. In its base configuration, it includes 12 layers, with each layer employing 12 attention heads.
(5) Aleph-BeRT (onlplab/alephbert-base): Aleph-BeRT is a Hebrew language model based on BERT. Its base configuration consists of 12 layers, with each layer utilizing 12 attention heads.

The use of multiple layers and heads in these models allows for the extraction of rich contextual information from the input sequences, enabling the models to achieve state-of-the-art performance in a wide range of natural language processing tasks.

4. Persistent Homology

4.1. **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 section, 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:

$$X_a = f^{-1}((-\infty, a]), \tag{18}$$

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

$$X_{a_1} \subseteq X_{a_2} \subseteq \cdots \subseteq X_{a_n}, \tag{19}$$

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.

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.

4.2. **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 4.1. 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 4.2. 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. 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.

4.3. **Discrete Time Reduction.** In the context of the attention mechanism in transformer-based models, we can employ two-parameter persistent homology to study the topological properties of the attention distribution and how they evolve over time. In this case, the two parameters correspond to the Jensen-Shannon distance and time (as discrete steps in the input sequence). This approach provides a richer understanding of the underlying linguistic structures captured by the attention mechanism in transformer models.

The first parameter, Jensen-Shannon distance, is a symmetrized and smoothed version of the Kullback-Leibler divergence, which measures the dissimilarity between two probability distributions. Given the attention mechanism's output, we can view it as a collection of probability distributions across tokens in the input sequence. The Jensen-Shannon distance can be used to construct a distance matrix between all pairs of tokens, which in turn can be used to build a weighted simplicial complex to represent the relationships between tokens.

The second parameter, time, represents the discrete steps in the input sequence. As tokens are added to the input sequence, the attention mechanism updates its representation of the sequence and the relationships between tokens. This time component allows us to capture the temporal evolution of the attention mechanism and investigate how linguistic structures change as the input sequence grows.

To perform a two-parameter persistent homology analysis of the attention mechanism, we first construct double-sublevel sets based on both the Jensen-Shannon distance and time. For each pair of parameters (a, b), we compute the homology groups of the double-sublevel set corresponding to the attention mechanism's output at time b and with a distance threshold of a. This yields a $\mathbb{C}[x, y]$-module, where the vector spaces are the homology groups of the double-sublevel sets, and the linear maps are induced by inclusions between the sets.

Analyzing the structure of this $\mathbb{C}[x, y]$-module can reveal important insights into the topological properties of the attention mechanism in transformer models. For instance, we can identify persistent topological features that are robust to changes in the attention mechanism over time and across various levels of similarity, as measured by the Jensen-Shannon distance. Furthermore, we can investigate how these topological features evolve as new tokens are added to the input sequence, shedding light on the model's ability to capture long-range dependencies and complex linguistic structures.

Two-parameter persistent homology with the Jensen-Shannon distance and time as parameters provides a powerful tool for studying the topological properties and temporal evolution of the attention mechanism in transformer models. This analysis can lead to valuable insights into the

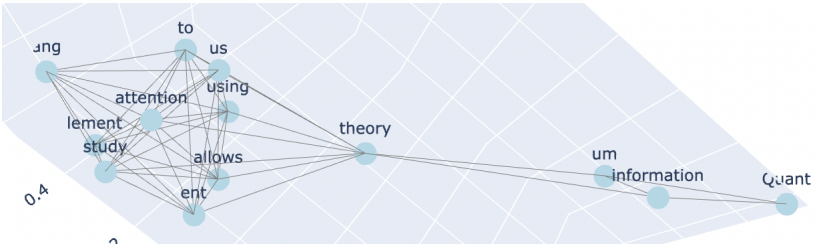
underlying linguistic structures captured by these models and help guide future improvements in transformer-based natural language processing tasks.

5. Examples with 1-parameter Persistent Homology

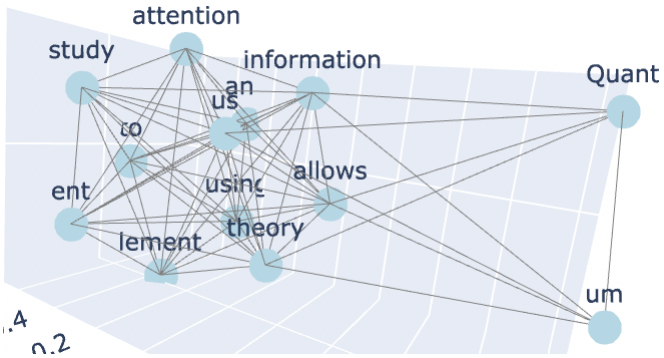
Gudhi, a library for topological data analysis, provides efficient tools to compute persistent homology and persistence diagrams from a distance matrix. In the case of the Jensen-Shannon distance matrix, the matrix is constructed using pairwise distances between probability distributions associated with tokens. Here, we present a step-by-step explanation of how Gudhi computes persistent homology and persistence diagrams for H_0 and H_1, and how it derives a simplicial complex for plotting.

(1) Jensen-Shannon Distance Matrix: Given a set of probability distributions associated with tokens, compute the pairwise distances between these distributions using the Jensen-Shannon distance metric. This results in a symmetric distance matrix.
(2) Vietoris-Rips Complex: From the Jensen-Shannon distance matrix, Gudhi constructs a Vietoris-Rips complex, which is a simplicial complex built from a set of points in a metric space. The complex is constructed by including a simplex for each subset of points whose pairwise distances are less than or equal to a certain filtration value ϵ. As the filtration value increases, the simplicial complex becomes more connected, capturing the underlying structure of the data.
(3) Simplicial Complex Filtration: Gudhi applies a filtration process to the Vietoris-Rips complex by incrementally increasing the filtration value ϵ. At each filtration step, new simplices are added to the complex, and the homology groups are updated. The filtration captures the evolution of topological features in the data, such as connected components (H_0) and loops (H_1), as the filtration value increases.
(4) Persistent Homology Computation: Persistent homology is computed from the filtered simplicial complex. Gudhi tracks the birth and death of topological features throughout the filtration process. For each homology group (H_0, H_1, etc.), Gudhi creates persistence pairs representing the birth and death filtration values of each topological feature. These persistence pairs are then used to compute the persistence diagrams or barcodes.
(5) Persistence Diagrams (or barcodes): For each homology group, Gudhi constructs a persistence diagram, which is a scatter plot of the persistence pairs in the birth-death plane. Points in the persistence diagram represent topological features, and their distance from the diagonal indicates their persistence. Points far from the diagonal correspond to significant topological features that persist across a wide range of filtration values, while points close to the diagonal are considered noise or less significant features.
(6) Simplicial Complex for Plotting: To visualize the structure captured by the persistent homology computation, Gudhi can extract a simplicial complex from the Vietoris-Rips complex at a specific filtration value. This simplicial complex can be plotted to visualize the relationships between tokens in the language model, based on the Jensen-Shannon distance metric.

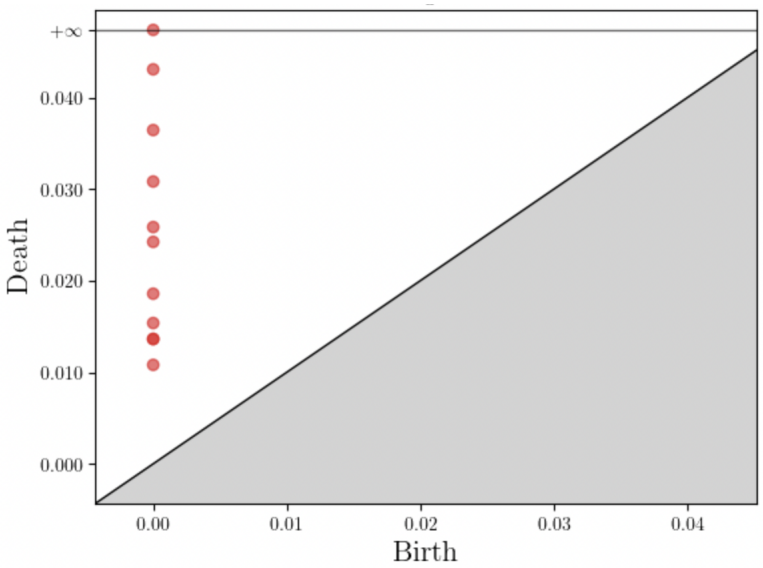
Below, we can see the 1-skeleton of the simplicial complex associated to one of the attention heads (and for some fixed JS-distance thresholds):



For another head and value of the JS-distance threshold we can see the simplicial complex changes:



The persistence diagram corresponding to layer 1 and head 2 of GPT-2 for the input text "Quantum information theory allows us to study attention using entanglement" is pictured below:



6. Applications

6.1. **Information Compression and Reducing Complexity of the Attention Mechanism.** As highlighted in [SJENMC], transformers face two main limitations, one being their quadratic complexity ($O(N^2)$) due to pairwise affinity computation, posing a significant challenge for video applications. Consequently, a method to compress the information needed to represent the attention

mechanism in a transformer is highly desirable. In this context, we explore how persistent homology might aid in achieving this goal. Given a "significant" persistent topological structure, we may want to replace the distributions associated with tokens by the softmax of the attention matrix, using a representative akin to choosing a cluster representative in data or a centroid for probability distributions. Persistent homology allows for the continued use of clusters, but with the added benefit of capturing additional topological features that persist over a wide range of filtration values.

If such a persistent topological feature exists in the attention mechanism for specific input text, we could replace it in a principled manner with one or more representatives, reducing the amount of information necessary to represent the attention mechanism applied to that input text. This concept might be incorporated into model training, potentially leading to more efficient text representation.

6.2. **Temporal Coherence.** Considering the addition of tokens to an input in a sequential fashion as "time" passing, with one discrete time step per token, we can ask questions such as "Which topological features might persist in time?" and "How might the persistent homology for a text corpus change over time?" Utilizing discrete time and associating a persistence diagram with each time segment allows us to model the topology of language as time progresses. This approach introduces an entirely new mathematical perspective for studying linguistics.

6.3. **Language Translation.** Do topological features in language persist after translation into another language? Can we identify persistent homology fingerprints that are similar, or that undergo a principled transformation when translating from one language to another? Are there unique structures specific to certain languages? These are all crucial questions that arise when employing persistent homology to study language through the attention mechanism of transformers. Investigating the topological properties of various models can also reveal how different models represent languages more efficiently than others. In particular, we might discover persistent topological features that exist in some models but not in others, shedding light on their efficiency in representing languages.

6.4. **Loose Connections to Quantum Information Theory.** We note that replacing each probability distribution associated to a token by the attention mechanism with a density matrix (or state vector) as follows:

References

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

[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/

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

[SJENMC] Javier Selva1, Anders S. Johansen, Sergio Escalera1, Kamal Nasrollahi, Thomas B. Moeslund, Albert Clapes, *Video Transformers: A Survey*, https://arxiv.org/pdf/2201.05991.pdf

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