Extended Abstract Track

# Topological Classification in a Wasserstein Distance Based Vector Space

**Editors:** List of editors' names


## 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 1-dimensional 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

Connected components and cycles are the most dominant and fundamental topological features of real networks. Many networks naturally organize into modules or connected components (Bullmore and Sporns, 2009; Honey et al., 2007). Similarly, cycle structure is ubiquitous and is often interpreted in terms of information propagation, redundancy and feedback loops (Kwon and Cho, 2007; Ozbudak et al., 2005; Weiner et al., 2002).

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 (Carrière and Bauer, 2019). 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.

## 2. Topological Space with Wasserstein Distance

**Graph Filtration** Define a network as an undirected weighted graph $G = (V, \boldsymbol{w})$ with a set of nodes $V$, and a weighted adjacency matrix $\boldsymbol{w} = (w_{ij})$. The number of nodes is denoted by $|V|$. Define a binary graph $G_\epsilon$ 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 simplicial complex consisting of only nodes and edges, that is, a 1-skeleton (Munkres, 2018). As $\epsilon$ increases, more and more edges are removed from the network $G$. Thus, we have a nested sequence of 1-skeletons: $G_{\epsilon_0} \supseteq G_{\epsilon_1} \supseteq \cdots \supseteq G_{\epsilon_k}$, where $\epsilon_0 \leq \epsilon_1 \leq \cdots \leq \epsilon_k$ are called filtration values. This sequence of 1-skeletons is called a *graph filtration*.

© 2022 .

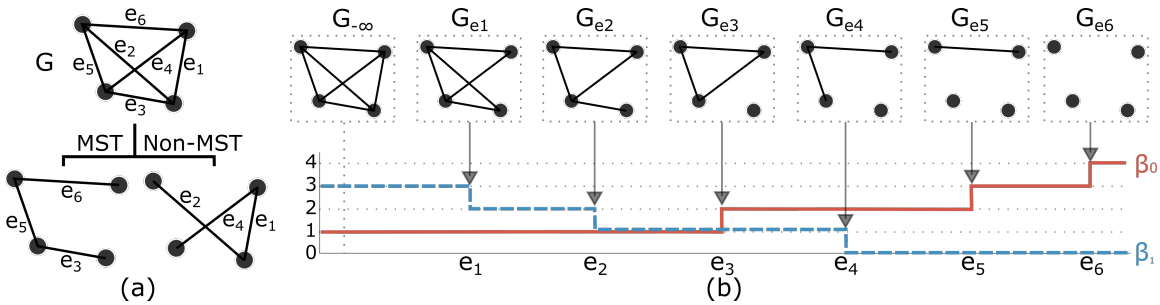


Figure 1: (a) Four-node network $G$ decomposes into its maximum spanning tree (MST) and a subnetwork with non-MST edge weights. (b) As the filtration value increases, the number of connected components $\beta_0$ monotonically increases while the number of cycles $\beta_1$ monotonically decreases.

**One Dimensional Persistence Barcodes** Persistent homology keeps track of the birth and death of topological features over filtration values $\epsilon$. A topological feature that is born at a filtration $b_i$ and persists up to a filtration $d_i$, is represented as a 2-dimensional point $(b_i, d_i)$ in a plane. A set of all the points $\{(b_i, d_i)\}$ is called *persistence barcode* (Ghrist, 2008). In the 1-skeleton, the only non-trivial topological features are connected components (0-dimensional topological features) and cycles (1-dimensional topological features). As $\epsilon$ increases, the number of connected components $\beta_0(G_\epsilon)$ and cycles $\beta_1(G_\epsilon)$ are monotonically increasing and decreasing, respectively (Songdechakraiwut et al., 2021). Thus, the representation of the connected components can be simplified to a collection of sorted birth values $B(G) = \{b_i\}_{i=1}^{|V|-1}$. Similarly, we can simplify the representation of the cycles as a collection of sorted death values $D(G) = \{d_i\}$. The example network of Figure 1 has $B(G) = \{e_3, e_5, e_6\}$ and $D(G) = \{e_1, e_2, e_4\}$. Supplementary description of the birth-death simplification is provided in Appendix A.3.2.

**Wasserstein Distance Simplification** The Wasserstein distance between the 1-dimensional barcodes of the graph filtration can be obtained using a closed-form solution. Let $G_i$ be a network. Its underlying probability density function on the persistence barcodes for connected components is defined in the form of Dirac masses (Turner et al., 2014) 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 $F_{G_i,B}(x) = \frac{1}{|B(G_i)|}\sum_{b\in B(G_i)} \mathbb{1}_{b\leq x}$ where $\mathbb{1}_{b\leq x}$ is an indicator function taking the value 1 if $b \leq x$, and 0 otherwise. A pseudoinverse of $F_{G_i,B}$ is defined as $F^{-1}_{G_i,B}(z) = \inf\{b \in \mathbb{R} \,|\, F_{G_i,B}(b) \geq z\}$, i.e., $F^{-1}_{G_i,B}(z)$ is the smallest $b$ for which $F_{G_i,B}(b) \geq z$. Then the empirical Wasserstein distance for connected components has a closed-form solution in terms of pseudoinverses as $W_{p,B}(G_1, G_2) = \left(\int_0^1 |F^{-1}_{G_1,B}(z) - F^{-1}_{G_2,B}(z)|^p \, dz\right)^{1/p}$. 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

Extended Abstract Track

$G_2$ with the same partitions of $m$ and $n$, respectively. Then the approximated Wasserstein distances are given by $\widehat{W}_{p,B}(G_1, G_2) = \left(\frac{1}{m^p}\sum_{k=1}^{m}\left|F^{-1}_{G_1,B}(k/m) - F^{-1}_{G_2,B}(k/m)\right|^p\right)^{1/p}$ and $\widehat{W}_{p,D}(G_1, G_2) = \left(\frac{1}{n^p}\sum_{k=1}^{n}\left|F^{-1}_{G_1,D}(k/n) - F^{-1}_{G_2,D}(k/n)\right|^p\right)^{1/p}$.

**Vector Representation of Persistence Barcodes** A collection of 1-dimensional persistence barcodes together with the Wasserstein distance is a metric space. 1-dimensional 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, $\boldsymbol{v}_{B,i} := \left(F^{-1}_{G_i,B}(1/m), F^{-1}_{G_i,B}(2/m), ..., F^{-1}_{G_i,B}(m/m)\right)^\top$. 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 ($W_1$) (Rubner et al., 2000), 2-Wasserstein distance ($W_2$), and the bottleneck distance ($W_\infty$) (Kerber et al., 2017), 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 $M_B \times M_D = \{(\boldsymbol{v}_{B,i}, \boldsymbol{v}_{D,i}) \,|\, \boldsymbol{v}_{B,i} \in M_B, \boldsymbol{v}_{D,i} \in M_D\}$ be the Cartesian product between $M_B$ 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 illustrated in Figure 1. 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

$$\begin{aligned} d_{p,\times}\big((\boldsymbol{v}_{B,i}, \boldsymbol{v}_{D,i}), (\boldsymbol{v}_{B,j}, \boldsymbol{v}_{D,j})\big) &= \left([d_{p,B}(\boldsymbol{v}_{B,i}, \boldsymbol{v}_{B,j})]^p + [d_{p,D}(\boldsymbol{v}_{D,i}, \boldsymbol{v}_{D,j})]^p\right)^{1/p} \\ &= \left([m\widehat{W}_{p,B}]^p + [n\widehat{W}_{p,D}]^p\right)^{1/p}, \end{aligned}$$

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.

Statistical validation and runtime experiment for TopVS are provided in Appendix A.

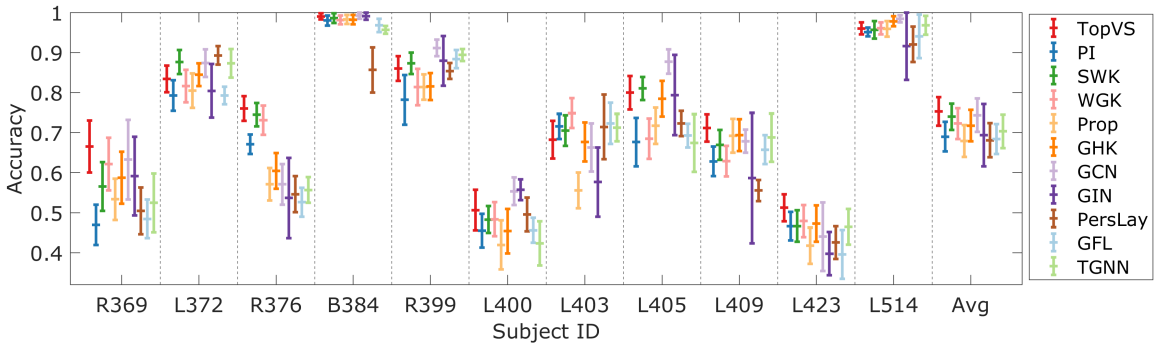


Figure 2: Accuracy classifying brain networks within individual subjects. The last column displays the average accuracy obtained across all subjects. The center markers and bars depict the means and standard deviations obtained over 100 trials.

## 3. Application to Functional Brain Networks

**Dataset** We evaluate our method using a brain network dataset from the anesthesia study reported by Banks et al. (2020) (see Appendix D for details). The measured brain networks are based on eleven neurosurgical patients during administration of increasing doses of the general anesthetic propofol prior to surgery. 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, graph kernels and graph neural networks including persistence image (PI) (Adams et al., 2017), sliced Wasserstein kernel (SWK) (Carriere et al., 2017), persistence weighted gaussian kernel (PWGK) (Kusano et al., 2016), propagation kernel (Prop) (Neumann et al., 2016), graph hopper kernel (GHK) (Feragen et al., 2013), graph convolutional networks (GCN) (Kipf and Welling, 2017), graph isomorphism network (GIN) (Xu et al., 2019a), PersLay (Carrière et al., 2020), graph filtration learning (GFL) (Hofer et al., 2020) and topological graph neural network (TGNN) (Horn et al., 2022). 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 ($11 \times 100$ scores) to obtain an overall accuracy. Supplementary description and implementation details of the performance evaluation and methods are provided in Appendix B.

**Results** Figure 2 compares classification accuracy for individual subjects. There is variability in performance across subjects and across methods. In most subjects all methods perform relatively well. The consistently poorer performance of PI, Prop, GIN, PersLay and GFL is evident in the lower overall performance. TopVS is demonstrated to perform favorably against the graph neural network classification methods. The results suggest that the use of complex classification methods, such as GCN, GIN, PersLay, GFL and TGNN, does not result in significant increase in generalizability when classifying brain networks. Our TopVS method is consistently among the best performing classifiers, resulting in the higher overall performance.

## References

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.

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.

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

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.

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

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.

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.

Peter Bubenik. Statistical topological data analysis using persistence landscapes. *Journal of Machine Learning Research*, 16:77–102, 2015.

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

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

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

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.

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.

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

Mathieu Carrière, Steve Y Oudot, and Maks Ovsjanikov. Stable topological signatures for points on 3D shapes. In *Computer Graphics Forum*, volume 34, pages 1–12. Wiley Online Library, 2015.

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

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*, pages 2786–2796. PMLR, 2020.

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.

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.

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.

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

David Cohen-Steiner, Herbert Edelsbrunner, John Harer, and Yuriy Mileyko. Lipschitz functions have Lp-stable persistence. *Foundations of Computational Mathematics*, 2010.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in Neural Information Processing Systems*, 26:2292–2300, 2013.

Michel Marie Deza and Elena Deza. Encyclopedia of distances. pages 1–583. Springer, 2009.

Herbert Edelsbrunner and John Harer. Persistent homology–a survey. *Contemporary Mathematics*, 453:257–282, 2008.

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

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.

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*, pages 216–224, 2013.

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

Christoph Hofer, Florian Graf, Bastian Rieck, Marc Niethammer, and Roland Kwitt. Graph filtration learning. In *International Conference on Machine Learning*, pages 4314–4323. PMLR, 2020.

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

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.

Max Horn, Edward De Brouwer, Michael Moor, Yves Moreau, Bastian Rieck, and Karsten Borgwardt. Topological graph neural networks. In *International Conference on Learning Representations*, 2022.

Xiaoling Hu, Fuxin Li, Dimitris Samaras, and Chao Chen. Topology-preserving deep image segmentation. *Advances in Neural Information Processing Systems*, 32, 2019.

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

Joel Keizer, Yue-Xian Li, Stanko Stojilković, and John Rinzel. InsP3-induced Ca2+ excitability of the endoplasmic reticulum. *Molecular Biology of the Cell*, 6(8):945–951, 1995.

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

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

Soheil Kolouri, Se Rim Park, Matthew Thorpe, Dejan Slepcev, and Gustavo K Rohde. Optimal mass transport: Signal processing and machine-learning applications. *IEEE Signal Processing Magazine*, 34(4):43–59, 2017.

Joseph B Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical society*, 7(1):48–50, 1956.

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

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.

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.

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.

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

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

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

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.

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.

Giovanni Petri, Martina Scolamiero, Irene Donato, and Francesco Vaccarino. Topological strata of weighted complex networks. *PLOS ONE*, 8(6):1–8, 2013.

Robert Clay Prim. Shortest connection networks and some generalizations. *The Bell System Technical Journal*, 36(6):1389–1401, 1957.

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*, pages 435–446. Springer, 2011.

Jan Reininghaus, Stefan Huber, Ulrich Bauer, and Roland Kwitt. A stable multi-scale kernel for topological machine learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4741–4748, 2015.

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.

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

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.

Primoz Skraba and Katharine Turner. Wasserstein stability for persistence diagrams. *arXiv preprint arXiv:2006.16824*, 2020.

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.

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/.

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.

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

KV Venkatesh, Sharad Bhartiya, and Anurag Ruhela. Multiple feedback loops are key to a robust dynamic performance of tryptophan regulation in Escherichia coli. *FEBS Letters*, 563(1-3):234–240, 2004.

Martin Vinck, Robert Oostenveld, Marijn Van Wingerden, Franscesco Battaglia, and Cyriel MA Pennartz. An improved index of phase-synchronization for electrophysiological data in the presence of volume-conduction, noise and sample-size bias. *NeuroImage*, 55(4):1548–1565, 2011.

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

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.

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

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019a.

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, 2019b.

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

Afra Zomorodian. Fast construction of the Vietoris-Rips complex. *Computers & Graphics*, 34(3):263–271, 2010.

## Appendix A.

### A.1. Motivation

Networks are ubiquitous representations for describing complex, highly interconnected systems that capture potentially intricate patterns of relationships between nodes. (Barrat et al., 2004). Finding meaningful characterizations of network structure is very difficult, especially for large and dense networks with node degrees ranging over multiple orders of magnitude (Bullmore and Sporns, 2009; Honey et al., 2007).

Persistent homology (Barannikov, 1994; Edelsbrunner et al., 2000; Wasserman, 2018) is an emerging tool for understanding, characterizing and quantifying the topology of complex networks (Carrière et al., 2020; Songdechakraiwut et al., 2021). Topology is characterized using connected components (0-dimensional topological features), cycles (1-dimensional topological features), voids (2-dimensional 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 (Bullmore and Sporns, 2009; Honey et al., 2007). Similarly, cycle structure is ubiquitous and is often interpreted in terms of information propagation, redundancy and feedback loops (Keizer et al., 1995; Kwon and Cho, 2007; Ozbudak et al., 2005; Venkatesh et al., 2004; Weiner et al., 2002). While voids have recently been considered in cosmology, they tend to be relatively rare, thus reducing their discriminative power (Biagetti et al., 2021). Anything of higher dimension than the void is beyond the apparent physical space of how we see the world, and thus is difficult to understand and interpret. This motivates use of only connected components and cycles to differentiate networks.

Topological features are represented using descriptors called *persistence barcodes* (Ghrist, 2008). Effective use of such topological descriptors requires a notion of proximity, that is, a metric that quantifies the distance between persistence barcodes. However, incorporating barcodes into learning tasks requires more than identifying a suitable distance metric. In general persistence barcodes do not have the algebraic structures required by a large class of learning methods such as support vector machines (SVMs). For example, it is unclear whether elementary operations such as addition, scalar multiplication, and inner product have analogues in the space of persistence barcodes.

Approaches that embed persistence barcodes into a space with adequate structure such as vector spaces or Hilbert spaces have recently been proposed to address this limitation. For example, Adams et al. (2017); Bubenik (2015); Carrière et al. (2015) describe methods for extracting vector representations from persistence barcodes. Other authors (Carriere et al., 2017; Kusano et al., 2016; Reininghaus et al., 2015) define implicit feature representations in a Hilbert space using the kernel trick (Hofmann et al., 2008). However, none of these methods preserves the underlying distance in the original space of persistence barcodes (Carrière and Bauer, 2019).

### A.2. 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

Extended Abstract Track

vectorization methods that utilize limited topological information extracted from barcodes, such as statistics (Lee et al., 2012), Betti numbers (Cang et al., 2015) and tropical coordinates (Kališnik, 2019), to more systematic featurizations, such as binning approach (Cang et al., 2018), persistent codebooks (Zieliński et al., 2021), persistent functions (Adams et al., 2017; Bubenik and Dłotko, 2017; Xia et al., 2018) and persistent paths (Chevyrev et al., 2020). 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 (Cohen-Steiner et al., 2007). It is thus of great interest to understand if such properties carry over to barcode embeddings. Previously proposed feature maps (Adams et al., 2017; Bubenik and Dłotko, 2017; Xia et al., 2018) achieve limited stability guarantees (Carrière and Bauer, 2019). 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 (Bubenik and Kim, 2007; Chazal et al., 2014; Fasy et al., 2014; Turner et al., 2014). 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 (Kerber et al., 2017; Lacombe et al., 2018) 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 (Bullmore and Sporns, 2009; Honey et al., 2007). While higher-order topological features have been considered in cosmology (Xu et al., 2019b), recent work shows that cosmic voids are not as discriminative as filament loops, i.e., cycles, due to their relatively rare occurrence (Biagetti et al., 2021). Hence, while higher-order topological information is of theoretical interest, it is rarely used in applications of topology.

### A.3. One Dimensional Persistence Barcodes

#### A.3.1. Graph Filtration

Define a network as an undirected weighted graph $G = (V, \boldsymbol{w})$ with a set of nodes $V$, and a weighted adjacency matrix $\boldsymbol{w} = (w_{ij})$. The number of nodes is denoted by $|V|$. Define a binary graph $G_\epsilon$ 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 simplicial complex consisting of only nodes and edges, that is, a 1-skeleton (Munkres, 2018). As $\epsilon$ increases, more and more edges are removed from the network $G$. Thus, we have a nested sequence of 1-skeletons:

$$G_{\epsilon_0} \supseteq G_{\epsilon_1} \supseteq \cdots \supseteq G_{\epsilon_k}, \tag{1}$$

where $\epsilon_0 \leq \epsilon_1 \leq \cdots \leq \epsilon_k$ are called filtration values. This sequence of 1-skeletons is called a *graph filtration* (Lee et al., 2012). Figure 1 illustrates the graph filtration of a four-node network. Note that other filtrations for analyzing graphs have been proposed based on

descriptor functions such as heat kernels (Carrière et al., 2020) and task-specific learning (Hofer et al., 2020), in contrast to the use of edge weights.

### A.3.2. Birth-death Decomposition

Persistent homology keeps track of the birth and death of topological features over filtration values $\epsilon$. A topological feature that is born at a filtration $b_i$ and persists up to a filtration $d_i$, is represented as a 2-dimensional point $(b_i, d_i)$ in a plane. A set of all the points $\{(b_i, d_i)\}$ is called *persistence diagram* (Edelsbrunner and Harer, 2008) or, equivalently, *persistence barcode* (Ghrist, 2008). In the 1-skeleton, the only non-trivial topological features are connected components (0-dimensional topological features) and cycles (1-dimensional topological features). There are no higher-dimensional topological features in the 1-skeleton, in contrast to clique complexes (Otter et al., 2017; Zomorodian, 2010) and Rips complexes (Ghrist, 2008). This simplifies the persistence barcode and significantly reduces the computational complexity of the corresponding topological analysis.

The graph filtration given in (1) begins with a complete graph $G_{-\infty}$, sequentially removes edges at higher filtration values $\epsilon$, and arrives at an edgeless graph $G_{+\infty}$. As $\epsilon$ increases, the number of connected components $\beta_0(G_\epsilon)$ and cycles $\beta_1(G_\epsilon)$ are monotonically increasing and decreasing, respectively (Songdechakraiwut et al., 2021). Specifically, $\beta_0(G_\epsilon)$ increases from the complete graph consisting of a single connected component $\beta_0(G_{-\infty}) = 1$ to the node set $\beta_0(G_\infty) = |V|$. There are $\beta_0(G_\infty) - \beta_0(G_{-\infty}) = |V| - 1$ connected components that are born over the filtration.

Once connected components are born, they will remain until the node set $G_{+\infty}$ is reached, so all death values are at $+\infty$. Thus, the representation of the connected components can be simplified to a collection of sorted birth values $B(G) = \{b_i\}_{i=1}^{|V|-1}$. On the other hand, all cycles are born with the complete graph $G_{-\infty}$ and thus have birth values at $-\infty$. Again we can simplify the representation of the cycles as a collection of sorted death values $D(G) = \{d_i\}$. The removal of an edge must result in either the birth of a connected component or the death of a cycle. Thus every edge weight must also be in either $B(G)$ or $D(G)$, resulting in the decomposition of the edge weight set $W = \{w_{ij} \,|\, i > j\}$ into $B(G)$ and $D(G)$ (Songdechakraiwut et al., 2021). Since the complete graph $G_{-\infty}$ has $\frac{|V|(|V|-1)}{2}$ edge weights, the number of cycles in $G_{-\infty}$ is equal to $\frac{|V|(|V|-1)}{2} - (|V| - 1) = 1 + \frac{|V|(|V|-3)}{2}$. Thus, every different network with the same node size $|V|$ has a birth set $B$ and a death set $D$ of the same cardinality as $|V| - 1$ and $1 + \frac{|V|(|V|-3)}{2}$, respectively. Note that other filtrations (Carrière et al., 2020; Ghrist, 2008; Hofer et al., 2020; Otter et al., 2017; Petri et al., 2013; Zomorodian, 2010) do not necessarily share this monotonicity property. Thus, their persistence barcodes are not 1-dimensional, and the number of points in the persistence barcodes may vary for different networks of the same size.

$B(G)$ comprises edge weights in the *maximum spanning tree* (MST) of $G$ (Lee et al., 2012), and can be computed using standard methods such as Kruskal's (Kruskal, 1956) and Prim's algorithms (Prim, 1957). Once $B(G)$ is identified, $D(G)$ is given as the remaining edge weights that are not in the MST. Thus $B(G)$ and $D(G)$ are computed very efficiently in $O(|V|^2 \log |V|)$ operations. The example network of Figure 1 has $B(G) = \{e_3, e_5, e_6\}$ and $D(G) = \{e_1, e_2, e_4\}$.

Extended Abstract Track

### A.4. Topological Space with Wasserstein Distance

#### A.4.1. Wasserstein Distance Simplification

The graph filtration given in (1) results in significant simplification of the Wasserstein distance between barcode descriptors of networks. Let $\boldsymbol{X}$ and $\boldsymbol{Y}$ be 2-dimensional random vectors describing the stochastic nature of points in conventional 2-dimensional persistence barcodes (Ghrist, 2008). Let $\phi_{\boldsymbol{X}}$ and $\phi_{\boldsymbol{Y}}$ be probability distributions of $\boldsymbol{X}$ and $\boldsymbol{Y}$, respectively. The $p$-Wasserstein distance between $\phi_{\boldsymbol{X}}$ and $\phi_{\boldsymbol{Y}}$ is defined as (Kolouri et al., 2017)

$$W_p(\phi_{\boldsymbol{X}}, \phi_{\boldsymbol{Y}}) := \Big( \inf_{\phi_{\boldsymbol{X},\boldsymbol{Y}} \in \Phi} \int_{\mathbb{R}^2} (||x-y||_p)^p \, d\phi_{\boldsymbol{X},\boldsymbol{Y}}(x,y) \Big)^{1/p}, \tag{2}$$

where $||\cdot||_p$ denotes the $p$-norm, and the infimum is taken over all joint distributions $\Phi$ of the random vectors $\boldsymbol{X}$ and $\boldsymbol{Y}$ with marginal distributions $\phi_{\boldsymbol{X}}$ and $\phi_{\boldsymbol{Y}}$. Intuitively, each distribution is a unit mass, and the Wasserstein metric measures the optimal transport plan with the minimum work of turning one mass into the other by moving points over $p$-norm induced distance.

Solving the optimization problem in (2) is computationally costly and multiple approximation algorithms have been proposed to manage its computational complexity (see, e.g., (Cuturi, 2013; Kerber et al., 2017; Lacombe et al., 2018; Rabin et al., 2011)). Consequently, use of Wasserstein distance with the 2-dimensional persistence barcodes resulting from conventional filtrations (Carrière et al., 2020; Ghrist, 2008; Hofer et al., 2020; Otter et al., 2017; Petri et al., 2013; Zomorodian, 2010) is computationally challenging.

On the other hand, the Wasserstein distance between the 1-dimensional barcodes of the graph filtration defined in (1) can be obtained using a closed-form solution. Let $G_1$ and $G_2$ be two given networks possibly with different node sizes, i.e., their birth and death sets may vary in size. Their underlying probability density functions on the persistence barcodes for connected components are defined in the form of Dirac masses (Turner et al., 2014):

$$f_{G_1,B}(x) := \frac{1}{|B(G_1)|} \sum_{b \in B(G_1)} \delta(x-b), \quad f_{G_2,B}(x) := \frac{1}{|B(G_2)|} \sum_{b \in B(G_2)} \delta(x-b), \tag{3}$$

where $\delta(x-b)$ is a Dirac delta centered at the point $b$. Then the empirical distributions are the integration of $f_{G_1,B}$ and $f_{G_2,B}$ as

$$F_{G_1,B}(x) = \frac{1}{|B(G_1)|} \sum_{b \in B(G_1)} \mathbb{1}_{b \le x}, \quad F_{G_2,B}(x) = \frac{1}{|B(G_2)|} \sum_{b \in B(G_2)} \mathbb{1}_{b \le x}, \tag{4}$$

where $\mathbb{1}_{b \le x}$ is an indicator function taking the value 1 if $b \le x$, and 0 otherwise. A pseudoinverse of $F_{G_1,B}$ is defined as $F^{-1}_{G_1,B}(z) = \inf\{b \in \mathbb{R} \,|\, F_{G_1,B}(b) \ge z\}$, i.e., $F^{-1}_{G_1,B}(z)$ is the smallest $b$ for which $F_{G_1,B}(b) \ge z$. Similarly, we define a pseudoinverse of $F_{G_2,B}$ as $F^{-1}_{G_2,B}(z) = \inf\{b \in \mathbb{R} \,|\, F_{G_2,B}(b) \ge z\}$. Then the empirical Wasserstein distance for connected components has a closed-form solution in terms of these pseudoinverses as (Kolouri et al., 2017)

$$W_{p,B}(G_1, G_2) = \Big( \int_0^1 |F^{-1}_{G_1,B}(z) - F^{-1}_{G_2,B}(z)|^p \, dz \Big)^{1/p}. \tag{5}$$

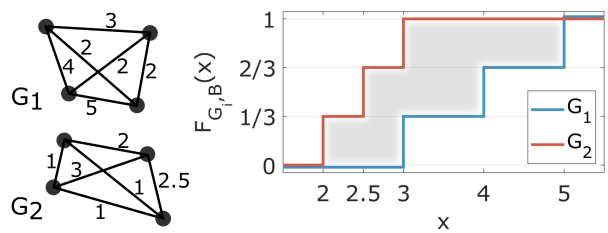


Figure 3: Four-node networks $G_1$ and $G_2$ have birth sets of connected components as $B(G_1) = \{3, 4, 5\}$ and $B(G_2) = \{2, 2.5, 3\}$, respectively. The 1-Wasserstein distance $W_{1,B}(G_1, G_2) = \frac{1}{3}\big(|3-2| + |4-2.5| + |5-3|\big)$, which is exactly equal to the shaded area situated between two empirical distributions $F_{G_1,B}$ and $F_{G_2,B}$.

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 in (5) numerically as follows. Let $\widehat{B}(G_1) = \{F^{-1}_{G_1,B}(1/m), F^{-1}_{G_1,B}(2/m), ..., F^{-1}_{G_1,B}(m/m)\}$ 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

$$\widehat{W}_{p,B}(G_1, G_2) = \Big(\frac{1}{m^p}\sum_{k=1}^{m}\big|F^{-1}_{G_1,B}(k/m) - F^{-1}_{G_2,B}(k/m)\big|^p\Big)^{1/p}, \tag{6}$$

$$\widehat{W}_{p,D}(G_1, G_2) = \Big(\frac{1}{n^p}\sum_{k=1}^{n}\big|F^{-1}_{G_1,D}(k/n) - F^{-1}_{G_2,D}(k/n)\big|^p\Big)^{1/p}. \tag{7}$$

For the special case when networks $G_1$ and $G_2$ have the same number of nodes, the exact computation of the Wasserstein distance is achieved using the original birth and death sets $B(G_1), B(G_2), D(G_1)$, and $D(G_2)$ as (Songdechakraiwut et al., 2021)

$$W_{p,B}(G_1, G_2) = \Big(\frac{1}{|B(G_1)|^p}\sum_{b\in B(G_1)}|b - \tau_0^*(b)|^p\Big)^{1/p}, \tag{8}$$

$$W_{p,D}(G_1, G_2) = \Big(\frac{1}{|D(G_1)|^p}\sum_{d\in D(G_1)}|d - \tau_1^*(d)|^p\Big)^{1/p}, \tag{9}$$

where $\tau_0^*$ maps the $l$-th smallest birth value in $B(G_1)$ to the $l$-th smallest birth value in $B(G_2)$, and $\tau_1^*$ maps the $l$-th smallest death value in $D(G_1)$ to the $l$-th smallest death value in $D(G_2)$, for all $l$. The exact Wasserstein distances $W_{p,B}$ and $W_{p,D}$ are well-defined since the bijective mapping $\tau_0^*$ between same-cardinality sets of births is well-defined, as does $\tau_1^*$ for same-cardinality sets of deaths is well-defined. Figure 3 illustrates the exact computation of the 1-Wasserstein distance for connected components between four-node networks.

Extended Abstract Track

### A.4.2. Vector Representation of Persistence Barcodes

A collection of 1-dimensional persistence barcodes together with the Wasserstein distance is a metric space. 1-dimensional 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 $1/m, 2/m, ..., m/m$:

$$\boldsymbol{v}_{B,i} := \left(F^{-1}_{G_i,B}(1/m), F^{-1}_{G_i,B}(2/m), ..., F^{-1}_{G_i,B}(m/m)\right)^\top. \tag{10}$$

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

$$d_{p,B}(\boldsymbol{v}_{B,i}, \boldsymbol{v}_{B,j}) = ||\boldsymbol{v}_{B,i} - \boldsymbol{v}_{B,j}||_p = m\widehat{W}_{p,B}. \tag{11}$$

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 ($W_1$) (Rubner et al., 2000), 2-Wasserstein distance ($W_2$), and the bottleneck distance ($W_\infty$) (Kerber et al., 2017), 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 $M_B \times M_D = \{(\boldsymbol{v}_{B,i}, \boldsymbol{v}_{D,i}) \,|\, \boldsymbol{v}_{B,i} \in M_B, \boldsymbol{v}_{D,i} \in M_D\}$ be the Cartesian product between $M_B$ 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 discussed in Appendix A.3.2. 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 (Deza and Deza, 2009)

$$d_{p,\times}\big((\boldsymbol{v}_{B,i}, \boldsymbol{v}_{D,i}), (\boldsymbol{v}_{B,j}, \boldsymbol{v}_{D,j})\big) = \left([d_{p,B}(\boldsymbol{v}_{B,i}, \boldsymbol{v}_{B,j})]^p + [d_{p,D}(\boldsymbol{v}_{D,i}, \boldsymbol{v}_{D,j})]^p\right)^{1/p} \tag{12}$$

$$= \left([m\widehat{W}_{p,B}]^p + [n\widehat{W}_{p,D}]^p\right)^{1/p}, \tag{13}$$

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). Note the form of $d_{p,\times}$ given in (13) results in an unnormalized mass after multiplying $m$ and $n$ by their reciprocals given in (6) and (7). This unnormalized variant of Wasserstein distance is widely used in both theory (Cohen-Steiner et al., 2010; Skraba and Turner, 2020) and application (Carriere

et al., 2017; Hu et al., 2019; Songdechakraiwut et al., 2021) of persistent homology. A direct consequence of the equality given in (13) 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.

For a special case in which networks $G_1, G_2, ..., G_N$ have the same number of nodes, the vectors $\boldsymbol{v}_{B,i}$ and $\boldsymbol{v}_{D,i}$ are simply the original birth set $B(G_i)$ and death set $D(G_i)$, respectively, and the $p$-norm metric $d_{p,\times}$ is expressed in terms of exact Wasserstein distances as

$$d_{p,\times} = ([mW_{p,B}]^p + [nW_{p,D}]^p)^{1/p}. \tag{14}$$

### A.5. 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) (Chang and Lin, 2011) 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 as discussed in Appendix A.3.2, 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 (Adams et al., 2017), the *Sliced Wasserstein* kernel (SWK) (Carriere et al., 2017) and the *Persistence Weighted Gaussian* kernel (PWGK) (Kusano et al., 2016). The other two benchmark methods are based on graph kernels: the *Propagation* kernel (Prop) (Neumann et al., 2016) and the *GraphHopper* kernel (GHK) (Feragen et al., 2013). 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 (Hofmann et al., 2008) to perform classification.

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. (2017) 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 (Tralie et al., 2018) of the Rips filtration (Ghrist, 2008) 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. (2020) 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 C.

**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

Extended Abstract Track

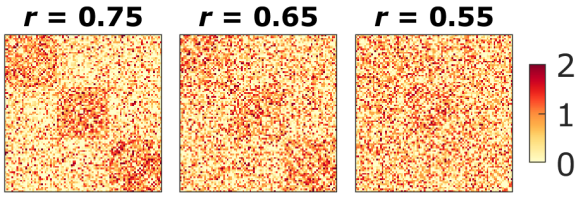


Figure 4: Example networks with $|V| = 90$ nodes distributed evenly among $m = 3$ modules according to within-module-connection probabilities $r = 0.75, 0.65$ and $0.55$.

preserving the percentage of network samples for each group label. The inner loop is used to tune hyperparameters via grid search (Bergstra and Bengio, 2012) 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 C.

**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 4 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 4. 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

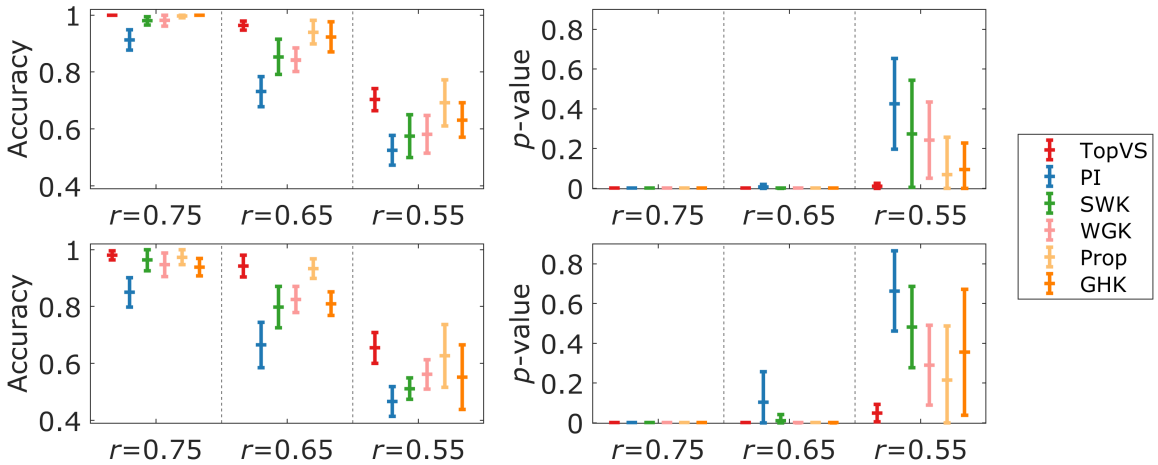


Figure 5: Classification performance comparison for simulated networks with $m = 3, 5$ modules and either $|V| = 90$ nodes (Top) or $|V| = 60, 90, 120$ nodes (bottom) with respect to average accuracy (left) and average $p$-values (right). Results for within-module connection probabilities $r = 0.55, 0.65$ 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.

under the null hypothesis that sample networks and their group labels are independent (Ojala and Garriga, 2010). 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 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 5 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 (Ojala and Garriga, 2010). 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.

Extended Abstract Track

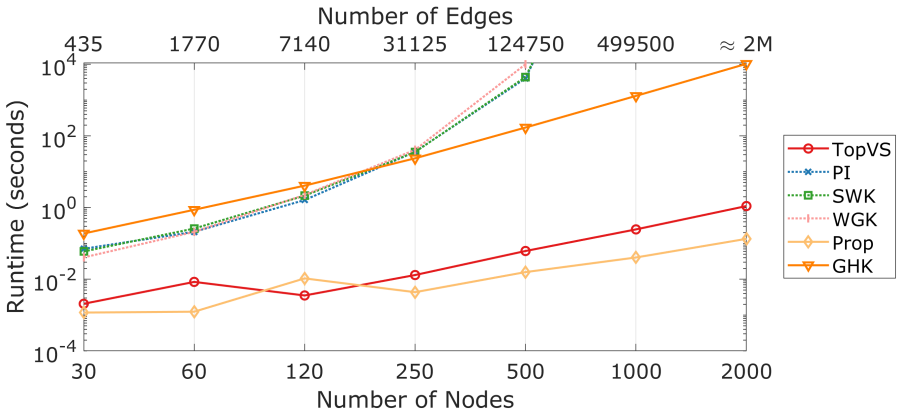


Figure 6: Runtime experiment. We measured the runtime as the average amount of time each algorithm takes to compute its kernel between two complete graphs starting from edge weights as a given input. The complete graphs were constructed using our modular network generation, described in Appendix A.5. The runtime is plotted with respect to network size in terms of both the number of nodes and edges.

### A.6. 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 6 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.

## Appendix B. Application to Functional Brain Networks

**Dataset** We evaluate our method using an extended brain network dataset from the anesthesia study reported by Banks et al. (2020) (see Appendix D for details). The measured brain networks are based on alpha band (8-12 Hz) weighted phase lag index (Vinck et al., 2011) applied to 10-second segments of resting state intracranial electroencephalography recordings. These recordings were made from eleven neurosurgical patients during administration of increasing doses of the general anesthetic propofol just prior to surgery. Each segment is labeled as one of the three arousal states: pre-drug *wake* (W), *sedated* but

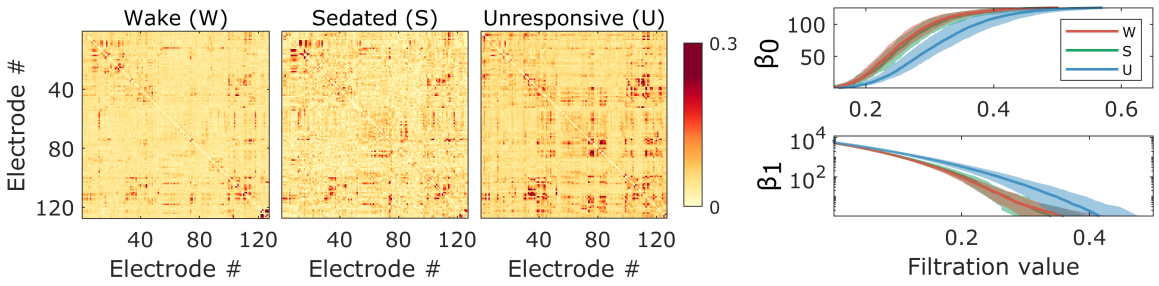


Figure 7: Data visualization of an example subject (ID L405). Left: Sample mean networks of wake, sedated, and unresponsive brains. Right: Network topology is completely characterized by connected component count $\beta_0$ and cycle count $\beta_1$. A thick line representing the mean of persistence barcodes and shaded areas around the mean representing standard deviation.

responsive to command (S), or *unresponsive* (U). The number of networks (10-second segments) per subject varies from 71 to 119. The network size varies from 89 to 199 nodes across subjects. Figure 7 illustrates sample mean networks and 1-dimensional persistence barcodes describing topology for a representative subject.

**Classification performance evaluation** All candidate methods used in the simulation study, described in Appendix A.5, are evaluated on the brain network dataset. In addition, we also evaluate state-of-the-art graph neural network methods including graph convolutional networks (GCN) (Kipf and Welling, 2017), graph isomorphism network (GIN) (Xu et al., 2019a), PersLay (Carrière et al., 2020), graph filtration learning (GFL) (Hofer et al., 2020) and topological graph neural network (TGNN) (Horn et al., 2022). Implementation details of all the baseline methods are provided in Appendix C.

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 ($11 \times 100$ 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.

**Results** Figure 2 compares classification accuracy for individual subjects. There is variability in performance across subjects and across methods. In most subjects all methods perform relatively well. The PI, Prop, GIN, PersLay and GFL methods perform poorest in most subjects. The consistently poorer performance of PI, Prop, GIN, PersLay and

Extended Abstract Track

Table 1: Mean accuracy and standard deviation for classifying brain networks of test subjects not used for training.

| TopVS | PI | SWK | WGK | Prop | GHK |
|---|---|---|---|---|---|
| $0.65 \pm 0.21$ | $0.58 \pm 0.22$ | $0.57 \pm 0.20$ | $0.60 \pm 0.21$ | $0.36 \pm 0.12$ | $0.43 \pm 0.14$ |

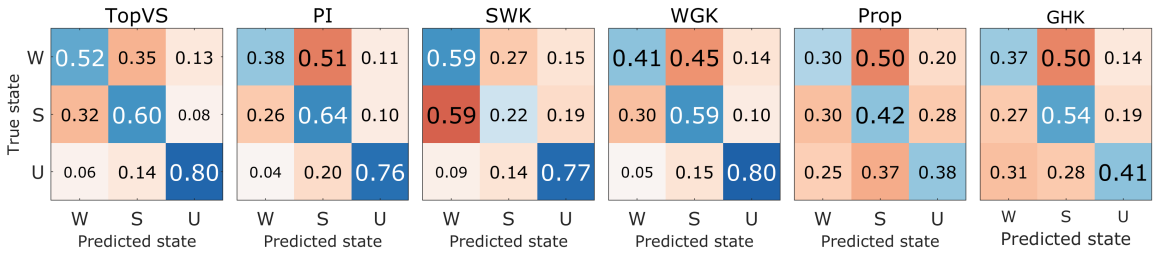


Figure 8: Confusion matrices illustrating method performance for classifying across subjects. The numbers represent the fraction of brain networks in the test subjects being predicted as one of the three possible states: wake (W), sedated (S), and unresponsive (U). The confusion matrices are normalized with the entries in each row summing to 1.

GFL is evident in the lower overall performance. Although the Prop method is among the best performing methods for the simulated networks, here it suffers when presented with functional brain networks. TopVS is demonstrated to perform favorably against the graph neural network classification methods. The results suggest that the use of complex classification methods, such as GCN, GIN, PersLay, GFL and TGNN, does not result in significant increase in generalizability when classifying brain networks. Our TopVS method is consistently among the best performing classifiers, resulting in the higher overall performance. TopVS has the best generalizability performance based on average accuracy.

Table 1 shows that TopVS is also among the best methods for classifying across subjects, while the performance of both graph kernel methods suffers. Figure 8 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 (Banks et al., 2020). 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.

TopVS is a vector representation for 1-dimensional persistence barcodes of connected components and cycles in complex networks. The computational practicality of TopVS extends its applicability to the large-scale analyses of complex networks that cannot be analyzed using prior methods based on 2-dimensional persistence barcodes. In addition, TopVS is a discriminative representation of complex networks with sound theoretical grounding based on the Wasserstein distance of the underlying barcode space. Our simulated and human-brain network studies demonstrate that TopVS performs well across a wide variety of classification scenarios relative to state-of-the-art methods in the persistent homology literature. Lastly, TopVS is interpretable and allows the computation of summary statistics for statistical inference. The practical computation, discriminative power and interpretability of TopVS will have a high impact on the analyses of large and complex network representations.

## Appendix C. Implementation Details of Candidate Methods

Implementation of *Persistence Image* (PI) vectorization (Adams et al., 2017) is performed using Persim (Saul and Tralie, 2019). *Sliced Wasserstein* kernel (SWK) (Carriere et al., 2017) and *Persistence weighted Gaussian* kernel (PWGK) (Kusano et al., 2016) are implemented using the Gudhi library (The GUDHI Project, 2022). *Propagation* kernel (Prop) (Neumann et al., 2016) and *GraphHopper* kernel (GHK) (Feragen et al., 2013) are implemented via GraKel library (Siglidis et al., 2020).

For the PI method, we follow a parameter setting used by Adams et al. (2017) to convert two persistence barcodes for connected components and cycles into two 2-dimensional pixel images of $20 \times 20$ 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 (Rabin et al., 2011) over 10 directions. PWGK is based on the RBF kernel and an arctan weight function recommended by Kusano et al. (2016). 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 (Bergstra and Bengio, 2012) 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 $(C, \gamma) \in \mathcal{C} \times \Gamma$.

For *graph convolutional networks* (GCN) (Kipf and Welling, 2017) and *graph isomorphism network* (GIN) (Xu et al., 2019a), we follow configurations and choices of hyperparameter values of Xu et al. (2019a) as follows. Five graph neural network layers are applied, and Adam optimizer with initial learning rate and weight decay of 0.01 is employed. We tune the following hyperparameters: the number of hidden units in $\{16, 32\}$, the batch size in $\{32, 128\}$ and the dropout ratio in $\{0, 0.5\}$. The number of epochs is set to 200 to train

Extended Abstract Track

both methods. The average performance is reported over ten independent trials. PyTorch Geometric library is used to implement GCN and GIN.

PersLay (Carrière et al., 2020) has several hyperparameters, and their choices are non-canonical. The work in (Carrière et al., 2020) does not provide explicit guidance for choosing these parameters, and their benefits over others are unclear. To address this challenge, we employ the six predefined architectures used in (Carrière et al., 2020), which are publicly available at the authors' code repository website. We report the performance of the model architecture with the highest average accuracy. We tune the batch size in $\{32, 128\}$ using nested cross validation and use the number of epochs as 200.

The predefined *graph filtration learning* (GFL) model used by the original authors of (Hofer et al., 2020) is employed to classify brain networks. The details of the configurations and choices of hyperparameter values are publicly available at the authors' code repository website. We closely follow the training procedure provided in (Hofer et al., 2020), which does not tune any hyperparameter.

For *topological graph neural network* (TGNN) method (Horn et al., 2022), we closely follow the setup from (Horn et al., 2022) by using the same architecture and same choices of hyperparameter values used by the original authors. That is, the predefined architecture with publicly available implementation at the authors' code repository website is employed, and the hyperparameter values are provided in the appendix of (Horn et al., 2022). The number of max epochs is 200.

## Appendix D. Brain Network Dataset

Brain network data were obtained from eleven neurosurgical patients between 19 and 59 years old as described in Table 2. 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 (Banks et al., 2020).

Table 2: 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 |