MODELING HIERARCHICAL TOPOLOGICAL STRUCTURE IN SCIENTIFIC IMAGES WITH GRAPH NEURAL NETWORKS

Samuel Leventhal [*‡] *Attila Gyulassy*[‡] *Valerio Pascucci*[‡] *Mark Heimann*[†]

[‡] University of Utah, Salt Lake City, UT, USA
[†]Lawrence Livermore National Laboratory, Livermore, CA, USA

2023 IEEE International Conference on Image Processing (ICIP) | 978-1-7281-9835-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICIP49359.2023.10222089

ABSTRACT

Topological analysis reveals meaningful structure in data from a variety of domains. Tasks such as image segmentation can be effectively performed on an image's topological connectivity using graph neural networks (GNNs). We propose two methods for using GNNs to learn from the hierarchical information captured by complexes at multiple levels of topological persistence: one modifies the training procedure of an existing GNN, and one extends the message passing across all levels of the complex. Experiments on real-world data from three domains show the performance benefits to GNNs from using a hierarchical topological structure.

***Index Terms*—** graph neural networks, topological data analysis, image segmentation, persistence, Morse-Smale complex

1. INTRODUCTION

Topological data analysis has been used in many application domains, such as segmentation of neurons [1], structural components of interest in metallic foams [2], eddies in ocean currents [3], bubble formation in mixing fluids [4], and ignition kernels in combustion [5]. It has also served as a preprocessing step for machine learning tasks [6], and topological principles have influenced the design of loss functions [7] and architectures [8, 9] for deep neural networks.

In each case, semantic objects appear as collections of data structures encoding the topological abstraction. For example, vertices adjoined by higher dimensional polylines result in a graph representation of a topological complex. Using this graph representation, we formulate the task of image segmentation as a node classification problem, which we solve using graph neural networks (GNNs). Instead of deriving only one high-granularity complex, we model multiple scales of topological information in an image using a nested hierarchy of graph representations, each derived from different levels of topological persistence. Our work makes the following contributions:

- We propose two methods for applying GNNs to a hierarchy of topological graphs: one trains a conventional GNN on each successive level, and the other extends the message passing to take place jointly across all levels.
- We use GNNs to perform scientific image segmentation operating on the topological complex, for which we have developed an interactive tool to allow a human to label training data. We show a performance improvement by learning from hierarchical topological structure.

[*]This work was funded in part by NSF OAC award 2138811, NSF CI CoE Award 2127548 Department of Energy (DoE) award DE-FE0031880, and the Intel oneAPI Centers of Excellence at University of Utah, the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the DoE and the NNSA, and UT-Battelle, LLC under contract DE-AC05-00OR22725 as well as under the auspices of the U.S. DoE while an intern at Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. Supported by the LDRD Program under project 21-ERD-012.

2. BACKGROUND: COMPUTATIONAL TOPOLOGY

2.1. The Morse-Smale Complex (MSC)

A Morse function $f : \mathscr{M} \to \mathbb{R}$ is a smooth function on a manifold with nondegenerate, distinct critical points. The gradient, ∇f, defines a vector field with critical points as zeroes. *Integral lines* are paths tangent to ∇f with limits at critical points of f. Noncritical points in $\mathscr{M}$ belong to a single integral line, with upper and lower limits at critical points (called *destination* and *origin*). The domain is partitioned into monotonic connected components defined by integral lines that share origin and destination points, forming the MSC. *Cells* of this complex have a dimension equal to the difference between the number of source directions between the destination and origin critical points of their constituent integral lines. The *1-skeleton* of the complex is formed by critical points and the *arcs* or integral lines that connect them, where these differ in index by 1. Concepts from continuous functions can be applied to a discrete pixel space using discrete Morse theory [10], for which we use the open-source MSCEER library [11] to compute a discrete MSC.

2.2. Topological Simplification

Topological abstractions come equipped with well-understood techniques to order and simplify their elements to obtain successively coarser representations. For example, *topological persistence* allows for a multiscale simplification by pairing the critical point that creates a connected component with the critical point that destroys that component during a filtration [12]. The time span in the filtration in which the connected component lives, i.e. the difference in function value between the birth and death critical points, is called persistence. The admissible persistence defining this filtration defines the granularity of the resulting MSC [12, 13, 14].

2.3. Topology in the Context of Digital Images

In various image analysis tasks, like segmentation, it is common to initially convert the original image representation (e.g., RGB) into a single scalar value. For instance, for object detection in images, a successful approach involves converting multichannel image data to grayscale, applying a Sobel filter, and computing watershed regions [15, 16, 17]. Similarly, in the context of topology, persistent homology has been applied to images to better understand root architectures, identify cells in microscopy, and more [18, 19, 20].

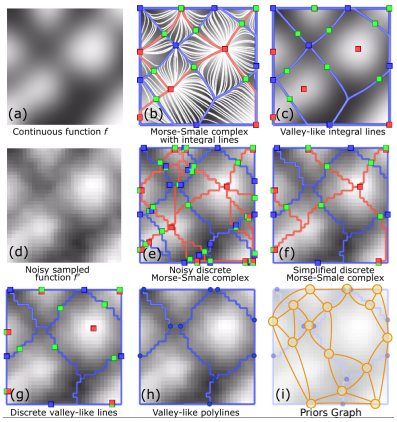


Fig. 1: Morse-Smale complexes are defined for functions with continuous gradients **(a-c)**. A smooth function **(a)** can be partitioned based on the behavior of integral lines **(b)** (selected integral lines shown in white). This partition forms a cell complex, where integral lines within each cell share a common origin and destination. The 0-dimensional cells are maxima (red), saddles (green), and minima (dark blue); the 1-dimensional cells are formed by ascending (orange) and descending lines (light blue) from saddles (green); and the 2-dimensional cells are bounded by 0- and 1-cells **(b)**. Elements of this complex often correspond to meaningful features in scientific domains, such as valley-like lines **(c)**. Real-world functions often come from noisy sources and are available as samples on a grid **(d)**. Discrete Morse-theory-based methods allow practical computation of Morse-Smale complexes **(e)**, which encode both noise and discretization artifacts that may be simplified to recover the coarse-scale behavior of the function **(f)**. The valley-like structures may be extracted from this complex **(g)**, and converted to a set of priors between non-degree-2 vertices denoted the valley graph **(h)**. The priors graph (yellow), **(i)**, represents each prior as a vertex with edges between incident priors.

2.4. Topological Priors Graph

To minimize the effects of image noise, we simplify the initial computation of the discrete MSC by canceling nodes connected by a single arc in the 1-skeleton. We then represent the image with a refinement of its MSC called the *ridge graph* [1], whose edges are polyline segments of the MSC. Our *priors graph* is the line graph representation [21] of the ridge graph, in which ridge graph edges or polylines become nodes that are classified as foreground or background. The priors graph allows us to segment the image through a graph representation of its topological structure. We illustrate the MSC, ridge graph, and our priors graph construction in Fig. 1.

2.5. Topological Graph Hierarchy

To model multiscale topological information, we compute the MSC at P levels of persistence. In contrast to hierarchical graph-level representation learning methods that learn to pool each input graph [22], this gives us several graph representations of the input to train a graph neural network. Computationally, a smaller persistence value produces a finer granularity graph, higher in the MSC hierarchy. Thus, we obtain a hierarchy of graphs $G_1,\ldots,G_P$ where $\forall p_i$ for $i \in [1,\ldots,P], G_{p_{i-1}} \subseteq G_{p_i}$: that is, $G_{p_{i-1}}$ is an induced subgraph defined on a subset of the vertices in G_{p_i}. Node neighborhoods thus share this property: for each node $v_i \in \mathbf{V}_{p_i} \cap \mathbf{V}_{p_j}$, e.g. $\mathcal{N}_{p_i}(v_i)$ for $v_i \in \mathbf{V}_{p_i}$ and $\mathcal{N}_{p_j}(v_i)$ for $v_i \in \mathbf{V}_{p_j}$, we have that $\mathcal{N}_{p_i}(v_i) \subseteq \mathcal{N}_{p_j}(v_i)$. We give all graphs the full node set found at the lowest persistence level of the hierarchy, but nodes that would not otherwise exist in the graph at a higher persistence level are disconnected.

3. FRAMEWORK FOR LEARNING

3.1. Feature Construction and Labeling

We construct features by applying standard transformations to an input image used in a similar previous study [23]. These include the identity of the image; Gaussian blurring, differences of Gaussians with standard deviation $\sigma \in \{2,4,8,16\}$ to serve as a smoothing kernel for capturing pixel neighborhood information; the maximum, minimum, median, and variance of pixel neighborhoods for radius sizes $\{2,4,8,16\}$; Sobel filtration; Gaussian edge detection; and Hessian eigenvalue filtration. Nodes originating from topological priors also afford aggregate pixel statistics for those covered by priors. We compute the median, minimum, maximum, standard deviation, and variance among pixel intensities for these. Altogether, the features computed give us 235-dimensional features per topological prior. If the original pixels have d-dimensional features, the priors are represented by a $5d-$dimensional feature vector resulting in 235 features per topological prior.

We use an interactive tool first proposed in [23] to facilitate fast selection and human labeling of priors graphs. Through the tool, a human expert can directly label topological priors as foreground or background. The tool also allows a user to construct and visualize the MSC persistence hierarchy, interactively select a persistence threshold affording an MSC found to cover the semantic object sufficiently, and subsequently label topological priors from the priors graph derived from the user's chosen MSC with several easy interactions to "paint" label assignments to priors.

3.2. Identifying and Labeling Topological Graph Hierarchies

A higher persistence subgraph can encompass multiple topological primitives from a lower persistence supergraph. In a labeled priors supergraph, a topological primitive in the subgraph receives the majority class label from corresponding parts of the supergraph if the proportion of primitives in the supergraph surpasses a user-defined union threshold. In this way, each prior in the subgraph is given the predominant label of it's corresponding priors in the supergraph. This approach identifies significant priors from the supergraph as components of the subgraph. Formally, for persistence levels p_i and $p_j \geq p_i$, the class ℓ_j of a topological prior v_{p_j} in the priors subgraph of persistence p_j is a subset to priors supergraph of persistence p_i with labeling scheme ℓ_i. Given a union threshold u the label ℓ_j is given as follows: $\ell_j = \ell_i$ if $\frac{1}{|v_{p_i} : v_{p_i} \in v_{p_j}|} \sum_{v_{p_i} \subseteq v_{p_j}} 1(\ell(v_{p_i}) = \ell_i) \geq u$.

3.3. Models Compared

We train a random forest (RF) and multilayer perceptron (MLP) on pixel-level feature statistics, denoted **RF Pixel** and **MLP Pixel**. We also investigate the performance of pixel-level classification using a U-Net [24]. We construct features for each topological prior using the aggregate statistics (median, minimum, maximum, standard deviation, and variance) for pixel intensities of the pixels covered by the geometry of the topological prior. We perform classification in the topological prior space without using the graph structure using a random forest and an MLP, denoted **RF Priors** and **MLP Priors**, and with the priors graph's graph structure and topological priors' features using GraphSAGE [25] as a baseline **GNN** model to learn node representations. We denote our hierarchical GNN methods as **GNN-HJT** and **GNN-HST** for hierarchical joint training and hierarchical successive training, respectively.

4. METHODOLOGY

We consider two methods for learning node representations for classification using GNNs while exploiting the full hierarchy of topological information. The first method modifies the training of GNN over graph hierarchies, and the second modifies the architecture to pass messages jointly between all hierarchical graph levels.

4.1. Hierarchical Successive Training (HST)

For our first method, we use a GNN, which learns node features via message passing in the spatial domain [25]. For the i^{th} persistence level p_i (moving from sparsest to densest), the aggregated node embedding for node v_i from neighbors $u_i \in \mathscr{N}_{p_i}(v_i)$ of persistence subgraph G_i is given by $h^{k_i}_{\mathscr{N}_{p_i}(v_i)} = \sigma\big(AGGR(h^{k_i-1}_{u_i} : \forall\, u_i \in \mathscr{N}_{p_i}(v_i))\big)$. Node v_i's updated embedding is concatenated with the target node embedding from the previous iteration of aggregation $h^{k_i}_{v_i}$ and the aggregated neighbor features, parameterized with target embedding and neighbor embedding weight matrices $W^{k_i}_s$ and $W^{k_i}_n$, respectively. Similarly, the target node embeddings and neighboring node embeddings are passed through a two-layer multilayer perceptron (MLP), which after sum-pooling, are multiplied with weight matrices W_s and W_n to complete the AGGR step.

Instead of training a GNN for N epochs on the finest graph G_P, we train it for $\frac{N}{P}$ epochs each on graphs $G_1,\ldots,G_P$, using the embeddings from G_{i-1} to initialize the embeddings of corresponding nodes in $G_i\ \forall\ i \in [1\ldots,P]$. The MLPs and node embedding weight matrices for target and neighboring node embeddings are shared between graph hierarchies. Training begins with the smallest subset graph and iteratively increases in graph size. We also successively share the learned embedding weight matrices and matrices of the MLPs as well. As a result, when we begin training on graph G_p, the embedding for node $v_p \in \mathscr{V}(G_p)$ is the combined embedding from previous aggregations at lower persistence subgraphs: $h^{k_p}_{v_p} = COMBINE\big(h^{k_{p-1}}_{v_{p-1}}, h^{k_p-1}_{v_p}\big)$, where we implement COMBINE with concatenation.

4.2. Hierarchical Joint Training (HJT)

For node v_i in node set $\mathbf{V}_{p_i}$ of subgraph $\mathbf{G}_{p_i}$, the feature representation $h^k_{v_i}$ at GNN layer $k \in \{1,\cdots,K\}$ first combines self-representations from the previous level of aggregation with aggregated neighbor information:

$$h^k_{v_i} = COMBINE\big(W^{k-1}_s h^{k-1}_v, AGGR\big(\{W^{k-1}_n h^{k-1}_u : u \in \mathscr{N}_{p_i}(v_i)\}\big)\big)$$

The resulting embedding is combined with the embedding representation from other persistence subgraphs. As an example for two persistence levels p_i and p_j where $v_i \in \mathbf{V}_{p_i} \subset \mathbf{G}_{p_i}$ and $v_j \in \mathbf{V}_{p_j} \subset \mathbf{G}_{p_j}$, we have $h^k_v = \sigma\big(COMBINE(h^k_{v_i}, h^k_{v_j})\big)$.

5. RELATED WORK

Related works that consider topologically informed message passing for GNNs [9, 26] are typically presented with a single graph structure a priori as input (either one graph in which to make node-level predictions, or one graph per data point for graph-level prediction tasks). In contrast, we are constructing an ordered set of graphs from a hierarchy of successive topological simplifications of our (image) data. Each topological summary is determined by topological persistence, and the sequence of these summaries is ordered by persistence level, which allows us to obtain a hierarchy of graphs representing different scales of topological information. Our new challenge, then, is to adapt the message passing of graph neural networks to happen across several graphs, whether jointly or successively (we consider and compare both strategies).

Our work has parallels to the exciting research area of multi-level frameworks for graph neural networks [27, 28], which derive a multi-scale hierarchy of graphs from a single input graph using spectral coarsening methods (differing from our topological constructions). Note that their primary goal is to speed up the computation of GNNs by training them on the smallest coarsened graph and projecting the results back to the larger original graph, whereas we perform training at all scales of information.

6. EXPERIMENTAL SETUP

6.1. Datasets

We evaluate our methods on three image datasets used in prior studies [23], which were selected to ensure sufficient variety to explore the general applicability of the methods we introduce in this work. For each image, a functional operator is first applied. This preprocessing step allows for a topologically informative scalar field representation of the image, or *scalar field image*, which optimizes the expressivity of the MSC summary and allows the corresponding topological priors to cover the target semantic object best.

The datasets chosen were *Retinal* segmentation of blood vessel arbors (biomedicine) [29], cell boundary segmentation from computed tomography imaging of closed-cell *Foam* (materials science) [30], and segmentation of projected sparse *Neuron* imaging (neuroscience) [31]. For preprocessing and the construction of scalar field images, we follow as is described in [23], which also includes justification in the scalar field image design choice.

As a basic demonstration of using a topological hierarchy, we use two levels of persistence (resulting in one subgraph and one supergraph). We chose persistence levels per dataset such that the subgraph was sufficiently smaller in size and sufficiently covered the semantic object. We give more detailed summary statistics and descriptions of the datasets used for evaluation in Table 1.

Name	Retinal	Foam	Neuron
Image Shape	700×605	828×846	$1,737 \times 1,785$
$\lvert\mathbf{V}\rvert$ **Sub/Sup**	676 / 24,851	1,703 / 8,069	323 / 23,038
$\lvert\mathbf{E}\rvert$ **Sub/Sup**	1,097 / 39,765	2,429 / 13,234	480 / 34,527
p_i **Sub/Sup**	0.8 / 0.01	80 / 20	45 / 11
Total Length	269,036	142,895	425,441
%Fgrnd.	11.6%	69.9%	15.5%

Table 1: Dataset and Priors Graph Statistics.

Homophily and Class Balance: We report the homophily and class balance ratios for the derived priors graphs in Table 2. The class balance ratio is defined as the total positively labeled foreground over the total negatively labeled background nodes. For nodes u and v with labels y_u and y_v and edge set $\mathscr{E}_i$ for graph G_i in the persistence hierarchy, the homophily ratio [32] is given by $h_i = \frac{1}{|\mathscr{E}_i|} \sum_{v \in \mathscr{E}_i} \frac{|\{u \in \mathscr{N}(v) : y_u = y_v\}|}{|\mathscr{N}(v)|}$.

Data	Homophily Ratio		Class Ratio	
	Subgraph	Full Graph	Subgraph	Full Graph
Retinal	0.39	0.91	0.55	0.16
Foam	0.38	0.76	5.12	1.93
Neuron	0.35	0.86	1.32	0.28

Table 2: Homophily and class balance ratio for sub- and supergraphs.

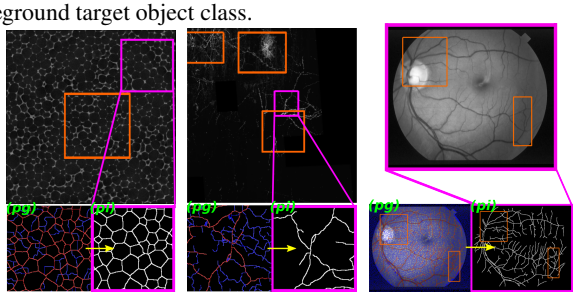
Intuitively, we expect the class imbalance to be larger in subgraphs due to being associated with a sparser persistence subgraph since the ridge/valley graphs these subgraphs originate from span larger regions of connectivity in the image space, which are more likely, but with less accuracy, to be associated to the ridge lines of the semantic object. Similarly, we anticipate the homophily ratio to be lower for lower level persistence graphs since, being sparser and of low granularity, topological priors cover and are adjacent to more expansive but less accurate topological 1-cells.

6.2. Metrics

We use the class F_1 scores of all pixel- and priors-based models for comparison. We perform a parameter sweep for each run over each model's inference foreground/background probability threshold to maximize the class F_1 score.

6.3. Training and Inference Procedure

For training, we chose subregions that accurately capture the diversity of geometric information and variability within each dataset, starting with size 64 $\times$ 64. We then grew the training boxes by approximately 10% of the image until terminating once the percentage of training exceeded more than 60% of the image. Figure 2 illustrates an instance of the training subset regions from all datasets, as was used by all models during our study. Topological priors belonging to the priors graph entirely within the regions of intersecting tiles of size 64 $\times$ 64 were used for training, and the remainder for inference probabilities were assigned to all priors as belonging to the foreground target object class.



(a) Foam (b) Neuron (c) Retinal

Fig. 2: A summary of each dataset with the subset region(s) (highlighted in orange) used to train all models for the experimental run at which the accuracy results of all models begin to plateau, as seen in Figure 3. Highlighted in pink is the enlarged region demonstrating the priors graph (pg) and pixel-level (pi) ground truths.

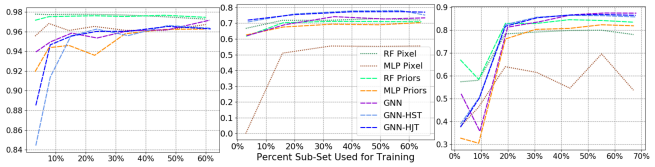
6.4. Hyperparameters and Computing Environment

We performed a parameter sweep of learning rates $\{1,2,3\} \times \{1e^{-2}, 1e^{-3}, 1e^{-4}\}$, finding $3e^{-3}$ to be the best. We trained each model for 10 epochs following the original GraphSAGE paper [25] and used a weight decay of $1e^{-7}$. For each variant of GNN, we used four layers and hidden vertex embedding dimensions of 512 and 1024 with output vertex embedding of 256, and aggregate neighbors' embeddings by maximum pooling. We also added jumping knowledge between GNN layers [33], which has been found helpful when class heterophily is high, as in our datasets [32]. All experiments were run on a laptop with 3GB GeForce GTX 970M with 1280 CUDA Cores GPU and 3.5GHz i7-6700HQ processor running Ubuntu, Linux.

7. EXPERIMENTAL RESULTS

Figure 3 shows that our hierarchical approaches generally improve over the conventional GNN and non-GNN baselines, particularly for the Neuron and Retinal datasets, where there is a larger difference between the topological summary of the computed subgraphs and supergraphs. This difference can be seen by comparing the respective number of nodes and edges shown in Table 1.

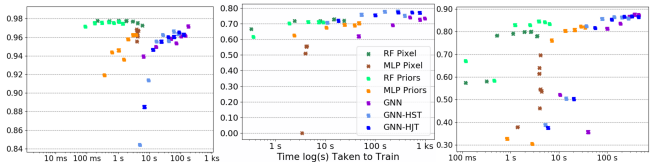
In Table 3, we provide empirical measurements of the time required to compute the topological structures needed for our methods. Compared to traditional image segmentation approaches operating in pixel space, our methods result in appreciable speed-up even when accounting for the overhead of the topological data analysis. In Figure 4, we demonstrate the comparatively low time and effort



(a) Foam (b) Neuron (c) Retinal

Fig. 3: Class F_1 vs. percent priors graph used in training. We see improvements from hierarchical training on the Neuron and Retinal datasets with lower performance on Foam.

expected of a user to achieve a given segmentation accuracy (also shown in Figure 3), with our methods showing competitive results to that of contemporary pixel-based approaches, and especially in comparison to U-Net, which takes considerably longer to train. Moreover, of the GNN approaches, the hierarchical approaches are the fastest, notably GNN-HST, which performs a portion of the training on smaller graphs in the hierarchy as opposed to traditional methods, which train exclusively on the larger supergraph.



(a) Foam (b) Neuron (c) Retinal

Fig. 4: Class F_1 vs. time taken to train priors graph used in training. Times and accuracies correspond to training region size and accuracy as provided in Figure 3. Among the GNN approaches, the fastest are the hierarchical methods and notably GNN-HST.

Dataset	Scalar Field Image	Hierarchical MSC	Prior Features
Retinal	16.39	3.65	6.11
Foam	27.05	3.56	6.52
Neuron	100.17	9.54	7.23

Table 3: Computational times (secs.), explained by column, for each (1) dataset, associated scalar field image to improve MSC coverage (2), the multilevel MSC hierarchy (3), and feature vectors for all topological priors (4).

Our GNN-HST variation also enjoys runtime gains compared to the conventional GNN using the full supergraph for all of training, due to the successive training scheme allowing us to perform a portion of training on the lower complexity smaller subgraph. Our joint training yields slightly higher accuracy as the model simultaneously uses multiple scales of topological information, and the runtime remains manageable. Although non-network baselines, using famously fast ML models such as random forest and few-layer MLPs, are faster, we have seen that they generally are less accurate.

Limitations: Hierarchical GNN training was less effective on the Foam dataset, which had fewer differences between subgraph and supergraph. On this dataset, methods performing learning in the pixel space achieved higher performance than learning with topological priors, which may be due to the sparsity of the priors graphs and high class imbalance, as shown in Table 2. Future work should explore how well topological priors model semantic structure, how to construct informative topological information, including scalar field images, and the impact of homophily and class balance.

8. CONCLUSION

We propose graph neural network methods that learn from hierarchies of graphs representing the multiscale topological structure of image data. Our results demonstrate promise for increased accuracy and improvement in training time compared to conventional GNNs, and we provide exploratory insights into the effect of class imbalance and heterophily in graph learning.

9. REFERENCES

[1] Torin McDonald, Will Usher, Nate Morrical, Attila Gyulassy, Steve Petruzza, Frederick Federer, Alessandra Angelucci, and Valerio Pascucci, "Improving the usability of virtual reality neuron tracing with topological elements," *IEEE TVCG*, vol. 27, no. 2, pp. 744–754, 2020.

[2] Steve Petruzza, Attila Gyulassy, Samuel Leventhal, John J Baglino, Michael Czabaj, Ashley D Spear, and Valerio Pascucci, "High-throughput feature extraction for measuring attributes of deforming open-cell foams," *IEEE TVCG*, vol. 26, no. 1, pp. 140–150, 2019.

[3] Hongjuan Xue and Yaolin Gu, "Application of topological analysis in ocean feature extraction," *Computer Engineering*, vol. 35, no. 3, pp. 263, 2009.

[4] Valerio Pascucci, Xavier Tricoche, Hans Hagen, and Julien Tierny, *Topological Methods in Data Analysis and Visualization: Theory, Algorithms, and Applications*, Springer Science & Business Media, 2010.

[5] Peer-Timo Bremer, W Cabot, A Cook, D Laney, A Mascarenhas, P Miller, and V Pascucci, "Understanding the structure of the turbulent mixing layer in hydrodynamic instabilities," in *Journal of Physics: Conference Series*. IOP Publishing, 2006, vol. 46, p. 077.

[6] Samik Banerjee, Lucas Magee, Dingkang Wang, Xu Li, Bing-Xing Huo, Jaikishan Jayakumar, Katherine Matho, Meng-Kuan Lin, Keerthi Ram, Mohanasankar Sivaprakasam, et al., "Semantic segmentation of microscopic neuroanatomical data by combining topological priors with encoder–decoder deep networks," *Nature Machine Intelligence*, vol. 2, no. 10, pp. 585–594, 2020.

[7] Xiaoling Hu, Yusu Wang, Li Fuxin, Dimitris Samaras, and Chao Chen, "Topology-aware segmentation using discrete Morse theory," *arXiv preprint arXiv:2103.09992*, 2021.

[8] Michael Moor, Max Horn, Bastian Rieck, and Karsten Borgwardt, "Topological autoencoders," in *International conference on machine learning*. PMLR, 2020, pp. 7045–7054.

[9] Qi Zhao, Ze Ye, Chao Chen, and Yusu Wang, "Persistence enhanced graph neural network," in *AISTATS*. PMLR, 2020, pp. 2896–2906.

[10] Robin Forman, "A user's guide to discrete Morse theory," *Sém. Lothar. Combin*, vol. 48, pp. 35pp, 2002.

[11] Attila Gyulassy, "MSCEER: Morse-Smale complex extraction, exploration, reasoning," 2018.

[12] Herbert Edelsbrunner, David Letscher, and Afra Zomorodian, "Topological persistence and simplification," in *FOCS*. IEEE, 2000, pp. 454–463.

[13] P.-T. Bremer, H. Edelsbrunner, B. Hamann, and V. Pascucci, "A multi-resolution data structure for two-dimensional Morse-Smale functions," in *IEEE Visualization, 2003. VIS 2003.*, 2003, pp. 139–146.

[14] Attila Gyulassy, Vijay Natarajan, Valerio Pascucci, Peer-Timo Bremer, and Bernd Hamann, "A topological approach to simplification of three-dimensional scalar functions," *IEEE TVCG*, vol. 12, no. 4, pp. 474–484, 2006.

[15] Pinaki Pratim Acharjya and Dibyendu Ghoshal, "Watershed segmentation based on distance transform and edge detection techniques," *IJCA*, vol. 52, no. 13, 2012.

[16] Serge Beucher, "Watersheds of functions and picture segmentation," in *ICASSP*. IEEE, 1982, vol. 7, pp. 1928–1931.

[17] Jos BTM Roerdink and Arnold Meijster, "The watershed transform: Definitions, algorithms and parallelization strategies," *Fundamenta informaticae*, vol. 41, no. 1-2, pp. 187–228, 2000.

[18] Milan Sonka, Vaclav Hlavac, and Roger Boyle, *Image processing, analysis, and machine vision*, Cengage Learning, 2014.

[19] Paul Bendich, David Cohen-Steiner, Herbert Edelsbrunner, John Harer, and Dmitriy Morozov, "Inferring local homology from sampled stratified spaces," in *FOCS*. IEEE, 2007, pp. 536–546.

[20] Herbert Edelsbrunner and John Harer, *Computational topology: an introduction*, American Mathematical Soc., 2010.

[21] Zhengdao Chen, Lisha Li, and Joan Bruna, "Supervised community detection with line graph neural networks," in *ICLR*, 2019.

[22] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec, "Hierarchical graph representation learning with differentiable pooling," *NeurIPS*, 2018.

[23] Samuel Leventhal, Attila Gyulassy, Mark Heimann, and Valerio Pascucci, "Exploring classification of topological priors with machine learning for feature extraction," *IEEE TVCG*, 2023.

[24] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[25] William L Hamilton, Rex Ying, and Jure Leskovec, "Inductive representation learning on large graphs," in *NeurIPS*, 2017, pp. 1025–1035.

[26] Max Horn, Edward De Brouwer, Michael Moor, Yves Moreau, Bastian Rieck, and Karsten Borgwardt, "Topological graph neural networks," in *ICLR*, 2022.

[27] Chenhui Deng, Zhiqiang Zhao, Yongyu Wang, Zhiru Zhang, and Zhuo Feng, "Graphzoom: A multi-level spectral approach for accurate and scalable graph embedding," in *ICLR*, 2020.

[28] Zengfeng Huang, Shengzhong Zhang, Chong Xi, Tang Liu, and Min Zhou, "Scaling up graph neural networks via graph coarsening," in *KDD*, 2021.

[29] EL Van der Merwe and SH Kidson, "Advances in imaging the blood and aqueous vessels of the ocular limbus," *Experimental eye research*, vol. 91, no. 2, pp. 118–126, 2010.

[30] Yoav Lahini, Omer Gottesman, Ariel Amir, and Shmuel M Rubinstein, "Nonmonotonic aging and memory retention in disordered mechanical systems," *Physical review letters*, vol. 118, no. 8, pp. 085501, 2017.

[31] Todd A Gillette, Kerry M Brown, and Giorgio A Ascoli, "The diadem metric: comparing multiple reconstructions of the same neuron," *Neuroinformatics*, vol. 9, pp. 233–245, 2011.

[32] Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra, "Beyond homophily in graph neural networks: Current limitations and effective designs," in *NeurIPS*, 2020.

[33] Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka, "Representation learning on graphs with jumping knowledge networks," in *ICML*. 2018, pp. 5449–5458, PMLR.