Zigzag Persistence of Large Language Models Representations

Yuri Gardinazzi YURI.GARDINAZZI@PHD.UNITS.IT
AREA Science Park, Trieste; University of Trieste, Trieste

Karthik Viswanathan
AREA Science Park, Trieste; University of Amsterdam, Amsterdam

Giada Panerai
AREA Science Park, Trieste; University of Trieste, Trieste

Alessio Ansuini
Alberto Cazzaniga
Matteo Biagetti
AREA Science Park, Trieste

Editors: Michael Bleher, Freya Jensen, Levin Maier, Diaaeldin Taha, and Anna Wienhard

Abstract

We analyze internal representations of large language models with zigzag persistent homology, treating depth as a discrete time axis for point clouds of last-token embeddings. At each layer we build a k-nearest-neighbors clique complex, connect adjacent layers via intersections, and summarize the resulting diagrams with effective persistence images. From these we derive two descriptors: Births' Relative Frequency (at what rate new p-dimensional features appear) and Inter-Layer Persistence (how long they survive across depth). On the SST movie reviews dataset and three open-source models (Llama-3.1, OSS-20B, Phi-4), we consistently observe three evolving phases: early rapid changes, a middle regime of stable organization, and a final reorganization before output. Using the stability signal (inter-layer persistence) to guide where to remove contiguous blocks of layers, we find that pruning within high-persistence regions maintains 5-shot MMLU performance (with the same trend visible even for the more pruning-sensitive OSS-20B). This suggests that zigzag-based summaries capture meaningful, system-level dynamics and can inform lightweight pruning.

1. Introduction

Large language models implement a deep sequence of representation transformations whose behavior can be understood as a discrete dynamical system evolving across layers (Geshkovski et al., 2024; Vuckovic et al., 2020). From a topological viewpoint, each layer induces a point cloud in representation space, and the model's computation rearranges these clouds over depth. While topological data analysis (TDA) has been used to probe neural representations, most prior work applies persistent homology to single snapshots, then aggregates post hoc across layers (Rieck, Bastian Alexander et al., 2023; Naitzat et al., 2020; Lacombe et al., 2021; Magai and Ayzenberg, 2022). This snapshot paradigm cannot directly express how features form, persist, and disappear along the network's computational trajectory. Zigzag persistence (Carlsson et al., 2009), by contrast, is expressly designed to track homological

structure along sequences of inclusions that can move forward and backward, making it a natural tool for layer-wise dynamics.

This paper introduces a zigzag-persistence pipeline tailored to transformer models and uses it to quantify how topological features evolve across depth, developed by Gardinazzi et al. (2025), and extends the results to more recent models. Concretely, we study last-token embeddings (one point per prompt per layer), build a k-nearest-neighbor (kNN) clique complex at each layer, stitch adjacent layers via intersections to obtain a zigzag filtration, and compute zigzag persistence up to a fixed dimension. The choice of kNN-based complexes rather than Rips across a global distance scale in this context is motivated by the fact that distances and angles can be reparameterized across layers, whereas local neighbor relations remain fairly stable and allow for a more robust comparison.

From this description, we build two main descriptors:

- Births' relative frequency $B_p(\ell)$: the layerwise distribution of births, with power weights that emphasize short- or long-lived features. This summarizes when the model creates new relations in representation space.
- Inter-layer persistence $\bar{\mathcal{Z}}_p(\ell)$: the fraction of features alive at ℓ that also survive across other layers (again power-weighted). This quantifies how stable the model keeps relative positions over depth.

Applied to SST prompts on three open models (Llama-3.1, OpenAI OSS-20B, and Phi-4), our descriptors show a common qualitative pattern across depth. In dimension $p = 1$, Births' Relative Frequency is high and dominated by short-lived features in early layers (rapid local rearrangements), peaks in long-lived structure in the middle (stabilization of relative positions) and shows a final increase in non-persistent births near the output layers (preparation for decoding). While this phase structure is qualitatively shared, its quantitative expression differs by model.

As a showcase downstream task, we use this information to guide layer pruning. Using a sliding-window pruning protocol that drops contiguous blocks of layers and evaluates 5-shot MMLU accuracy, we observe that pruning within the late-middle region of the models causes a negligible drop in performance for Llama-3.1 and Phi-4. For OSS-20B, accuracy is more affected by pruning overall, but we still observe slight gains when pruning overlaps regions of high persistence. These results support the view that zigzag-derived descriptors capture how models manage and preserve relational structure across depth, and that this information can guide simple, unsupervised pruning choices.

2. Method

In this section, we present the zigzag persistence framework used to analyze internal representations of large language models trained with an autoregressive objective. Such models take as input a sequence of n tokens (e.g., a sentence) embedded in $\mathbb{R}^d$ and propagate these embeddings through the layers without changing the ambient dimension. Owing to the autoregressive setup, the representation of the final token encodes information about the

entire sentence and is used to predict the next token. We therefore focus on the last-token representation at each layer. The resulting point cloud consists of last-token embeddings $\mathbf{x}_i(\ell_j) \in \mathbb{R}^d$, for $i = 1, \ldots, N$sentences and $j = 1, \ldots, N_{\text{layers}}$. These embeddings are collected from multiple datasets and serve as probes of how the model processes input across depth.

Our goal is to characterize representation dynamics by tracking statistical changes in the creation and disappearance of p-dimensional holes formed by connecting nearby points within each layer ℓ_i. The first step in the TDA pipeline is a rule for connecting points. We construct, at every layer ℓ_i, a k-nearest-neighbor graph $G_{\ell_i} = (V_{\ell_i}, E_{\ell_i})$, with the number of neighbours fixed as a hyperparameter (see (Le and Taylor, 2024) for prior use of k_{NN}-based filtrations). We then *fill* a simplex when its boundary, composed of lower-dimensional simplices (such as vertices and edges), is complete. In particular, we consider a triangle as filled when it has three vertices with pairwise connections. Similarly, a tetrahedron is filled when four vertices are all interconnected by edges, totaling six edges. This concept extends to higher dimensions up to a specified maximum dimension m. In each layer, we construct the simplicial complex K_{ℓ_i} defined by:

$$K_{\ell_i} = \bigcup_{S \subseteq V_{\ell_i}} \big\{ S \;\big|\; \forall x_s, x_l \in S,\ (x_s, x_l) \in E_{\ell_i} \text{ and } |S| \leq m+1 \big\}.$$

we compute intersection layers by identifying simplices present simultaneously in both adjacent layers. This allows us to construct a sequence of inclusions between these complexes where we define $L \equiv N_{\text{layers}}$ for conciseness:

$$K_{\ell_1} \;\hookleftarrow K_{\ell_1} \cap K_{\ell_2} \;\hookrightarrow\; K_{\ell_2} \;\hookleftarrow\; \cdots \tag{1}$$

We thus define a notion of *birth* and *death* of p-dimensional holes, with $p = 0, ..., m-1$, where m is the maximum dimension to which the complex is expanded. Throughout this work, we choose $m = 4$, which implies that the p-dimensional holes are well defined up to dimension $p = 3$. The output of the zigzag algorithm is then a multiset of birth-death pairs $[\mathrm{b}, \mathrm{d}]$[1], known as the *persistence diagram*:

$$\mathrm{Pers_p}(\Phi) = \Big\{ [\mathrm{b}, \mathrm{d}] \;\;|\;\; \mathrm{b}, \mathrm{d} \in \{0, \ldots, 2(\mathrm{N_{layers}} - 1)\} \Big\}. \tag{2}$$

We index the zigzag filtration by integers $0, 1, \ldots, 2(L-1)$, using even indices for model layers and odd indices for intersection layers. The corresponding zigzag persistence image PI_p lives on a $(2L-1) \times (2L-1)$ birth–death grid. Because model and intersection layers alternate, PI_p varies non-smoothly across depth. To obtain smooth, layer-indexed summaries, we define an effective persistence image $\widehat{PI}_p$ on an $L \times L$ grid over model layers only. Any interval whose birth or death falls on an intersection index is pushed to the adjacent model index; equivalently, $\widehat{PI}_p$ is obtained from PI_p by a local 2×2 aggregation

1. The repetition of a pair $[\mathrm{b}, \mathrm{d}]$ indicates that multiple holes in dimension p have been created and destroyed in correspondence of the same layers.

around each model–model cell (cf. Kim and Mémoli (2017)). From $\widehat{PI}_p$ we derive two layerwise descriptors:

Births' relative frequency $B_p(\ell)$, i.e. the fraction of p-dimensional features born at model layer ℓ, optionally reweighted by a power of lifetime using $\omega(\Delta\ell) = |\Delta\ell|^\alpha$ to emphasize short-lived ($\alpha < 0$) or long-lived ($\alpha > 0$) features;

Inter-layer persistence $\bar{\mathcal{Z}}_p(\ell)$: among p-features alive at layer ℓ, the power-weighted average fraction that also persist to other model layers. Operationally, we normalize $\widehat{PI}_p$ columnwise by the Betti number $\beta_p(\ell)$ and average across rows with the same weight $\omega(\cdot)$. An extended definition of $\widehat{PI}_p$ and of these descriptors is found in Appendix A.2

3. Experiments

Models and Datasets: We analyze different recent open-source transformer models: Llama3.1 (AI@Meta, 2024), OpenAI OSS 20B (OpenAI, 2025) and Phi 4 (Abdin et al., 2024). We extract internal model representations using the Standford Sentiment Treebank (SST), a dataset of movie reviews.[2] To ensure the consistency of our measurements, we extract subsets of 500 prompts from the dataset and compute mean and variance of our descriptors over all the subsets. For the pruning tasks, performance of the outputs is analyzed on the MMLU benchmark (Hendrycks et al., 2021). The evaluation is conducted using the lm-eval-harness library (Gao et al., 2024) in a 5-shot configuration.

Topological Descriptors. We compute the Births' Relative Frequency and Interlayer Persistence for all models over all subsets. We show them in the Top and Middle panel of Fig. 1, respectively. The analysis reveals different phases of input processing for each model characterized by the behavior of 1-dimensional holes across layers.[3] In the *early to middle layers*, a large number of short-lived holes indicates a rapid initial rearrangement of prompts, which is linked to local contextualization and increased dimensionality. This is followed by a stabilization in the *middle layers* where the high probability of long-lived holes suggests that relative positions are kept stable, corresponding to a decrease in dimensionality. In the last layers, a strong, final rearrangement is marked by a surge in new but non-persistent 1-dimensional holes as the model formats its required output.

Sliding Window and Pruning Layers. To test the hyphothesis of the input rearrangement that corresponds to the persistence of 1-cycles, we prune consecutive layers in blocks and evaluate performance on a benchmark after every pruning. We show performance results for this pruning in the Bottom Panel of Fig. 1. We observe that 5-shot MMLU performance is significantly better preserved when the pruning window is located in the second half of the model's depth, particularly for Phi-4 and Llama-3.1, coherently with a higher rate of persisting features. For the model OSS 20B, the overall decrease in performance is

2. Note that a larger set of models and datasets was analysed in a similar fashion in a recent paper by the same authors (Gardinazzi et al., 2025).
3. We only show results for $p = 1$ as it provides the highest number of features, making the result more statistically stable. Other dimensions show a generally lower and less stable number of features. For a discussion about how this connects to our choice of filtation, and further discussion for $p = 0, 2, 3$ see (Gardinazzi et al., 2025).

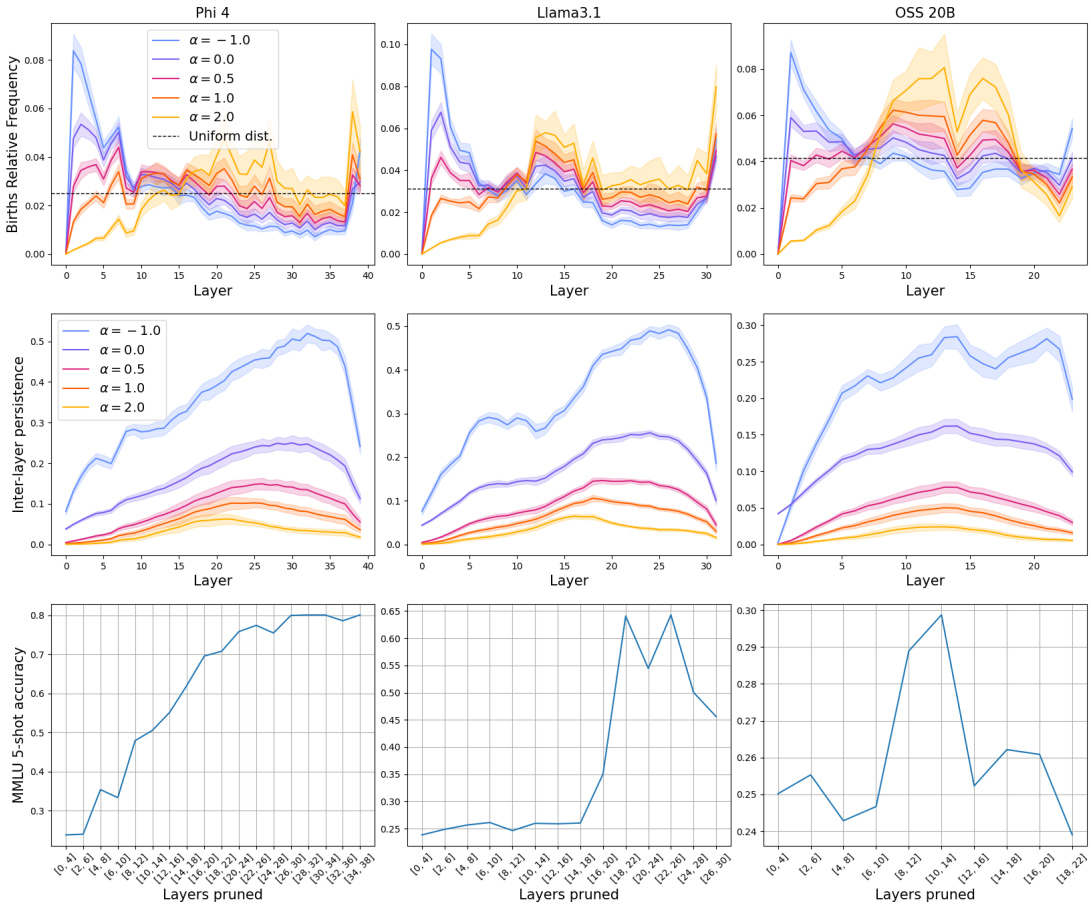

Figure 1: *First row:* Births' Relative Frequency, B_1, as a function of model's layers on the SST dataset, for varying α. The dashed line represents a uniform distribution of births across the layers. *Second row:* Inter-Layer Persistence as a function of model's layers on SST dataset, for varying α. *Third row:* Layer pruning with sliding window.

higher and only slightly above chance. This might be caused by the characteristic Mixture of Expert architecture of the model, making it particularly sensitive to layer pruning. Nevertheless, a slight increase in performance is observed when pruning layers with higher rate of persistence even in this case.

4. Conclusion

We introduced a zigzag-persistence framework for tracking the evolution of large language models representations across layers and summarized the resulting dynamics with two interpretable descriptors: Births' Relative Frequency and Inter-Layer Persistence. On SST prompts and three open models, these summaries reveal a consistent three-phase computation pattern and provide actionable signals for pruning. In sliding-window experi-

ments, pruning contiguous late-middle blocks identified by elevated Inter-Layer Persistence maintains or improves 5-shot MMLU accuracy for Llama-3.1 and Phi-4, and offers modest benefits even for the more pruning-sensitive OSS-20B. Overall, the results indicate that zigzag-based topological summaries capture system-level organization in transformer depth and can inform lightweight compression heuristics. Future work will investigate how topology interacts with architectural choices and training regimes.

ACKNOWLEDGMENTS

M.B. Y.G. and G.P. are partially supported by the Programma Nazionale della Ricerca (PNR) grant J95F21002830001 with title "FAIR-by-design". K.V. was partially supported by Programma Nazionale della Ricerca (PNR) grant J95F21002830001 with the title "FAIR-by-design" during his visit to Area Science Park while this project was in its development phase. A.A. and A.C. were supported by the project "Supporto alla diagnosi di malattie rare tramite l'intelligenza artificiale" CUP: F53C22001770002. A.A. and A. C. were supported by the European Union – NextGenerationEU within the project PNRR "PRP@CERIC" IR0000028 - Mission 4 Component 2 Investment 3.1 Action 3.1.1.

We thank Area Science Park supercomputing platform ORFEO made available for conducting the research reported in this paper and the technical support of the Laboratory of Data Engineering staff.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Xin Wang, Rachel Ward, Yue Wu, Dingli Yu, Cyril Zhang, and Yi Zhang. Phi-4 technical report, 2024. URL https://arxiv.org/abs/2412.08905.

AI@Meta. Llama 3 model card. 2024. URL https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md.

Gunnar E. Carlsson, Vin de Silva, and Dmitriy Morozov. Zigzag persistent homology and real-valued functions. In *SCG '09*, 2009. URL https://api.semanticscholar.org/CorpusID:5801261.

Tamal K. Dey and Tao Hou. Fast Computation of Zigzag Persistence. In Shiri Chechik, Gonzalo Navarro, Eva Rotenberg, and Grzegorz Herman, editors, *30th Annual European Symposium on Algorithms (ESA 2022)*, volume 244 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 43:1–43:15, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISBN 978-3-95977-247-1. doi: 10.4230/LIPIcs.ESA.2022.43. URL https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ESA.2022.43.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle

McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, 07 2024. URL https://zenodo.org/records/12608602.

Yuri Gardinazzi, Karthik Viswanathan, Giada Panerai, Alberto Cazzaniga, Matteo Biagetti, et al. Persistent topological features in large language models. In *Forty-second International Conference on Machine Learning*, 2025.

Borjan Geshkovski, Cyril Letrouit, Yury Polyanskiy, and Philippe Rigollet. A mathematical perspective on transformers, 2024. URL https://arxiv.org/abs/2312.10794.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

Woojin Kim and Facundo Mémoli. Stable signatures for dynamic graphs and dynamic metric spaces via zigzag persistence. *arXiv: Algebraic Topology*, 2017. URL https://api.semanticscholar.org/CorpusID:44017453.

Théo Lacombe, Yuichi Ike, Mathieu Carriere, Frédéric Chazal, Marc Glisse, and Yuhei Umeda. Topological uncertainty: Monitoring trained neural networks through persistence of activation graphs. *arXiv [stat.ML]*, May 2021.

Minh Quang Le and Dane Taylor. Persistent homology with k-nearest-neighbor filtrations reveals topological convergence of pagerank. *Foundations of Data Science*, 2024. doi: 10.3934/fods.2024038. URL https://www.aimsciences.org/article/id/66c30c8be7a25d6c964d771b.

German Magai and A Ayzenberg. Topology and geometry of data manifold in deep learning. *ArXiv*, abs/2204.08624, April 2022.

Gregory Naitzat, Andrey Zhitnikov, and Lek-Heng Lim. Topology of deep neural networks. *Journal of Machine Learning Research*, 21(184):1–40, 2020. URL http://jmlr.org/papers/v21/20-345.html.

OpenAI. gpt-oss-120b and gpt-oss-20b model card, 2025. URL https://arxiv.org/abs/2508.10925.

Rieck, Bastian Alexander, Togninalli, Matteo, Bock, Christian, Moor, Michael, Horn, Max, Gumbsch, Thomas, and Borgwardt, Karsten. Neural persistence: A complexity measure for deep neural networks using algebraic topology. 2023. doi: 10.3929/ETHZ-B-000327207. URL http://hdl.handle.net/20.500.11850/327207.

James Vuckovic, Aristide Baratin, and Remi Tachet des Combes. A mathematical theory of attention, 2020. URL https://arxiv.org/abs/2007.02876.

Appendix A. Methodological Details

A.1. Zigzag Persistence: Mathematical and Computational Background

Zigzag persistence is an extension of classical persistent homology equipped to handle filtrations that progress forward and backward, thus capturing topological changes when the underlying data structure changes non-monotonically across a sequence (layers, time steps, etc).

A *zigzag filtration* is a sequence of simplicial complexes

$$K_1 \longleftrightarrow K_2 \longleftrightarrow \cdots \longleftrightarrow K_L, \tag{3}$$

where the arrows may alternate direction (either inclusions or co-inclusions). In our layer-wise setting, K_i are complexes built for each LLM layer (see main text), with intersections $K_i \cap K_{i+1}$ building the backward inclusions such that the filtration is:

$$K_1 \hookleftarrow K_1 \cap K_2 \hookrightarrow K_2 \hookleftarrow K_2 \cap K_3 \hookrightarrow \cdots \hookrightarrow K_L$$

Given this sequence, applying the pth homology functor H_p (over a field) yields a zigzag module (sequence of vector spaces with linear maps). Such modules admit a decomposition into intervals $[b, d]$, corresponding to features that are "born" at index b and "die" at d. The multiset of these intervals forms the zigzag persistence diagram.

Our approach constructs K_i at each model layer as a k-nearest neighbor (kNN) graph complex:

$$K(\ell_i) = S \subset V_i : |S| \leq m + 1, \quad (v_s, v_l) \in E_i \;\; \forall v_s, v_l \in S \tag{4}$$

where V_i is the set of last-token embeddings at layer ℓ_i, E_i are kNN edges, and m is the maximal simplex dimension (typically $m = 4$).

A.2. Implementation Details: Construction and Computation

Step-by-step pipeline:

1. **Extract last-token representations** from N prompts for all L transformer layers, yielding $N \times L$ vectors in $\mathbb{R}^d$.
2. **Construct a kNN graph** for each layer ℓ_i (edges E_i).
3. **Build maximal simplices (clique complex):** Fill all cliques up to order $m + 1$ for each graph to get K_i.
4. **Compute intersection complexes:** For each pair (K_i, K_{i+1}), compute $K_i \cap K_{i+1}$.
5. **Assemble the zigzag filtration:** Interweave K_i and $K_i \cap K_{i+1}$ as in above equation.

6. **Run zigzag persistent homology:** Build the filtration using the open source code Dionysus2[4] and FastZigZag (Dey and Hou, 2022) to obtain the full set of p-dimensional birth–death intervals Pers_p.
7. **Build (effective) persistence images and descriptors:** See below and main text Sec 2.

A.3. Effective Persistence Image Transformation

As in the main text, the raw zigzag diagram yields a persistence image on a $(2L-1)\times(2L-1)$ birth–death grid (including both model and intersection layers). To facilitate layer-level summaries, we define the effective persistence image $\widehat{PI}_p$ as follows:

For model layers indexed by even integers and for any pixel $(b/2, d/2)$ on an $L \times L$ grid (with b, d even),

$$\widehat{PI}_p(b/2, d/2) = PI_p(b, d) + PI_p(b-1, d) + PI_p(b, d-1) + PI_p(b-1, d-1) \tag{5}$$

where $PI_p(b, d)$ is the zigzag persistence image, and pixels associated to intersection layers are collapsed to the adjacent model-layer indices (see also Kim and Mémoli (2017)).

Formally, for each interval $[b, d]$ in the zigzag diagram, if b or d falls on an intersection, one shifts it to the nearest model layer above. This transformation yields an $L \times L$ grid where births and deaths are aligned with actual transformer layers.

A.4. Formal Descriptor Definitions

Given the effective persistence image $\widehat{PI}_p$, we define our descriptors:

- **Births' Relative Frequency**:

$$B_p(\ell) = \frac{\sum_{\ell_i} \omega(\ell, \ell_i)\, \widehat{PI}_p(\ell, \ell_i)}{\sum_{\ell_i} \omega(\ell, \ell_i) \sum_{\ell_i} \widehat{PI}_p(\ell, \ell_i)}, \tag{6}$$

  where

$$\omega(\ell, \ell_i) = |\ell - \ell_i|^\alpha \tag{7}$$

  weights by feature lifespan ($\alpha > 0$: long-lived emphasis; $\alpha < 0$: short-lived emphasis).

- **Inter-layer Persistence:**

$$\mathcal{Z}_p(\ell, \ell') = \frac{\sum_{\ell \le M_1, \ell' > M_2} \widehat{PI}_p\,(\ell, \ell')}{\beta_p(\ell)}, \tag{8}$$

  where $M_1 = \min(\ell, \ell')$; $M_2 = \max(\ell, \ell')$ and $\beta_p(\ell)$ is the pth Betti number at layer ℓ. We summarize by power-weighted averaging across ℓ':

$$\bar{\mathcal{Z}}_p(\ell) = \frac{\sum_{\ell_i=1}^{N_{\text{layers}}} \omega(\ell, \ell_i)\, \mathcal{Z}_p(\ell, \ell_i)}{\sum_{\ell_i=1}^{N_{\text{layers}}} \omega(\ell, \ell_i)} \tag{9}$$

4. https://www.mrzv.org/software/dionysus2/

following the same weighting ω.

These descriptors are well-defined provided that $\beta_p(\ell) > 0$ (otherwise set to 0).

A.5. Summary of Algorithmic Complexity

The computational cost for the pipeline is dominated by

$$\begin{aligned} &\mathcal{O}(n^2 \cdot L) && \text{for } k\text{NN construction across } L \text{ layers} \\ &+ \mathcal{O}(m^\omega) && \text{for zigzag reduction; with } \omega < 2.373 \end{aligned}$$

where n is the number of prompts and m the maximal simplex order. In practice, fast C++/Python implementations (using FastZigZag + Dionysus2) make the pipeline feasible for $n \sim 10,000$ points and $L = 32$ layers.

A.6. Code and Reproducibility

All code to produce the zigzag filtrations, persistence diagrams, effective images, and descriptors is available at https://github.com/RitAreaSciencePark/ZigZagLLMs.