Zigzag Persistence on Dynamic Graphs: Oracle-Based Verification and the Blind Spot of Event-Level Metrics

Semen A. Kurkin
Plekhanov Russian University of Economics
Moscow, Russia
0000-0002-3438-5717

Alexander K. Kuc
Plekhanov Russian University of Economics
Moscow, Russia
0000-0001-6233-2594

Vladimir S. Khorev
Plekhanov Russian University of Economics
Moscow, Russia
0000-0001-6613-8940

Aleksandra Yu. Antipova
Plekhanov Russian University of Economics
Moscow, Russia

Timofey Yu. Vedeneev
Plekhanov Russian University of Economics
Moscow, Russia

Zolzaya Dashdorj
Mongolian University of Science and Technology
Ulaanbaatar, Mongolia
0000-0001-6367-9039

2026 10th Scientific School Dynamics of Complex Networks and their Applications (DCNA) | 979-8-3195-2599-4/26/$31.00 ©2026 IEEE | DOI: 10.1109/DCNA72004.2026.11733455

***Abstract*—Zigzag persistence summarises a time-ordered sequence of graphs by a collection of bars. Each bar is the lifetime of one topological feature (a connected component or an independent cycle) and therefore records both the moment the feature appeared and the moment it disappeared. A sequence of Betti numbers reports only how many features exist at each moment; the barcode says in addition which disappearance belongs to which appearance.**

Programs that compute such barcodes are normally validated on synthetic graphs whose events are scheduled in advance, and scored by recall, precision and localisation error. We built such a program and report two findings about that practice. The first concerns how the program itself is checked. Instead of comparing it against a second library, we recompute the barcode directly from the mathematical definition of the invariant; over the two-element field this requires only ranks of Boolean matrices and no persistence software at all. On 18 benchmarks the two computations agree on every one of 3192 bars. The second concerns what the score proves. We exhibit two different barcodes of the same benchmark that produce identical Betti curves and identical event logs; recall and precision cannot distinguish them, and so cannot confirm the pairing of births with deaths — the one piece of information a barcode adds to Betti numbers.

***Index Terms*—zigzag persistence, dynamic graphs, topological data analysis, verification, generalised rank invariant, benchmark design**

I. Introduction

A dynamic graph is a network observed repeatedly over time. The case that motivates this work is the sliding-window functional connectome, in which a sequence of correlation matrices is thresholded into a sequence of graphs [1]–[3]. Functional networks of this kind are built from fMRI [4], [5], EEG [6] and fNIRS [7] recordings, and brain activity is routinely followed through the course of cognitive tasks and training [8]–[11]. The natural object of study is then a network that changes in time, and a natural question is when its connected components and cycles appear and disappear.

Ordinary persistent homology cannot answer that question, because it assumes a filtration: a family of simplicial complexes in which each is contained in the next. Between two consecutive snapshots of a dynamic graph some edges appear while others vanish, so neither snapshot is contained in the other and no filtration exists. Zigzag persistence removes this obstacle [12], [13]. An intermediate object is placed between every pair of neighbours in such a way that both neighbours are contained in it; the inclusions then alternate in direction, which is what the name refers to. The resulting diagram is a representation of a quiver of type A_N, its indecomposable summands are interval modules by Gabriel's theorem [14], [15], and by the Krull–Schmidt theorem their multiplicities are unique. The barcode is therefore determined by the data alone and not by the algorithm that computes it.

This guarantee concerns the mathematical object, not any particular program. How programs are actually validated, and how work of this kind is judged, is by event recovery: one synthesises dynamic graphs in which births and deaths are scheduled in advance, runs the program, and reports recall, precision and localisation error against the schedule. The same protocol, with expert annotations in place of a schedule, is standard in automatic seizure detection from EEG [16], [17]. The present paper asks what that protocol does and does not establish.

Our contributions are methodological rather than theoretical. (i) We describe a way to verify an implementation that does not rely on a second implementation: the barcode is

Authorized licensed use limited to: Kant Baltic Federal Univ. Downloaded on October 23,2026 at 15:25:30 UTC from IEEE Xplore. Restrictions apply.

recomputed from the definition of the invariant, so that the library is checked against algebra. (ii) We show that recall and precision cannot detect an error in the pairing of births with deaths, and give an explicit instance of such an error in a standard benchmark. (iii) We measure what it costs to have no significance criterion for zigzag bars: on a null model the pipeline reports an event on 98.3% of snapshots, and under mild edge noise its precision falls to 0.17%.

II. Construction and Conventions

A. From snapshots to a zigzag

A dynamic graph is a sequence of snapshots $G_0, \dots, G_{T-1}$ on a fixed vertex set. Which pairs of vertices count as connected is decided by the data and not by the program reading it: in a similarity snapshot an edge appears when $w_{ij} > t$, in a distance snapshot when $d_{ij} \le \varepsilon$. The two rules run in opposite directions, and a program that applies the wrong one silently receives the complement of the intended graph, which is likewise connected and likewise yields plausible Betti numbers. Even with the correct rule the chosen level matters: the topology of thresholded fMRI networks changes qualitatively as the threshold window moves across the range of connection strengths [5].

Each snapshot becomes a simplicial complex $K_i = \mathrm{Fl}_{\le k_{\max}+1}(G_i)$, the flag complex of G_i truncated one dimension above the homology one intends to compute. The extra dimension is necessary: without the boundary operator ∂_{k+1} the k-th Betti number comes out too large. The zigzag is then

$$K_0 \hookrightarrow K_0 \cup K_1 \hookleftarrow K_1 \hookrightarrow \cdots \hookleftarrow K_{T-1}.$$

We index it by $I = \{0, \frac{1}{2}, 1, \dots, T-1\}$, where an integer index refers to a snapshot and a half-integer index to the object between two snapshots.

B. Choosing the intermediate object

The construction requires an object containing both K_i and K_{i+1}, and there are two obvious candidates: the union of the two complexes, and the flag complex of the union of the two graphs. They are not the same. Intersections behave well, $\mathrm{Fl}(G_i) \cap \mathrm{Fl}(G_j) = \mathrm{Fl}(G_i \cap G_j)$, but for unions one has only the inclusion $\mathrm{Fl}(G_i) \cup \mathrm{Fl}(G_j) \subseteq \mathrm{Fl}(G_i \cup G_j)$, and it is usually strict: if the three edges of a triangle are distributed between the two snapshots, the triangle belongs to the flag complex of the union graph and to neither of the two complexes.

We take the union of the complexes. The reason is that the intermediate object must not contain a simplex that exists in no snapshot. If it does, a triangle present in neither G_i nor G_{i+1} fills a cycle at the half-integer index, and the barcode gains a birth and a death that correspond to no state of the data. Such a barcode is internally consistent, so no test based on it will object.

We also measured the difference between the two candidates and found it empty on all 59 transitions of both benchmarks. The choice rests on the argument above rather than on that measurement, but the measurement is worth recording, since a dataset with a different geometry may well separate the two constructions.

C. Why bar endpoints are sometimes half-integers

Endpoints of bars do not always fall on snapshots, and the reason is worth stating because it looks at first like an error of half a step.

Closing a path into a cycle at snapshot 12 *adds* an edge, so $K_{11} \subseteq K_{12}$ and the intermediate object $K_{11\frac{1}{2}}$ coincides with K_{12}. The cycle is already present there, and the class is therefore born at $11\frac{1}{2}$ rather than at 12. Deleting an edge reverses the situation: at snapshot 41 one has $K_{41} \subseteq K_{40}$, the intermediate object coincides with the *earlier* snapshot, and the dying class survives until $40\frac{1}{2}$. For H_0 the two roles are exchanged, since a component is born when an edge is deleted and dies when one is added, and the endpoints come out integral. A half-integer endpoint is thus neither an artefact nor a convention, but a consequence of the direction in which the transition is monotone (Fig. 1).

D. Translating bars into events

Bars live on I, whereas a ground-truth log speaks of snapshots, so a translation is needed. It consists of two rules, and they are not symmetric: a birth is reported at the first snapshot index *at or after* the birth of the bar, a death at the first snapshot index *strictly after* the death of the bar. The familiar shift $d \mapsto d+1$ is what the second rule gives when the endpoint is an integer; it is not a separate rule, and it is implemented in exactly one place. The care is warranted, because an error of one snapshot here shows up in a report as a systematic localisation error of the algorithm and is indistinguishable from one.

An earlier version of our specification treated half-integer endpoints as a third kind of outcome, an "event inside a transition". Measurement rejected that rule: *both* H_1 bars of the combinatorial benchmark have half-integer endpoints, while the ground-truth log records ordinary births and deaths at snapshots 12 and 41.

III. Checking the Computation Against the Definition

The order in which the components were built is the decision everything else rests on: the independent check was written before any persistence library was used. A second library is not a check. It implements the same class of algorithm, and when two libraries disagree neither result tells which of them is wrong. Worse, a library adopted first tends to become its own reference, so that the first mistake in converting indices is recorded as ground truth and never questioned again.

We therefore compute the barcode a second time straight from the definition of the invariant. For $b \le d$ in I, restrict the module V to the interval $[b, d]$ and let

$$r(b,d) = \mathrm{rank}\big(\varprojlim V|_{[b,d]} \to \varinjlim V|_{[b,d]}\big)$$

be the generalised rank of that restriction [18]. Because every module over A_N decomposes into intervals, $r(b,d)$ is exactly

Authorized licensed use limited to: Kant Baltic Federal Univ. Downloaded on October 23,2026 at 15:25:30 UTC from IEEE Xplore. Restrictions apply.

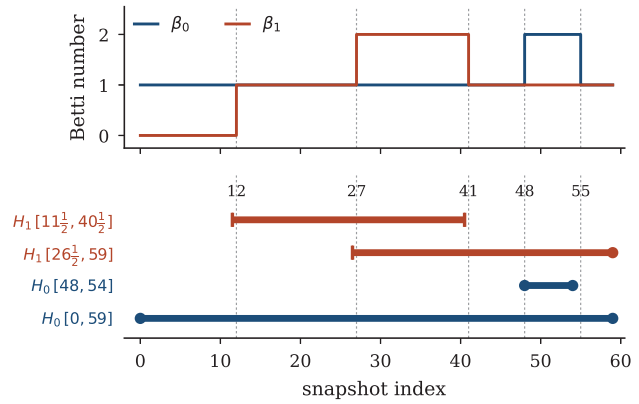


Fig. 1. Barcode of the `cycle_lifecycle` benchmark (24 vertices, 60 snapshots, 5 scheduled events) shown against its Betti curves. Round caps mark integer endpoints and vertical strokes half-integer ones. Three of the five events are derived from half-integer endpoints.

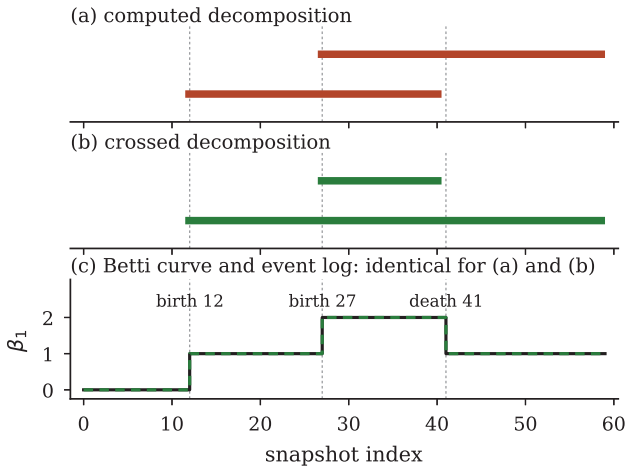


Fig. 2. Two decompositions of the H_1 module of the same benchmark. (a) is the one the algorithm computes; (b) crosses the pairing. The two agree on the Betti curve, on the number of bars crossing every arrow, and on the event log (c). No event-level metric distinguishes them.

the number of bars that contain $[b, d]$; the number of bars equal to $[b, d]$ then follows by inclusion–exclusion,

$$m_{[b,d]} = r(b, d) - r(b^-, d) - r(b, d^+) + r(b^-, d^+),$$

where b^- and d^+ are the neighbouring indices. Over the field with two elements the limit and the colimit are the kernel and the cokernel of a Boolean matrix, so the entire computation reduces to ranks of such matrices and uses no persistence software. Neither the generalised rank nor the inversion formula is new; what is new here is using them as an independent check on a library.

The independent computation was validated first on small zigzags whose barcodes can be written out by hand, among them one built as a miniature of the ambiguity discussed in Section IV. It was then compared, bar for bar and with multiplicities, against Dionysus 2 [19]: **18 benchmarks, 3192 bars, 0 discrepancies**. Restricting the comparison to the two clean benchmarks would have been the weakest available version of this claim: the faults one expects in the adapter (the order of elementary operations, the intervals during which a simplex is present, the conversion of half-open endpoints) show themselves on a barcode with nine hundred bars rather than on one with four, so the sixteen noisy benchmarks carry most of the comparison and most of its running time.

IV. What Event-Level Metrics Cannot Verify

This section presents two elementary observations that together limit what the standard validation protocol can establish.

Counting live classes does not determine the barcode. For an index x let $c(x)$ be the number of bars covering x; this is the Betti number at x, and the function c is the Betti curve. Let $c(e)$ be, similarly, the number of bars covering an arrow e of the zigzag. Consider the two collections of bars $\{[1, 4], [2, 3]\}$ and $\{[1, 3], [2, 4]\}$. They have the same $c(x) = (1, 2, 2, 1)$ on the four indices and the same $c(e) = (1, 2, 1)$ on the three arrows, yet they are different barcodes. Knowing how many classes are alive at each index, and how many survive each arrow, fixes the number of births and deaths everywhere but leaves open which birth is matched with which death.

Neither does the event log. The benchmark provides a concrete instance. After a chord is added at snapshot 27 the space H_1 is two-dimensional, and after an edge is deleted at snapshot 41 it is one-dimensional again, so exactly one of the two classes dies at 41. Which one is decided by the decomposition of the module and not by the age of the class: what matters is which line of the two-dimensional space is the image of the class that survives. The computed answer is $\{[11\frac{1}{2}, 40\frac{1}{2}],\ [26\frac{1}{2}, 59]\}$, so the class born at 12 is the one that dies. The intuitive guess, that breaking one of the two cycles created by the chord must kill the younger class, is wrong. Now take the crossed alternative $\{[11\frac{1}{2}, 59],\ [26\frac{1}{2}, 40\frac{1}{2}]\}$. It has the same Betti curve, the same number of bars on every arrow, and, under the translation rules of Section II-D, the same event log, $\{(12, \text{birth}), (27, \text{birth}), (41, \text{death})\}$ (Fig. 2).

Recall and precision computed from event logs therefore cannot certify the pairing of births with deaths, and the pairing is precisely what zigzag persistence contributes beyond a sequence of Betti numbers. What does certify it is the bar-by-bar comparison of Section III. The loss accumulates along the reporting chain: Betti curve, then arrow counts, then event log, with the pairing discarded again at each step.

A further remark concerns localisation error. With tolerance $\tau = 0$ only an exact hit counts as a match, so the error is identically zero and conveys nothing; we report it because the specification asks for it. Raising τ to one snapshot across the sixteen noisy benchmarks recovers one additional event in one of them: failure under noise is a matter of missed and spurious events, not of events reported at slightly wrong times.

V. Measured Behaviour

A. Benchmarks without noise

Table I reports the standard score together with four further quantities. The perfect scores are the expected outcome rather

Authorized licensed use limited to: Kant Baltic Federal Univ. Downloaded on October 23,2026 at 15:25:30 UTC from IEEE Xplore. Restrictions apply.

TABLE I
EVENT DETECTION AND ADDITIONAL QUANTITIES ON THE BENCHMARKS WITHOUT NOISE. MATCHING TOLERANCE $\tau = 0$.

	`cycle_lifecycle`	`geometric_loop`
snapshots / vertices	60 / 24	60 / 24
scheduled events	5 (H_0:2, H_1:3)	2 (H_1:2)
recall / precision	1.000 / 1.000	1.000 / 1.000
localisation error	0	0
bars in barcode	4	3
invisible bars	0	0
censored endpoints	3	4
events from half-integer	3 of 5	2 of 2
ephemeral (start+rest)	23=23+0	165=71+94

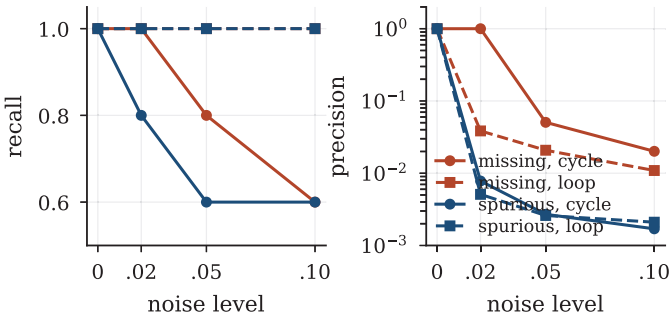


Fig. 3. Degradation under edge noise for the combinatorial (*cycle*) and metric (*loop*) benchmarks. Note the logarithmic precision axis.

than an achievement: the benchmarks were built so that the correct answer is known in advance, and the algorithm is exact on them. Their usefulness lies in what happens once noise is added.

The four further quantities are reported rather than discarded, because each is a property of the construction and not an incidental detail. An *invisible* bar contains no snapshot index in its support: the corresponding class exists only on intermediate objects, so it is a false positive by construction. A *censored* endpoint touches the boundary of the observation window; the associated events are excluded from the score, because a censored birth does not assert that a class was born, only that it already existed when observation began. The count of *events derived from half-integer endpoints* indicates how much of the answer depends on the choice of intermediate object. An *ephemeral* bar is born and dies within a single transition; such bars never appear at any index of the diagram, and become visible only when the library is given one time index per elementary operation rather than per index of I. Their number is reported in two parts, because building the first complex from the empty complex creates 24 components and destroys 23 of them, and those 23 reflect the order of insertion rather than the data. Merging them with the rest would put the zero point in the wrong place and understate any growth under noise.

B. Null model

A null model has no schedule and an empty ground-truth log, so every event the program reports is a false positive. The appropriate answer is not the assertion that there are none, but a frequency stated together with the number of trials and a confidence interval. On Erdős–Rényi graphs at the edge density of the combinatorial benchmark, with 3 seeds and 20 realisations each, the program reports at least one event on 98.3% of snapshots (3540 of 3600; 95% Wilson interval [20] $[0.979, 0.987]$) and on 100% of realisations (60 of 60, $[0.940, 1.000]$), at an average of 12.9 events per snapshot. We give both denominators because neither is satisfactory alone: consecutive snapshots share edges and so are not independent trials, whereas realisations are independent but few.

This is not a defect of the algorithm. A random graph at this density genuinely has $\beta_0 = 4 \ldots 5$ and $\beta_1 = 2 \ldots 5$ on every snapshot, and since edges are redrawn independently at every step, classes really are created and destroyed all the time. The program reports what is there.

C. Degradation under noise

Fig. 3 shows the response to two mechanisms of edge noise at four levels. Recall declines slowly, whereas precision collapses, reaching 0.0017. The way the method fails under noise is thus by flooding the log with spurious events, not by misplacing the true ones in time. Seizure detection on long EEG recordings meets the same asymmetry, and there a stage of outlier detection placed before a neural classifier trades a little recall for a large gain in precision [17].

Added edges are an order of magnitude more damaging than removed ones, which is to be expected: an added edge creates new cycles, whereas a removed edge usually only breaks a cycle that already existed. The metric benchmark keeps recall at unity at every level and loses only precision; this is a property of that particular benchmark, and a summary averaged over all sixteen noisy datasets would have concealed it. At noise level 0.0 the barcode reproduces the clean one exactly, which is a check that costs nothing and catches several kinds of wiring error at once.

Taken together with the null model, these measurements say that in the absence of a significance criterion for zigzag bars, an event log cannot be trusted on data for which no ground truth exists.

VI. LIMITATIONS

The barcode depends on a convention. Which intermediate object is used (Section II-B) changes the barcode, so barcodes produced under different conventions cannot be compared. We therefore store the convention inside the result file next to the bars, rather than leaving it to be inferred from the version of the code that produced them.

Homology is computed only up to a chosen dimension. Complexes are built to dimension $k_{\max} + 1$, which makes the bars in dimension $k_{\max}$ trustworthy. Higher dimensions are not computed, and their absence from the output is not a statement about the data. On the metric benchmark the library returns

Authorized licensed use limited to: Kant Baltic Federal Univ. Downloaded on October 23,2026 at 15:25:30 UTC from IEEE Xplore. Restrictions apply.

69 bars in H_2, which our adapter discards. We know these are artefacts only because the generator of that benchmark built its own complexes one dimension higher and recorded $\beta_2 = 0$ on every snapshot.

There is no significance criterion. Criteria calibrated for static point clouds do not carry over: the distribution of pairwise distances differs, and such criteria are not invariant under reparametrisation of the filtration although the quantity they compare is exactly the one reparametrisation distorts. We offer no replacement. Statistical treatments of the separation between signal and noise in diagrams [21], [22] are one place to look, and extreme value theory, which identifies rare events as outliers of a background distribution without labelled data [16], is another. Section V indicates the level any such criterion has to beat.

Transfer to dense networks is untested. For resting-state fMRI and EEG networks of the kind used in studies of depression and autism [4], [6], the quantities to watch are the invisible and ephemeral bar counts of Table I. Both are false positives generated by the intermediate-object construction itself, and both grow as snapshots become denser.

VII. Conclusion

We built a zigzag persistence pipeline for dynamic graphs and validated it by the criterion ordinarily applied to such work. On two benchmarks with scheduled events it attains perfect recall and precision with zero localisation error. Our argument is that this result, on its own, establishes less than it appears to. Metrics computed from event logs are blind to the pairing of births with deaths, and that pairing is what separates a temporal persistence diagram from a sequence of Betti numbers: we exhibited two barcodes that differ in the pairing and produce identical logs. Verification consequently belongs at the level of the barcode, and it can be carried out against the definition of the invariant instead of against a second implementation, since over the two-element field the generalised rank and its inversion amount to ranks of Boolean matrices.

On a null model and under noise the same pipeline floods its output with spurious events, which identifies the next problem — a significance criterion for zigzag bars. Stability results [23] and faster algorithms [24], [25] delimit what such a criterion may assume.

Acknowledgment

The study was funded by Russian Science Foundation (grant 26-41-03005).

References

[1] M. G. Preti, T. A. Bolton, and D. Van De Ville, "The dynamic functional connectome: State-of-the-art and perspectives," *NeuroImage*, vol. 160, pp. 41–54, 2017.

[2] A. E. Sizemore, J. E. Phillips-Cremins, R. Ghrist, and D. S. Bassett, "The importance of the whole: Topological data analysis for the network neuroscientist," *Network Neuroscience*, vol. 3, no. 3, pp. 656–673, 2019.

[3] B. J. Stolz, H. A. Harrington, and M. A. Porter, "Persistent homology of time-dependent functional networks constructed from coupled time series," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 27, no. 4, p. 047410, 2017.

[4] A. V. Andreev, S. A. Kurkin, D. Stoyanov, A. A. Badarin, R. Paunova, and A. E. Hramov, "Toward interpretability of machine learning methods for the classification of patients with major depressive disorder based on functional network measures," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 33, no. 6, p. 063140, 2023.

[5] A. N. Pisarchik *et al.*, "Topology switching during window thresholding fMRI-based functional networks of patients with major depressive disorder: Consensus network approach," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 33, no. 9, p. 093122, 2023.

[6] M. S. Kabir, S. A. Kurkin, G. Portnova, O. Martynova, Z. Wang, and A. E. Hramov, "Contrastive machine learning reveals in EEG resting-state network salient features specific to autism spectrum disorder," *Chaos, Solitons & Fractals*, vol. 185, p. 115123, 2024.

[7] A. A. Badarin *et al.*, "Brain compensatory mechanisms during the prolonged cognitive task: fNIRS and eye-tracking study," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 17, no. 2, pp. 303–314, 2025.

[8] M. V. Khramova *et al.*, "Monitoring the cortical activity of children and adults during cognitive task completion," *Sensors*, vol. 21, no. 18, p. 6021, 2021.

[9] S. A. Kurkin, V. V. Grubov, V. A. Maksimenko, E. N. Pitsik, M. V. Khramova, and A. E. Hramov, "System for monitoring and adjusting the learning process of primary schoolchildren based on the EEG data analysis," *Informatsionno-Upravliaiushchie Sistemy [Information and Control Systems]*, no. 5, pp. 50–61, 2020.

[10] S. A. Kurkin, A. E. Hramov, P. Chholak, and A. N. Pisarchik, "Localizing oscillatory sources in a brain by MEG data during cognitive activity," in *Proc. 4th International Conference on Computational Intelligence and Networks (CINE)*, 2020, pp. 1–4.

[11] A. A. Badarin *et al.*, "Psychophysiological parameters predict the performance of naive subjects in sport shooting training," *Sensors*, vol. 23, no. 6, p. 3160, 2023.

[12] G. Carlsson and V. de Silva, "Zigzag persistence," *Foundations of Computational Mathematics*, vol. 10, no. 4, pp. 367–405, 2010.

[13] G. Carlsson, V. de Silva, and D. Morozov, "Zigzag persistent homology and real-valued functions," in *Proc. 25th Annual Symposium on Computational Geometry (SoCG)*, 2009, pp. 247–256.

[14] P. Gabriel, "Unzerlegbare Darstellungen I," *Manuscripta Mathematica*, vol. 6, no. 1, pp. 71–103, 1972.

[15] S. Y. Oudot, *Persistence Theory: From Quiver Representations to Data Analysis*, ser. Mathematical Surveys and Monographs. American Mathematical Society, 2015, vol. 209.

[16] O. E. Karpov *et al.*, "Extreme value theory inspires explainable machine learning approach for seizure detection," *Scientific Reports*, vol. 12, no. 1, p. 11474, 2022.

[17] V. V. Grubov *et al.*, "Two-stage approach with combination of outlier detection method and deep learning enhances automatic epileptic seizure detection," *IEEE Access*, vol. 12, pp. 122 168–122 182, 2024.

[18] W. Kim and F. Mémoli, "Generalized persistence diagrams for persistence modules over posets," *Journal of Applied and Computational Topology*, vol. 5, no. 4, pp. 533–581, 2021.

[19] D. Morozov, "Dionysus 2," https://mrzv.org/software/dionysus2/, 2018, accessed 2026-08-23.

[20] E. B. Wilson, "Probable inference, the law of succession, and statistical inference," *Journal of the American Statistical Association*, vol. 22, no. 158, pp. 209–212, 1927.

[21] B. T. Fasy, F. Lecci, A. Rinaldo, L. Wasserman, S. Balakrishnan, and A. Singh, "Confidence sets for persistence diagrams," *The Annals of Statistics*, vol. 42, no. 6, pp. 2301–2339, 2014.

[22] D. Cohen-Steiner, H. Edelsbrunner, and J. Harer, "Stability of persistence diagrams," *Discrete & Computational Geometry*, vol. 37, no. 1, pp. 103–120, 2007.

[23] M. B. Botnan and M. Lesnick, "Algebraic stability of zigzag persistence modules," *Algebraic & Geometric Topology*, vol. 18, no. 6, pp. 3133–3204, 2018.

[24] T. K. Dey and T. Hou, "Fast computation of zigzag persistence," in *30th Annual European Symposium on Algorithms (ESA)*, ser. LIPIcs, vol. 244, 2022, pp. 43:1–43:15.

[25] W. Kim and F. Mémoli, "Spatiotemporal persistent homology for dynamic metric spaces," *Discrete & Computational Geometry*, vol. 66, pp. 831–875, 2021.

Authorized licensed use limited to: Kant Baltic Federal Univ. Downloaded on October 23,2026 at 15:25:30 UTC from IEEE Xplore. Restrictions apply.