Topological Classification in a Wasserstein Distance Based Vector Space

Tananun Songdechakraiwut
Department of Electrical & Computer Engineering
University of Wisconsin–Madison
songdechakra@wisc.edu

Bryan M. Krause & Matthew I. Banks
Department of Anesthesiology
Department of Neuroscience
University of Wisconsin–Madison

Kirill V. Nourski
Department of Neurosurgery
University of Iowa

Barry D. Van Veen
Department of Electrical & Computer Engineering
University of Wisconsin–Madison

Abstract

Classification of large and dense networks based on topology is very difficult due to the computational challenges of extracting meaningful topological features from real-world networks. In this paper we present a computationally tractable approach to topological classification of networks by using principled theory from persistent homology and optimal transport to define a novel vector representation for topological features. The proposed vector space is based on the Wasserstein distance between persistence barcodes. The 1-skeleton of the network graph is employed to obtain 1D persistence barcodes that represent connected components and cycles. These barcodes and the corresponding Wasserstein distance can be computed very efficiently. The effectiveness of the proposed vector space is demonstrated using support vector machines to classify brain networks.

1 Introduction

Networks are ubiquitous representations for describing complex, highly interconnected systems that capture potentially intricate patterns of relationships between nodes. [4]. Finding meaningful characterizations of network structure is very difficult, especially for large and dense networks with node degrees ranging over multiple orders of magnitude [10, 25].

Persistent homology [3, 20, 45] is an emerging tool for understanding, characterizing and quantifying the topology of complex networks [15, 41]. Topology is characterized using connected components (0D topological features), cycles (1D topological features) and higher dimensional, difficult to visualize, objects. Connected components and cycles are the most dominant and fundamental topological features of real networks. Many networks naturally organize into modules or connected components [10, 25]. Similarly, cycle structure is ubiquitous and is often interpreted in terms of information propagation, redundancy and feedback loops [29, 36, 46].

Here we present a novel *topological vector space* (TopVS) that embeds persistence barcodes for connected components and cycles. TopVS preserves the underlying distance in the original space of persistence barcodes while existing methods do not [13]. The p-norm distance in TopVS is equivalent to the p-Wasserstein distance in the original barcode space. This equivalence allows the computation of summary statistics such as the mean of persistence barcodes to be easily performed in TopVS. The utility of TopVS is illustrated by classifying measured functional brain networks associated with different levels of arousal during administration of general anesthesia. TopVS performs very well compared to other topology-based approaches.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

2 Wasserstein Distance Based Vector Space

Birth-Death Decomposition Define a network as an undirected weighted graph $G = (V, \boldsymbol{w})$ with a set of nodes V, and a weighted adjacency matrix $\boldsymbol{w} = (w_{ij})$. Define a binary graph G_ϵ with the identical node set V by thresholding the edge weights so that an edge between nodes i and j exists if $w_{ij} > \epsilon$. The binary graph is viewed as a 1-skeleton [32]. As ϵ increases, more and more edges are removed from the network G. Thus, we have a graph filtration: $G_{\epsilon_0} \supseteq G_{\epsilon_1} \supseteq \cdots \supseteq G_{\epsilon_k}$, where $\epsilon_0 \le \epsilon_1 \le \cdots \le \epsilon_k$ are called filtration values. Persistent homology keeps track of the birth and death of topological features over filtration values ϵ. A topological feature that is born at a filtration b_i and persists up to a filtration d_i, is represented as a 2D point (b_i, d_i) in a plane. A set of all the points $\{(b_i, d_i)\}$ is called *persistence barcode* [23]. In the 1-skeleton, the only non-trivial topological features are connected components and cycles. As ϵ increases, the number of connected components $\beta_0(G_\epsilon)$ and cycles $\beta_1(G_\epsilon)$ are monotonically increasing and decreasing, respectively [41]. Thus, the representation of the connected components and cycles can be simplified to a collection of sorted birth values $B(G) = \{b_i\}_{i=1}^{|V|-1}$ and a collection of sorted death values $D(G) = \{d_i\}_{i=1}^{1+|V|(|V|-3)/2}$, respectively [41].

Closed-Form Wasserstein Distance Let G_i be a network. Its underlying probability density function on the barcodes for connected components is defined in the form of Dirac masses [44] as $f_{G_i,B}(x) := \frac{1}{|B(G_i)|}\sum_{b\in B(G_i)} \delta(x-b)$ where $\delta(x-b)$ is a Dirac delta centered at the point b. Then the empirical distribution is the integration of $f_{G_i,B}$ as $F_{G_i,B}(x) = \frac{1}{|B(G_i)|}\sum_{b\in B(G_i)} \mathbb{1}_{b\le x}$ where $\mathbb{1}_{b\le x}$ is an indicator function taking the value 1 if $b \le x$, and 0 otherwise. A pseudoinverse of $F_{G_i,B}$ is defined as $F^{-1}_{G_i,B}(z) = \inf\{b \in \mathbb{R} \mid F_{G_i,B}(b) \ge z\}$. Then the empirical Wasserstein distance for connected components has a closed-form solution in terms of pseudoinverses as $W_{p,B}(G_1, G_2) = \Big(\int_0^1 |F^{-1}_{G_1,B}(z) - F^{-1}_{G_2,B}(z)|^p \, dz\Big)^{1/p}$. Similarly, the Wasserstein distance for cycles $W_{p,D}(G_1, G_2)$ is defined in terms of empirical distributions for death sets $D(G_1)$ and $D(G_2)$.

The empirical Wasserstein distances $W_{p,B}$ and $W_{p,D}$ are approximated by computing the Lebesgue integration numerically as follows. Let $\widehat{B}(G_1) = \{F^{-1}_{G_1,B}(1/m), ..., F^{-1}_{G_1,B}(m/m)\}$ and $\widehat{D}(G_1) = \{F^{-1}_{G_1,D}(1/n), ..., F^{-1}_{G_1,D}(n/n)\}$ be pseudoinverses of network G_1 sampled with partitions of equal intervals. Let $\widehat{B}(G_2)$ and $\widehat{D}(G_2)$ be sampled pseudoinverses of network G_2 with the same partitions of m and n, respectively. Then the approximated Wasserstein distances are given by $\widehat{W}_{p,B}(G_1, G_2) = \Big(\frac{1}{m^p}\sum_{k=1}^m \big|F^{-1}_{G_1,B}(k/m) - F^{-1}_{G_2,B}(k/m)\big|^p\Big)^{1/p}$ and $\widehat{W}_{p,D}(G_1, G_2) = \Big(\frac{1}{n^p}\sum_{k=1}^n \big|F^{-1}_{G_1,D}(k/n) - F^{-1}_{G_2,D}(k/n)\big|^p\Big)^{1/p}$.

Topological Vector Representation A collection of 1D persistence barcodes together with the Wasserstein distance is a metric space. 1D persistence barcodes can be embedded into a vector space that preserves the Wasserstein metric on the original space of persistence barcodes as follows. Let $G_1, G_2, ..., G_N$ be N observed networks possibly with different node sizes. Let $F^{-1}_{G_i,B}$ be a pseudoinverse of network G_i. The vector representation of a persistence barcode for connected components in network G_i is defined as a vector of the pseudoinverse sampled at $1/m, 2/m, ..., m/m$. That is, $\boldsymbol{v}_{B,i} := \big(F^{-1}_{G_i,B}(1/m), F^{-1}_{G_i,B}(2/m), ..., F^{-1}_{G_i,B}(m/m)\big)^\top$. A collection of these vectors $M_B = \{\boldsymbol{v}_{B,i}\}_{i=1}^N$ with the p-norm $||\cdot||_p$ induces the p-norm metric $d_{p,B}$ given by $d_{p,B}(\boldsymbol{v}_{B,i}, \boldsymbol{v}_{B,j}) = ||\boldsymbol{v}_{B,i} - \boldsymbol{v}_{B,j}||_p = m\widehat{W}_{p,B}$. Thus, for $p = 1$ the proposed vector space describes Manhattan distance, $p = 2$ Euclidean distance, and $p \to \infty$ the maximum metric, which in turn correspond to the earth mover's distance (W_1) [38], 2-Wasserstein distance (W_2), and the bottleneck distance (W_∞) [27], respectively, in the original space of persistence barcodes. Similarly, we can define a vector space of persistence barcodes for cycles $M_D = \{\boldsymbol{v}_{D,i}\}_{i=1}^N$ with the p-norm metric $d_{p,D}$. The normed vector space $(M_B, d_{p,B})$ describes topological space of connected components in networks, while $(M_D, d_{p,D})$ describes topological space of cycles in networks.

The topology of a network viewed as a 1-skeleton is *completely* characterized by connected components and cycles. Thus, we can fully describe the network topology using both M_B and M_D as follows. Let $M_B \times M_D = \{(\boldsymbol{v}_{B,i}, \boldsymbol{v}_{D,i}) \mid \boldsymbol{v}_{B,i} \in M_B, \boldsymbol{v}_{D,i} \in M_D\}$ be the Cartesian product between M_B

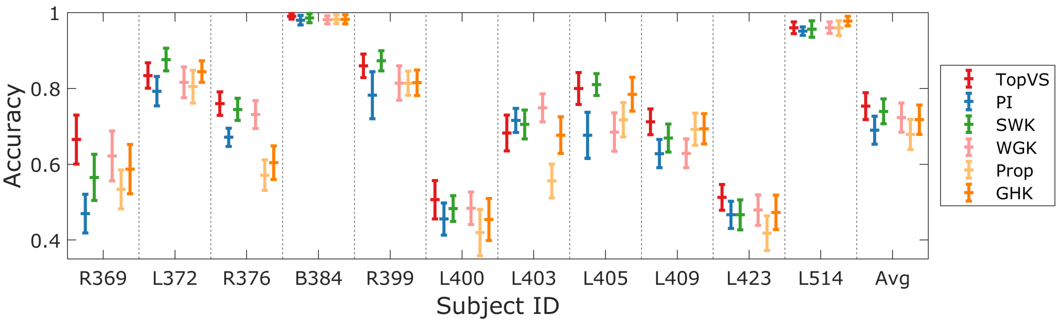

Figure 1: Accuracy classifying brain networks within individual subjects. The last column displays the average accuracy obtained across all subjects. The center markers and bars depict the means and standard deviations obtained over 100 trials.

and M_D so the vectors in $M_B \times M_D$ are the concatenations of $\boldsymbol{v}_{B,i}$ and $\boldsymbol{v}_{D,i}$. For this product space to represent meaningful topology of network G_i, the vectors $\boldsymbol{v}_{B,i}$ and $\boldsymbol{v}_{D,i}$ must be a network decomposition, as stated in Section 2. Thus $\boldsymbol{v}_{B,i}$ and $\boldsymbol{v}_{D,i}$ are constructed by sampling their psudoinverses with $m = \mathcal{V} - 1$ and $n = 1 + \frac{\mathcal{V}(\mathcal{V}-3)}{2}$, respectively, where $\mathcal{V}$ is a free parameter indicating a reference network size. The metrics $d_{p,B}$ and $d_{p,D}$ can be put together to form a p-product metric $d_{p,\times}$ on $M_B \times M_D$ as $d_{p,\times}\big((\boldsymbol{v}_{B,i}, \boldsymbol{v}_{D,i}), (\boldsymbol{v}_{B,j}, \boldsymbol{v}_{D,j})\big) = \big([d_{p,B}(\boldsymbol{v}_{B,i}, \boldsymbol{v}_{B,j})]^p + [d_{p,D}(\boldsymbol{v}_{D,i}, \boldsymbol{v}_{D,j})]^p\big)^{1/p} = \big([m\widehat{W}_{p,B}]^p + [n\widehat{W}_{p,D}]^p\big)^{1/p}$, where $(\boldsymbol{v}_{B,i}, \boldsymbol{v}_{D,i}), (\boldsymbol{v}_{B,j}, \boldsymbol{v}_{D,j}) \in M_B \times M_D$, $m = \mathcal{V} - 1$ and $n = 1 + \frac{\mathcal{V}(\mathcal{V}-3)}{2}$. Thus, $d_{p,\times}$ is a weighted combination of p-Wasserstein distances, and is simply the p-norm metric between vectors constructed by concatenating $\boldsymbol{v}_{B,i}$ and $\boldsymbol{v}_{D,i}$. The normed vector space $(M_B \times M_D, d_{p,\times})$ is termed *topological vector space* (TopVS). A direct consequence of the equality is that the mean of persistence barcodes under the approximated Wasserstein distance is equivalent to the sample mean vector in TopVS. In addition, the proposed vector representation is highly interpretable because persistence barcodes can be easily reconstructed from vectors by separating sorted births and deaths.

Related work, statistical validation and runtime experiment for TopVS are provided in Appendices A to C.

3 Application to Functional Brain Networks

Dataset We evaluate our method using a brain network dataset from the anesthesia study [2] (see Appendix E for details). The brain networks are based on eleven neurosurgical patients during administration of increasing doses of the general anesthetic propofol. Each segment is labeled as one of the three arousal states: pre-drug *wake*, *sedated* but responsive to command, or *unresponsive*.

Classification performance evaluation We compare the classification performance of the proposed TopVS relative to that of several state-of-the-art methods for persistence barcodes and graph kernels including persistence image (PI) [1], sliced Wasserstein kernel (SWK) [14], persistence weighted gaussian kernel (PWGK) [28], propagation kernel (Prop) [33] and graph hopper kernel (GHK) [22]. Implementation details of all the baseline methods are provided in Appendix D.

We are interested in whether 1) the candidate methods can differentiate arousal states within individual subjects, and 2) generalize their learned knowledge to unknown subjects afterwards. As a result, we consider two different nested CV tasks as follows. For the first task, we apply a nested CV comprising an outer loop of stratified 2-fold CV and an inner loop of stratified 3-fold CV, for each subject. Since we may get a different split of data folds each time, we perform the nested CV for 100 trials and report an average accuracy score and standard deviation for each subject. We also average these individual accuracy scores across subjects (11 × 100 scores) to obtain an overall accuracy.

For the second task, we use a different nested CV comprising both outer and inner loops with a leave-one-subject-out scheme. That is, a classifier is trained using all but one test subject. The inner loop is used to determine optimal hyperparameters, while the outer loop is used to assess generalization capacity of the candidate methods to unknown subjects in the population.

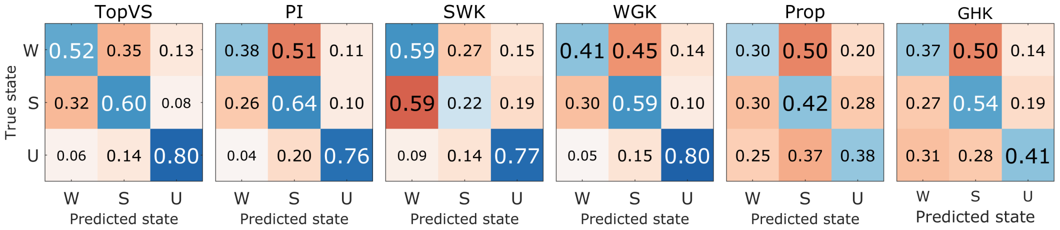

Figure 2: Confusion matrices illustrating method performance for classifying across subjects. The numbers represent the fraction of brain networks in the test subjects being predicted as one of the three possible states: wake (W), sedated (S), and unresponsive (U).

Results Figure 1 compares classification accuracy for individual subjects. There is variability in performance across subjects and across methods. In most subjects all methods perform relatively well. Our TopVS method is consistently among the best performing classifiers, resulting in the higher overall performance. On the other hand, the PI and Prop methods perform poorest in most subjects. The consistently poorer performance of PI and Prop is evident in the lower overall performance. For classifying across subjects, results are 0.65 ± 0.21, 0.58 ± 0.22, 0.57 ± 0.20, 0.60 ± 0.21, 0.36 ± 0.12 and 0.43 ± 0.14 for TopVS, PI, SWK, WGK, Prop and GHK, respectively. TopVS is still among the best methods for classifying across subjects, while the performance of both graph kernels suffers.

Figure 2 displays a summary of the across-subject prediction results using confusion matrices. Except for two graph kernels, the other methods are generally effective for separating unresponsive (U) from the other two states. However, the majority of classification errors are associated with the differentiation between wake (W) and sedated (S) states. This misclassification is consistent with prior biological expectations since the sedated brain, in which subjects have been administered propofol but are still conscious, is expected to have a great deal of similarity with the wake brain [2]. TopVS appears to show clear advantages over other baseline methods for differentiating wake and sedated states. This suggests that the proposed vector representation is an effective choice for representing subtle topological structure in networks.

References

[1] Henry Adams, Tegan Emerson, Michael Kirby, Rachel Neville, Chris Peterson, Patrick Shipman, Sofya Chepushtanova, Eric Hanson, Francis Motta, and Lori Ziegelmeier. Persistence images: A stable vector representation of persistent homology. *Journal of Machine Learning Research*, 18, 2017.

[2] Matthew I Banks, Bryan M Krause, Christopher M Endemann, Declan I Campbell, Christopher K Kovach, Mark Eric Dyken, Hiroto Kawasaki, and Kirill V Nourski. Cortical functional connectivity indexes arousal state during sleep and anesthesia. *NeuroImage*, 211:116627, 2020.

[3] Serguei Barannikov. The framed Morse complex and its invariants. *Advances in Soviet Mathematics, American Mathematical Society*, 1994.

[4] Alain Barrat, Marc Barthelemy, Romualdo Pastor-Satorras, and Alessandro Vespignani. The architecture of complex weighted networks. *Proceedings of the National Academy of Sciences*, 101(11):3747–3752, 2004.

[5] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(2), 2012.

[6] Matteo Biagetti, Alex Cole, and Gary Shiu. The persistence of large scale structures. Part I. Primordial non-Gaussianity. *Journal of Cosmology and Astroparticle Physics*, 2021(04):061, 2021.

[7] Karsten Borgwardt, Elisabetta Ghisu, Felipe Llinares-López, Leslie O'Bray, and Bastian Alexander Rieck. Graph kernels: State-of-the-art and future challenges. *Foundations and Trends in Machine Learning*, 13(5-6):531–712, 2020.

[8] Peter Bubenik and Paweł Dłotko. A persistence landscapes toolbox for topological statistics. *Journal of Symbolic Computation*, 78:91–114, 2017.

[9] Peter Bubenik and Peter T Kim. A statistical approach to persistent homology. *Homology, Homotopy and Applications*, 9(2):337–362, 2007.

[10] Ed Bullmore and Olaf Sporns. Complex brain networks: Graph theoretical analysis of structural and functional systems. *Nature Reviews Neuroscience*, 10(3):186–198, 2009.

[11] Zixuan Cang, Lin Mu, Kedi Wu, Kristopher Opron, Kelin Xia, and Guo-Wei Wei. A topological approach for protein classification. *Computational and Mathematical Biophysics*, 3(1), 2015.

[12] Zixuan Cang, Lin Mu, and Guo-Wei Wei. Representability of algebraic topology for biomolecules in machine learning based scoring and virtual screening. *PLoS Computational Biology*, 14(1):e1005929, 2018.

[13] Mathieu Carrière and Ulrich Bauer. On the metric distortion of embedding persistence diagrams into separable Hilbert spaces. In *Symposium on Computational Geometry*, 2019.

[14] Mathieu Carriere, Marco Cuturi, and Steve Oudot. Sliced Wasserstein kernel for persistence diagrams. In *International Conference on Machine Learning*, pp. 664–673. PMLR, 2017.

[15] Mathieu Carrière, Frédéric Chazal, Yuichi Ike, Théo Lacombe, Martin Royer, and Yuhei Umeda. Perslay: A neural network layer for persistence diagrams and new graph topological signatures. In *International Conference on Artificial Intelligence and Statistics*, pp. 2786–2796. PMLR, 2020.

[16] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):1–27, 2011.

[17] Frédéric Chazal, Marc Glisse, Catherine Labruère, and Bertrand Michel. Convergence rates for persistence diagram estimation in topological data analysis. In *International Conference on Machine Learning*, 2014.

[18] Ilya Chevyrev, Vidit Nanda, and Harald Oberhauser. Persistence paths and signature features in topological data analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42 (1):192–202, 2020.

[19] David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *Discrete & Computational Geometry*, 37(1):103–120, 2007.

[20] Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. Topological persistence and simplification. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pp. 454–463. IEEE, 2000.

[21] Brittany Terese Fasy, Fabrizio Lecci, Alessandro Rinaldo, Larry Wasserman, Sivaraman Balakrishnan, and Aarti Singh. Confidence sets for persistence diagrams. *The Annals of Statistics*, 42 (6):2301–2339, 2014.

[22] Aasa Feragen, Niklas Kasenburg, Jens Petersen, Marleen de Bruijne, and Karsten M Borgwardt. Scalable kernels for graphs with continuous attributes. In *Advances in Neural Information Processing Systems*, pp. 216–224, 2013.

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

[24] Thomas Hofmann, Bernhard Schölkopf, and Alexander J Smola. Kernel methods in machine learning. *The Annals of Statistics*, 36(3):1171–1220, 2008.

[25] Christopher J Honey, Rolf Kötter, Michael Breakspear, and Olaf Sporns. Network structure of cerebral cortex shapes functional connectivity on multiple time scales. *Proceedings of the National Academy of Sciences*, 104(24):10240–10245, 2007.

[26] Sara Kališnik. Tropical coordinates on the space of persistence barcodes. *Foundations of Computational Mathematics*, 19:101–129, 2019.

[27] Michael Kerber, Dmitriy Morozov, and Arnur Nigmetov. Geometry helps to compare persistence diagrams. *ACM J. Exp. Algorithmics*, 22, 2017.

[28] Genki Kusano, Yasuaki Hiraoka, and Kenji Fukumizu. Persistence weighted Gaussian kernel for topological data analysis. In *International Conference on Machine Learning*, pp. 2004–2013. PMLR, 2016.

[29] Yung-Keun Kwon and Kwang-Hyun Cho. Analysis of feedback loops and robustness in network evolution based on boolean models. *BMC Bioinformatics*, 8(1):1–9, 2007.

[30] Théo Lacombe, Marco Cuturi, and Steve Oudot. Large scale computation of means and clusters for persistence diagrams using optimal transport. *Advances in Neural Information Processing Systems*, 2018.

[31] Hyekyoung Lee, Hyejin Kang, Moo K Chung, Bung-Nyun Kim, and Dong Soo Lee. Persistent brain network homology from the perspective of dendrogram. *IEEE Transactions on Medical Imaging*, 31(12):2267–2277, 2012.

[32] James R Munkres. *Elements of Algebraic Topology*. CRC press, 2018.

[33] Marion Neumann, Roman Garnett, Christian Bauckhage, and Kristian Kersting. Propagation kernels: Efficient graph kernels from propagated information. *Machine Learning*, 102(2): 209–245, 2016.

[34] Markus Ojala and Gemma C Garriga. Permutation tests for studying classifier performance. *Journal of Machine Learning Research*, 11(6), 2010.

[35] Nina Otter, Mason A Porter, Ulrike Tillmann, Peter Grindrod, and Heather A Harrington. A roadmap for the computation of persistent homology. *EPJ Data Science*, 6:1–38, 2017.

[36] Ertugrul M Ozbudak, Attila Becskei, and Alexander Van Oudenaarden. A system of counteracting feedback loops regulates Cdc42p activity during spontaneous cell polarization. *Developmental Cell*, 9(4):565–571, 2005.

[37] Julien Rabin, Gabriel Peyré, Julie Delon, and Marc Bernot. Wasserstein barycenter and its application to texture mixing. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pp. 435–446. Springer, 2011.

[38] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

[39] Nathaniel Saul and Chris Tralie. Scikit-TDA: Topological data analysis for Python, 2019.

[40] Giannis Siglidis, Giannis Nikolentzos, Stratis Limnios, Christos Giatsidis, Konstantinos Skianis, and Michalis Vazirgiannis. GraKeL: A graph kernel library in Python. *Journal of Machine Learning Research*, 21(54):1–5, 2020.

[41] Tananun Songdechakraiwut, Li Shen, and Moo Chung. Topological learning and its application to multimodal brain network integration. *24th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2021.

[42] The GUDHI Project. *GUDHI User and Reference Manual*. GUDHI Editorial Board, 3.5.0 edition, 2022. URL `https://gudhi.inria.fr/doc/3.5.0/`.

[43] Christopher Tralie, Nathaniel Saul, and Rann Bar-On. Ripser.py: A lean persistent homology library for Python. *The Journal of Open Source Software*, 3(29):925, Sep 2018.

[44] Katharine Turner, Yuriy Mileyko, Sayan Mukherjee, and John Harer. Fréchet means for distributions of persistence diagrams. *Discrete & Computational Geometry*, 52(1):44–70, 2014.

[45] Larry Wasserman. Topological data analysis. *Annual Review of Statistics and Its Application*, 5: 501–532, 2018.

[46] Orion D Weiner, Paul O Neilsen, Glenn D Prestwich, Marc W Kirschner, Lewis C Cantley, and Henry R Bourne. A PtdInsP 3-and Rho GTPase-mediated positive feedback loop regulates neutrophil polarity. *Nature Cell Biology*, 4(7):509–513, 2002.

[47] Kelin Xia, Zhiming Li, and Lin Mu. Multiscale persistent functions for biomolecular structure characterization. *Bulletin of Mathematical Biology*, 80:1–31, 2018.

[48] Xin Xu, Jessi Cisewski-Kehe, Sheridan Beckwith Green, and Daisuke Nagai. Finding cosmic voids and filament loops using topological data analysis. *Astronomy and Computing*, 27:34–52, 2019.

[49] Bartosz Zieliński, Michał Lipiński, Mateusz Juda, Matthias Zeppelzauer, and Paweł Dłotko. Persistence codebooks for topological data analysis. *Artificial Intelligence Review*, 54:1969–2009, 2021.

A Related Work

The extraction of topological features from persistence barcodes is an active area of research. A large number of methods have been proposed in the literature, ranging from vectorization methods that utilize limited topological information extracted from barcodes, such as statistics [31], Betti numbers [11] and tropical coordinates [26], to more systematic featurizations, such as binning approach [12], persistent codebooks [49], persistent functions [1, 8, 47] and persistent paths [18]. Embeddings extracted from these methods, existing in both 1D and 2D representations, bridge the gap between barcode representations and machine learning, enabling the incorporation of topological information into various learning models.

The central trait of persistence barcodes is their stability properties [19]. It is thus of great interest to understand if such properties carry over to barcode embeddings. Previously proposed feature maps [1, 8, 47] achieve limited stability guarantees [13]. In contrast, TopVS is derived directly from the closed form expression of the Wasserstein distance and thus is faithful to the stability properties of persistence barcodes. Note also that performing statistical analysis in a metric space of such embeddings is also of interest [9, 17, 21, 44]. The capability to model and quantify variations in data, such as mean and variance, is important. TopVS allows closed form computation of such statistics and enables the corresponding analyses. The high computational cost of persistent homology methods is a hindrance to their use on large datasets. Previously proposed algorithms [27, 30] rely on approximate solutions, typically iterative, to manage computational complexity. In contrast, TopVS is computed efficiently using closed form expressions.

TopVS is limited to the representation of connected components and cycles. These two features play a dominant role in topological analyses. For example, they are widely utilized in the brain network community [10, 25]. While higher-order topological features have been considered in cosmology [48], recent work shows that cosmic voids are not as discriminative as filament loops, i.e., cycles, due to their relatively rare occurrence [6]. Hence, while higher-order topological information is of theoretical interest, it is rarely used in applications of topology.

B Validation using Simulated Networks

Simulated networks of different topological structure are used to compare the classification performance of the proposed TopVS relative to that of several state-of-the-art kernel methods for persistence barcodes and graph kernels. While nearly any classifier may be used with TopVS, here we illustrate results using the C-support vector machine (SVM) [16] with the linear kernel, which maximizes Wasserstein distance-based margin. When the TopVS method is applied to different-size networks, we compute birth and death sets of the largest network, and upsample birth and death sets of smaller networks to match that of the largest network in size. If the networks considered have the same size, we simply vectorize their birth and death sets.

The performance of TopVS is compared to five other methods published in the literature. Three of these methods are based on 2-dimensional persistence barcodes: the *Persistence Image* (PI) vectorization [1], the *Sliced Wasserstein* kernel (SWK) [14] and the *Persistence Weighted Gaussian* kernel (PWGK) [28]. The other two benchmark methods are based on graph kernels: the *Propagation* kernel (Prop) [33] and the *GraphHopper* kernel (GHK) [22]. The PI method embeds persistence barcodes into a vector space in which classification is performed using linear SVMs. The non-linear SWK, PWGK, Prop and GHK methods are combined with SVMs using the kernel trick [24] to perform classification.

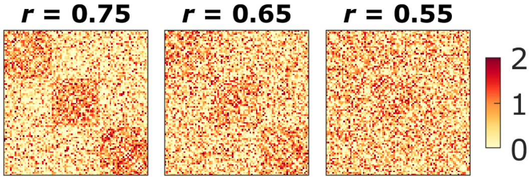

Figure 3: Example networks with $|V| = 90$ nodes distributed evenly among $m = 3$ modules according to within-module-connection probabilities $r = 0.75, 0.65$ and 0.55.

The three persistence barcode methods require computation of 2-dimensional persistence barcodes from networks. We compute a 2-dimensional persistence barcode using the approach of Otter et al. [35] in which edge weights are inverted via the function $f(w) = 1/(1 + w)$. Then a point cloud is obtained from the shortest path distance between nodes. Finally, the Ripser implementation [43] of the Rips filtration [23] generates the persistence barcode from the point cloud. The two graph kernel methods require node continuous attributes. We follow the experimental protocol of Borgwardt et al. [7] in which a node attribute is set to the sum of edge weights incident to the node. Implementation details of the baseline methods are provided in Appendix D.

Evaluation and tuning protocol Nested cross validation (CV) is used for selection of optimal hyperparameters and assessment of generalization capacity of the candidate algorithms for classifying networks. Nested CV comprises an outer loop of stratified 2-fold CV and an inner loop of stratified 5-fold CV. The folds in stratified CV are selected by preserving the percentage of network samples for each group label. The inner loop is used to tune hyperparameters via grid search [5] to determine the set of optimal hyperparameters that achieves the highest accuracy. The outer loop provides a generalized performance evaluation for the model trained using the optimal hyperparameters from the inner loop. Thus, the nested CV procedure finds the average of accuracy scores over 2 folds in the outer loop using a model trained by the optimal hyperparameters obtained from the inner loop. Additional details on hyperparameter values and tuning are provided in Appendix D.

Simulated modular network structure Random modular networks $\mathcal{X}_i$ are simulated with $|V|$ nodes and m modules such that the nodes are evenly distributed among modules. Figure 3 displays modular networks with $|V| = 90$ nodes and $m = 3$ modules such that $|V|/m = 30$ nodes are in each module. Edges connecting two nodes within the same module are assigned a random weight following a normal distribution $\mathcal{N}(1, 0.5^2)$ with probability r or otherwise Gaussian noise $\mathcal{N}(0, 0.5^2)$ with probability $1 - r$. On the other hand, edges connecting nodes in different modules have probability $1-r$ of being $\mathcal{N}(1, 0.5^2)$ and probability r of being $\mathcal{N}(0, 0.5^2)$. The modular structure becomes more pronounced as the within-module connection probability r increases. Any negative edge weights are set to zero. This procedure yields random networks $\mathcal{X}_i$ that exhibit topological connectedness.

Simulated dataset Two groups of modular networks $L_1 = \{\mathcal{X}_i\}_{i=1}^{30}$ and $L_2 = \{\mathcal{X}_i\}_{i=31}^{60}$ corresponding to $m = 3$ and 5 modules, respectively, are generated. This results in 60 networks in the dataset, each of which has a group label L_1 or L_2. Two different settings of network sizes are considered: 1) all 60 networks with $|V| = 90$ and 2) an equal number (ten) of networks with $|V| = 60, 90$ and 120 in each group. Three different settings of within-module connection probabilities are considered for each case: $r = 0.75, 0.65$ and 0.55 to vary the strength of the modular structure, as illustrated in Figure 3. Note that it is computationally challenging to apply statistical characterization such as a permutation test with persistence barcode and graph kernel based methods for even small networks of $|V| = 60$ and 90.

Classification performance evaluation Binary classification is performed on the generated dataset using the candidate algorithms. Nested CV is used to evaluate classification performance, resulting in an observed accuracy statistic s. Since the distribution of the accuracy s is unknown, a permutation test is used to determine the empirical distribution under the null hypothesis that sample networks and their group labels are independent [34]. The empirical distribution is calculated by repeatedly shuffling the group labels, thereby removing any dependency between the sample networks and the labels, and then re-computing the corresponding nested CV accuracy score for one thousand random permutations. By comparing the observed accuracy to this empirical distribution, we can determine

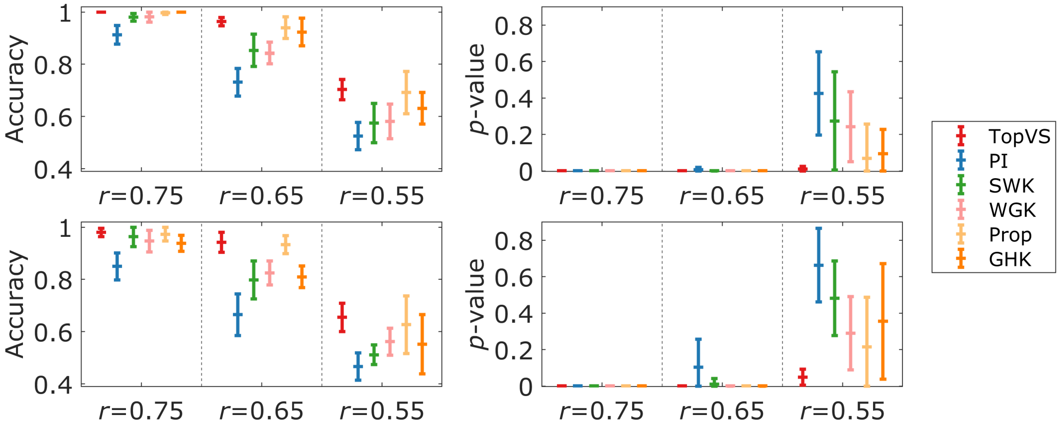

Figure 4: Classification performance comparison for simulated networks with $m = 3, 5$ modules and either $|V| = 90$ nodes (Top) or $|V| = 60, 90, 120$ nodes (bottom) with respect to average accuracy (left) and average p-values (right). Results for within-module connection probabilities $r = 0.55, 0.65$ and 0.75 are shown. Data points (middle horizontal lines) indicate the average results over ten independently generated datasets, while vertical error bars indicate standard deviations.

the statistical significance of the observed accuracy. The p-value is calculated as the fraction of permutations that give nested CV accuracy values higher than the observed accuracy s. The average p-value and average observed accuracy across ten independently generated datasets are reported.

Results Figure 4 indicates that all methods achieve relatively high accuracy on networks with pronounced modular structure ($r = 0.75$), and their accuracy decreases as the modularity strength diminishes, i.e., decreasing r. Our TopVS performs relatively well discriminating the more subtle modularity corresponding to $r = 0.65$ and 0.55. Since the dataset is purposefully generated to exhibit dependency between sample networks and their group labels, a low p-value provides statistical evidence that a trained classifier is able to leverage the dependency to differentiate network topology [34]. The proposed method has average p-values lower than 0.05 for all experimental settings, indicating that its improved accuracy over the baseline methods is significant. The Prop method has the closest accuracy to TopVS when $r = 0.55$, but has a higher p-value, indicating the accuracy is a less reliable indicator of performance.

C Runtime Experiment

All candidate methods used in the simulation study are evaluated for a runtime experiment. All methods were run on Intel Core i7 CPU with 16GB of RAM. Figure 5 displays the runtime vs input size plot. The result clearly shows that all three persistent homology based kernels (PI, SWK and WGK) are limited to dense networks with a few hundred nodes, representing the current scaling limit of persistent homology embedding methods. On the other hand, TopVS is able to compute a kernel between 2000-node networks each with approx. two million edges in about one second. The computational practicality of TopVS extends its applicability to the large-scale analyses of brain networks that cannot be analyzed using prior methods based on conventional persistence barcodes. Note that the time complexity of Prop is $O(|E|)$ while TopVS has the slightly higher complexity as $O(|E| \log |V|)$, where $|V|$ and $|E|$ are the number of nodes and edges. While Prop is the most efficient among all the methods, it has the lowest average accuracy when classifying the brain network data.

D Implementation Details of Candidate Methods

Implementation of *Persistence Image* (PI) vectorization [1] is performed using Persim [39]. *Sliced Wasserstein* kernel (SWK) [14] and *Persistence weighted Gaussian* kernel (PWGK) [28] are implemented using the Gudhi library [42]. *Propagation* kernel (Prop) [33] and *GraphHopper* kernel (GHK) [22] are implemented via GraKel library [40].

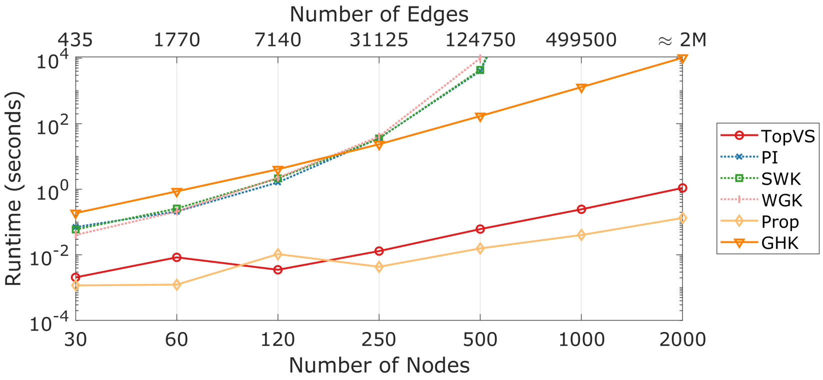

Figure 5: Runtime experiment. We measured the runtime as the average amount of time each algorithm takes to compute its kernel between two complete graphs starting from edge weights as a given input. The complete graphs were constructed using our modular network generation, described in Appendix B. The runtime is plotted with respect to network size in terms of both the number of nodes and edges.

For the PI method, we follow a parameter setting used by Adams et al. [1] to convert two persistence barcodes for connected components and cycles into two 2-dimensional pixel images of 20×20 resolution using a Gaussian function with variance 0.01. The two images are vectorized and concatenated into a single feature vector per network. Then linear SVMs are used to classify these vectors.

SWK is based on the sliced Wasserstein approximation [37] over 10 directions. PWGK is based on the RBF kernel and an $\arctan$ weight function recommended by Kusano et al. [28]. Both SWK and PWGK use combined persistence barcodes, each comprising 2-dimensional points of both connected components and cycles, to compute the Gram matrices.

Grid search [5] across different hyperparameter values is used to train all the candidate methods. SVMs have a regularization parameter $\mathcal{C} = \{0.01, 1, 100\}$. Thus, a grid search trains TopVS and PI methods with each $C \in \mathcal{C}$. The SWK and WGK methods have a bandwidth parameter $\Sigma = \{0.1, 1, 10\}$, and thus grid search trains both methods with each pair $(C, \sigma) \in \mathcal{C} \times \Sigma$. The Prop method has a maximum number of propagation iterations $T_{max} = \{1, 5, 10\}$, and thus is trained with each pair $(C, t_{max}) \in \mathcal{C} \times T_{max}$. GHK method uses the RBF kernel with a parameter $\Gamma = \{0.1, 1, 10\}$ between node attributes, and thus is trained with each pair $(C, \gamma) \in \mathcal{C} \times \Gamma$.

E Brain Network Dataset

Brain network data were obtained from eleven neurosurgical patients between 19 and 59 years old as described in Table 1. The patients were undergoing chronic invasive intracranial electroencephalography (iEEG) monitoring as part of their treatment for medically refractory epilepsy. The Code of Ethics of the World Medical Association (Declaration of Helsinki) for experiments involving humans was followed for all the experiments. The University of Iowa Institutional Review Board and the National Institutes of Health approved all research protocols, and written informed consent was obtained from all subjects. Acquisition of clinically required data was not impeded by the research and subjects were free to rescind their consent whenever they wished without interfering with their clinical evaluation. Subdural and depth electrodes (Ad-Tech Medical, Oak Creek, WI) used to obtain all research data were located by the team of epileptologists and neurosurgeons based solely on needs for clinical evaluation of the patients. Data collected in the operating room prior to electrode removal, before and during induction of general anesthesia with propofol were used to create the brain network dataset. Full description of the method for obtaining the brain network dataset and experimental procedure is provided in [2].

Table 1: Brain network dataset.

SUBJECT	AGE	GENDER	NETWORK SIZE
R369	30	M	199
L372	34	M	174
R376	48	F	189
B384	38	M	89
R399	22	F	175
L400	59	F	126
L403	56	F	194
L405	19	M	127
L409	31	F	160
L423	51	M	152
L514	46	M	118