Topological Pooling on Graphs

Yuzhou Chen[1], **Yulia R. Gel**[2,3]

[1]Department of Computer and Information Sciences, Temple University
[2]Department of Mathematical Sciences, University of Texas at Dallas
[3]National Science Foundation
yuzhou.chen@temple.edu, ygl@utdallas.edu

Abstract

Graph neural networks (GNNs) have demonstrated a significant success in various graph learning tasks, from graph classification to anomaly detection. There recently has emerged a number of approaches adopting a graph pooling operation within GNNs, with a goal to preserve graph attributive and structural features during the graph representation learning. However, most existing graph pooling operations suffer from the limitations of relying on node-wise neighbor weighting and embedding, which leads to insufficient encoding of rich topological structures and node attributes exhibited by real-world networks. By invoking the machinery of persistent homology and the concept of landmarks, we propose a novel topological pooling layer and witness complex-based topological embedding mechanism that allow us to systematically integrate hidden topological information at both local and global levels. Specifically, we design new learnable local and global topological representations Wit-TopoPool which allow us to simultaneously extract rich discriminative topological information from graphs. Experiments on 11 diverse benchmark datasets against 18 baseline models in conjunction with graph classification tasks indicate that Wit-TopoPool significantly outperforms all competitors across all datasets.

Introduction

Graph neural networks (GNNs) have emerged as a powerful machinery for various graph learning tasks, such as link prediction, node and graph classification (Zhou et al. 2020; Xia et al. 2021). In case of graph classification tasks, both graph-level and local-level representation learning play a critical role for GNNs success. Since pooling operators have shown an important role in image recognition and 3D shape analysis (Boureau, Ponce, and LeCun 2010; Shen et al. 2018), it appeared natural to expand the idea of pooling to graphs (Yu and Koltun 2016; Defferrard, Bresson, and Vandergheynst 2016). The earlier pooling techniques within GNN architectures achieved promising results but failed to capture graph structural information and to learn hidden node feature representations. These limitations have stimulated development of new families of graph pooling which simultaneously account for both the structural properties of graphs and node feature representations. Some of the most recent results on graph pooling in this direction include integration of (self)-attention mechanisms (Huang et al. 2019; Lee, Lee, and Kang 2019), advanced clustering approaches (Bianchi, Grattarola, and Alippi 2020; Wang et al. 2020; Bodnar, Cangea, and Liò 2021), and hierarchical graph representation learning (Yang et al. 2021). Nevertheless, these existing pooling techniques still remain limited in their ability to simultaneously extract and systematically integrate intrinsic structural information on the graph, including its node feature representations, at both local and global levels.

We address these limitations by introducing the concepts of shape, landmarks, and witnesses to graph pooling. In particular, our key idea is based on the two interlinked notions from computational topology: persistent homology and witness complexes, neither of which has ever been considered in conjunction with graph pooling. First, we propose a local topological score which measures each node importance based on how valuable shape information of its local vicinity is and which then results in a topological pooling layer. Second, inspired by the notion of landmarks in computer vision, we learn the most intrinsic global topological properties of the graph, using only a subset of the most representative nodes (or landmarks). The remaining nodes act as witnesses that govern which higher-order graph structures (or simplices) are to be included into the learning process. In computational topology, this approach is associated with a witness complex which enjoys competitive computational costs. The resulting new model Wit-TopoPool exhibits capabilities to learn rich discriminative topological characteristics of the graph as well as to extract essential information from node features. Significance of our contributions are the following:

- We propose a novel topological perspective to graph pooling by introducing the concepts of persistence, landmarks, and witness complexes.
- We develop a new model Wit-TopoPool which systematically and simultaneously integrates the most essential discriminative topological characteristics of the graph, including its node feature representations, at both local and global levels.
- We validate Wit-TopoPool in conjunction with graph clas-

sification tasks versus 18 state-of-the-art competitors on 11 diverse benchmark datasets from chemistry, bioinformatics, and social sciences. Our experiments indicate that Wit-TopoPool delivers the most competitive performance under all considered scenarios.

Related Work

Graph Pooling and Graph Neural Networks In the last few years GNNs have proven to become the primary machinery for graph classification tasks (Zhou et al. 2020; Xia et al. 2021). Inspired by the success of GNN-based models for graph representation learning, there has appeared a number of approaches introducing a pooling mechanism into GNNs which addresses the limitations of traditional graph pooling architectures, i.e., a limited capability to capture the graph substructure information. For instance, DiffPool (Ying et al. 2018) develops a differential pooling operator that learns a soft assignment at each graph convolutional layer. The similar idea is utilized in EigenGCN (Ma et al. 2019), which introduces a pooling operator based on the graph Fourier transform. Top-K pooling operations (Cangea et al. 2018; Gao and Ji 2019) design a pooling method by using node features and local structural information to propagate only the top-K nodes with the highest scores at each pooling step. Self-Attention Graph Pooling (SAGPool) (Lee, Lee, and Kang 2019) leverages self-attention mechanism based on GNN to learn the node scores and select the nodes by sorting their scores. Although these GNNs and graph pooling operations have achieved state-of-the-art performance in graph classification tasks, a common limitation among all aforementioned approaches is that they cannot accurately capture higher-order properties of graphs and incorporate this topological information into neural networks. Different from these existing methods, we propose a novel model Wit-TopoPool that not only adaptively captures local topological information in the hidden representations of nodes but yields a faster and scalable approximation of the simplicial representation of the global topological information.

Witness Complexes, Landmarks, and Topological Graph Learning Persistent homology (PH) is a suite of tools within topological data analysis (TDA) that has shown substantial promise in a broad range of domains, from bioinformatics to material science to social networks (Otter et al. 2017; Carlsson 2020). PH has been also successfully integrated as a fully trainable topological layer into various DL models, for such graph learning tasks as node and graph classification, link prediction and anomaly detection (see, e.g., overviews Carlsson (2020); Tauzin et al. (2021). In most applications, PH is based on a Vietoris-Rips ($\mathcal{VR}$) complex which enjoys a number of important theoretical properties on approximation of the underlying topological space. However, $\mathcal{VR}$ does not scale efficiently to larger datasets and, in general, computational complexity remains one of the primary roadblocks on the way of the wider applicability of PH. A promising but virtually unexplored alternative to $\mathcal{VR}$ is a witness complex (De Silva and Carlsson 2004) which recovers the topology of the underlying space using only a subset of points (or landmarks), thereby, substantially reducing computational complexity and potentially allowing us to focus only on the topological information delivered by the most representative points. Nevertheless, applications of witness complex in machine learning are yet nascent (Schönenberger et al. 2020; Poklukar, Varava, and Kragic 2021). Here we harness strengths of witness complex and graph landmarks to accurately and efficiently learn topological knowledge representations of graphs.

Methodology

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{X})$ be an attributed graph, where $\mathcal{V}$ is a set of nodes ($|\mathcal{V}| = N$), $\mathcal{E}$ is a set of edges, and $\boldsymbol{X} \in \mathbb{R}^{N\times F}$ is a node feature matrix (here F is the dimension of node features). Let d_{uv} be the distance on $\mathcal{G}$ defined as the shortest path between nodes u and v, $u, v \in \mathcal{V}$, and $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ be a symmetric adjacency matrix such that $\boldsymbol{A}_{uv} = \omega_{uv}$ if nodes u and v are connected and 0, otherwise (here ω_{uv} is an edge weight and $\omega_{uv} \equiv 1$ for unweighted graphs). Furthermore, $\boldsymbol{D}$ represents the degree matrix with $\boldsymbol{D}_{uu} = \sum_v \boldsymbol{A}_{uv}$, corresponding to $\boldsymbol{A}$.

Definition 1 (**k-hop Neighborhood based on Graph Structure**) *An induced subgraph $\mathcal{G}_u^k = (\mathcal{V}_u^k, \mathcal{E}_u^k) \subseteq \mathcal{G}$ is called a k-hop neighborhood of node $u \in \mathcal{V}$ if for any $v \in \mathcal{V}_u^k$, $d_{uv} \leq k$.*

Preliminaries on Persistent Homology and Witness Complexes

PH is a subfield in computational topology which allows us to retrieve evolution of the inherent shape patterns in the data along various user-selected geometric dimensions (Edelsbrunner, Letscher, and Zomorodian 2000; Zomorodian and Carlsson 2005). Broadly speaking, by "shape" here we mean the properties of the observed object which are preserved under continuous transformations, e.g., stretching, bending, and twisting. (The data can be a graph, a point cloud in Euclidean space, or a sample of points from any metric space). Since one of the most popular PH techniques is to convert the point cloud to a distance graph, for generality we proceed with the further description of PH on graph-structured data. By using a multi-scale approach to shape description, PH enables to address the intrinsic limitations of classical homology and to extract the shape characteristics which play an essential role in a given learning task. In brief, the key idea is to choose some suitable scale parameters α and then to study changes in homology that occur to $\mathcal{G}$ which evolves with respect to α. That is, we no longer study $\mathcal{G}$ as a single object but as a *filtration* $\mathcal{G}_{\alpha_1} \subseteq \ldots \subseteq \mathcal{G}_{\alpha_n} = \mathcal{G}$, induced by monotonic changes of α. To make the process of pattern counting more systematic and efficient, we build an abstract simplicial complex $\mathscr{K}(\mathcal{G}_{\alpha_j})$ on each $\mathcal{G}_{\alpha_j}$, resulting in a filtration of complexes $\mathscr{K}(\mathcal{G}_{\alpha_1}) \subseteq \ldots \subseteq \mathscr{K}(\mathcal{G}_{\alpha_n})$. For instance, we can select a scale parameter as a shortest (weighted) path between any two nodes; then abstract simplicial complex $\mathscr{K}(\mathcal{G}_{\alpha_*})$ is generated by subgraphs $\mathcal{G}^{'}$ of bounded diameter α_* (that is, $(k-1)$-simplex in $\mathscr{K}(\mathcal{G}_{\alpha_*})$ is made up by subgraphs $\mathcal{G}^{'}$ of k-nodes with $diam(\mathcal{G}^{'}) \leq \alpha_*$). If $\mathcal{G}$ is an edge-weighted graph $(\mathcal{V}, \mathcal{E}, w)$, with the edge-weight function $w : \mathcal{E} \mapsto \mathbb{R}$, then for each α_j we can consider only induced subgraphs of $\mathcal{G}$ with maximal degree of α_j, resulting

in a degree sublevel set filtration. (For the detailed discussion on graph filtrations see Hofer et al. (2020).)

Equipped with this construction, we trace data shape patterns such as independent components, holes, and cavities which appear and merge as scale α changes (i.e., for each topological feature ρ we record the indices b_ρ and d_ρ of $\mathscr{K}(\mathcal{G}_{b_\rho})$ and $\mathscr{K}(\mathcal{G}_{d_\rho})$, where ρ is first and last observed, respectively). We say that a pair (b_ρ, d_ρ) represents the birth and death times of ρ, and $(d_\rho - b_\rho)$ is its corresponding lifespan (or persistence). In general, topological features with longer lifespans are considered valuable, while features with shorter lifespans are often associated with topological noise. The extracted topological information over the filtration $\{\mathscr{K}_{\alpha_j}\}$ can be then summarized as a multiset in $\mathbb{R}^2$ called *persistence diagram (PD)* $\mathrm{Dg} = \{(b_\rho, d_\rho) \in \mathbb{R}^2 : d_\rho > b_\rho\} \cup \Delta$ (here $\Delta = \{(t,t)|t \in \mathbb{R}\}$ is the diagonal set containing points counted with infinite multiplicity; including Δ allows us to compare different PDs based on the cost of the optimal matching between their points).

Finally, there are multiple options to select an abstract simplicial complex $\mathscr{K}$ (Carlsson and Vejdemo-Johansson 2021). Due to its computational benefits, one of the most widely adopted choices is a Vietoris-Rips ($\mathcal{VR}$) complex. However, the $\mathcal{VR}$ complex uses the entire observed data to describe the underlying topological space and, hence, does not efficiently scale to large datasets and noisy datasets. In contrast, a witness complex captures the shape structure of the data based only on a significantly smaller subset $\mathfrak{L} \subseteq \mathcal{V}$, called a set of *landmark* points. In turn, all other points in $\mathcal{V}$ are used as "witnesses" that govern which simplices occur in the witness complex.

Definition 2 (Weak Witness Complex) *We call $w \in \mathcal{V}$ to be a weak witness for a simplex $\sigma = [v_0 v_1 \dots v_l]$, where $v_i \in \mathcal{V}$ for $i = 0, 1, \dots, l$ and nonnegative integer l, with respect to $\mathfrak{L}$ if and only if $d_{wv} \leq d_{wu}$ for all $v \in \sigma$ and $u \in \mathfrak{L} \setminus \sigma$. The weak witness complex $\mathcal{W}(\mathfrak{L}, \mathcal{G})$ of the graph $\mathcal{G}$ with respect to $\mathfrak{L}$ has a node set formed by the landmark points in $\mathfrak{L}$, and a subset σ of $\mathfrak{L}$ is in $\mathcal{W}(\mathfrak{L}, \mathcal{G})$ if and only if there exists a corresponding weak witness in the graph $\mathcal{G}$.*

Wit-TopoPool: The Proposed Model

We now introduce our Wit-TopoPool model which leverages two new topological concepts into graph learning, topological pooling and witness complex-based topological knowledge representation. The core idea of Wit-TopoPool is to simultaneously capture the discriminating topological information of the attributed graph $\mathcal{G}$, including its node feature representation, at both local and global levels. The first module of topological pooling assigns each node a topological score, based on measuring how valuable shape information of its local neighborhood is. In turn, the second module of witness complex-based topological knowledge representation allows us to learn the inherent global shape information of $\mathcal{G}$, by focusing only on the most essential landmarks of $\mathcal{G}$. The Wit-TopoPool architecture is illustrated in Figure 1.

GNN-based Topological Pooling Layer The ultimate goal of this module is to sort nodes based on the importance of topological information, exhibited by their local neighborhoods at different learning stages (i.e., different layers of GNN). That is, a node neighborhood can be defined either based on the connectivity of the observed graph $\mathcal{G}$ or based on the similarity among node embeddings. This allows us to adaptively learn hidden higher-order structural dependencies among nodes that originally may not be within the proximity to each other with respect to the graph distance. To extract such local topological node representation, we first learn a latent node embedding by aggregating information from its the k-hop neighbors ($k \geq 1$) through graph convolutions

$$\boldsymbol{H}^{(\ell+1)} = \sigma((\tilde{\boldsymbol{D}}^{-\frac{1}{2}}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{\frac{1}{2}})^k \boldsymbol{H}^{(\ell)}\boldsymbol{W}^{(\ell)}). \quad (1)$$

Here $\sigma(\cdot)$ is the non-linear activation function (e.g., $\mathrm{ReLU}(0,x) = \max(x)$), $\boldsymbol{W}^{(\ell)} \in \mathbb{R}^{d'_c \times d_c}$ is trainable weight of (ℓ)-th layer (where d'_c is the dimension of the $(\ell-1)$-th layer's output), $\boldsymbol{H}^{(0)} = \boldsymbol{X}$, $\boldsymbol{H}^{(\ell+1)} \in \mathbb{R}^{N \times d_c}$, and k-th power operator contains statistics from the k-th step of a random walk on the graph. Equipped with above node embedding, in order to capture the underlying structural information of nodes in latent feature space, we can measure the similarity between nodes. More specifically, for each node u, following Definition (3), we obtain the ϕ-distance neighborhood subgraph $\boldsymbol{Z}^\phi_u$.

Definition 3 (ϕ-distance Neighborhood of Node Embedding) *Let $\boldsymbol{H}^{(\ell+1)}$ be the node embedding of (ℓ)-th layer of GNN. For any $u, v \in \mathcal{V}$, we can calculate the similarity score $\boldsymbol{Z}_{uv}$ between nodes u and v as (i) Cosine Similarity: $\boldsymbol{Z}_{uv} = \frac{\boldsymbol{H}^{(\ell+1)}_u \cdot \boldsymbol{H}^{(\ell+1)}_v}{||\boldsymbol{H}^{(\ell+1)}_u|| ||\boldsymbol{H}^{(\ell+1)}_v||}$ or (ii) Gaussian Kernel: $\boldsymbol{Z}_{uv} = \exp(-\gamma||\boldsymbol{H}^{(\ell+1)}_u - \boldsymbol{H}^{(\ell+1)}_v||^2)$ (where γ is a free parameter). Given the pre-defined threshold $\phi > 0$, we have a ϕ-distance neighborhood subgraph $\boldsymbol{Z}^\phi_u = (\mathcal{V}^\phi_u, \mathcal{E}^\phi_u)$, i.e., for any $v \in \mathcal{V}^\phi_u$, $\boldsymbol{Z}_{uv} \geq \phi$ and for any $v, w \in \mathcal{V}^\phi_u$ with $e_{vw} \in \mathcal{E}^\phi_u$, the similarity score between nodes v and w is larger than or equal to ϕ, i.e., $\boldsymbol{Z}_{vw} \geq \phi$.*

Considering the ϕ-distance neighborhood subgraph $\boldsymbol{Z}^\phi_u$ allows us to capture some hidden similarity between unconnected yet relevant nodes, which in turn plays an important role for revealing high-order structural characteristics of the graph. The intuition behind this idea is the following. Suppose that there are two people on a social media network whose graph distance in-between is high, but these people share some joint interests (node attributes). Then, while these two people are not neighbors in terms of Definition (1), they might end up as neighbors in terms of Definition (3).

We now derive a topological score of each node u in terms of the importance of the topological information yielded by its new surrounding neighborhood. For each $u \in \mathcal{V}$, we first obtain its PD $\mathcal{D}_u = \mathrm{Dg}(\mathcal{VR}(\boldsymbol{Z}^\phi_u))$. Then given the intuition that the longer the persistence of ρ is, the more important the feature ρ is, the *topological score* of node u is defined via the persistence of its topological features in $\mathcal{D}_u$

$$y_u = \begin{cases} \sum_{\rho \in \mathcal{D}_u} (d_\rho - b_\rho) & (i) \\ \sum_{\rho \in \mathcal{D}_u} \arctan\left(C \times ((d_\rho - b_\rho)^\eta\right) & (ii) \end{cases}. \quad (2)$$

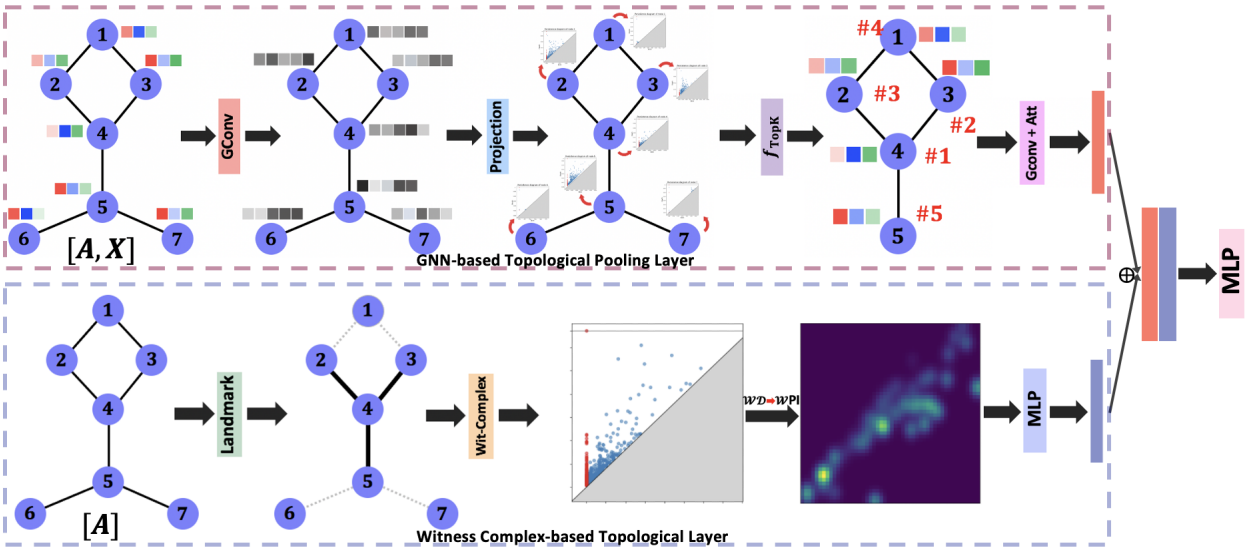

Figure 1: The overall architecture of Wit-TopoPool (for more details see Appendix A).

Here (i) is an unweighted function and (ii) is a piecewise-linear weighted function, C is a non-negative parameter, $\eta \geq 1$ defines a polynomial function, and $\arctan(\cdot)$ is a bounded and continuous function which is vital to guarantee the stability of the vectorization of the persistence diagram. (In Section "Sensitivity Analysis", we discuss experiments assessing the impact of different topological score functions on graph classification tasks.) Finally, we sort all nodes in terms of their topological scores $\boldsymbol{y} = [y_1, y_2, \ldots, y_N]$ and select the top-$\lceil \tau N \rceil$ (where $\lceil \cdot \rceil$ is the operation of rounding up) nodes

$$\text{idx} = f_{\text{TopK}}(\boldsymbol{y}, \lceil \tau N \rceil), \tag{3}$$

where $f_{\text{TopK}}(\cdot)$ is a sorting function and produces indexes of k nodes with largest topological scores. By taking the indices of above $\lceil \tau N \rceil$ nodes, the coarsened (pooled) adjacency matrix $\boldsymbol{A}_{\text{pool}} \in \mathbb{R}^{\lceil \tau N \rceil \times \lceil \tau N \rceil}$, indexed feature matrix $\boldsymbol{X}_{\text{pool}} \in \mathbb{R}^{\lceil \tau N \rceil \times F}$, and the topological pooling enhanced graph convolutional layer (TPGCL) can be formulated as

$$\boldsymbol{A}_{\text{pool}} = \boldsymbol{A}[\text{idx}, \text{idx}], \boldsymbol{X}_{\text{pool}} = \boldsymbol{X}[\text{idx}, \text{idx}], \tag{4}$$

$$\boldsymbol{H}_r = \sigma((\tilde{\boldsymbol{D}}_{\text{pool}}^{-\frac{1}{2}} \tilde{\boldsymbol{A}}_{\text{pool}} \tilde{\boldsymbol{D}}_{\text{pool}}^{\frac{1}{2}}) \boldsymbol{X}_{\text{pool}} \boldsymbol{W}_{\text{pool}}), \tag{5}$$

where $\tilde{\boldsymbol{A}}_{\text{pool}} = \boldsymbol{A}_{\text{pool}} + \boldsymbol{I}$ is the adjacency matrix with self-loop added to each node, $\tilde{\boldsymbol{D}}_{\text{pool}}$ a diagonal matrix where $\tilde{\boldsymbol{D}}_{\text{pool},uu} = \sum_v \tilde{\boldsymbol{A}}_{\text{pool},uv}$, $\boldsymbol{W}_{\text{pool}} \in \mathbb{R}^{F \times d_{\text{pool}}}$ is the learnable weight matrix in the topological pooling enhanced graph convolution operation (where d_{pool} is the output dimension), and $\boldsymbol{H}_r \in \mathbb{R}^{\lceil \tau N \rceil \times d_{\text{pool}}}$ is the output of TPGCL.

Furthermore, to capture second-order statistics of pooled features and generate a global representation, we apply the attention mechanism (i.e., the second-order attention mechanism of Girdhar and Ramanan (2017)) on the TPGCL embedding $\boldsymbol{H}_r$ as follows

$$\hat{\boldsymbol{H}}_r = \boldsymbol{H}_r^{\top}(\boldsymbol{H}_r \boldsymbol{W}_r), \tag{6}$$

where $\boldsymbol{W}_r \in \mathbb{R}^{d_{\text{pool}} \times 1}$ is a trainable weight matrix and $\hat{\boldsymbol{H}}_r \in \mathbb{R}^{d_{\text{pool}}}$ is the final embedding.

Witness Complex-based Topological Layer Now we turn to learning global shape characteristics of graph $\mathcal{G}$. To enhance computational efficiency and to focus on the most salient topological characteristics of $\mathcal{G}$, thereby mitigating the impact of noisy observations, we propose a global topological representation learning module based on a witness complex $\mathcal{W}$ on a set of landmarks $\mathfrak{L}$. Here we consider the landmark set $\mathfrak{L}$ obtained in one of three ways, i.e., (i) randomly, (ii) node degree centrality, and (iii) node betweenness centrality. Specifically, given a graph $\mathcal{G}$ with the number of nodes N and some user-selected parameter ψ ($\psi \in (0, 1]$), the ψN landmarks can be selected (i) uniformly at random resulting in $\mathfrak{L}_r$; (ii) in the decreasing order of their degree centrality resulting in $\mathfrak{L}_d$; and (iii) in the decreasing order of their betweenness centrality resulting in $\mathfrak{L}_b$. Our goal is to select the most representative landmarks which can enhance the quality of the approximate simplicial representation. To adaptively learn the global topological information and correlations between topological structures and node features, (i) we first calculate the similarity matrix $\boldsymbol{S} \in \mathbb{R}^{N \times N}$ among N nodes based on the node embedding of (ℓ)-th GNN layer $\boldsymbol{H}^{(\ell+1)}$ (see Eq. 1) by using either cosine similarity (i.e., $\boldsymbol{S}_{uv} = \frac{\boldsymbol{H}_u^{(\ell+1)} \cdot \boldsymbol{H}_v^{(\ell+1)}}{||\boldsymbol{H}_u^{(\ell+1)}|| ||\boldsymbol{H}_v^{(\ell+1)}||}$) or Gaussian kernel (i.e., $\boldsymbol{S}_{uv} = \exp(-\gamma ||\boldsymbol{H}_u^{(\ell+1)} - \boldsymbol{H}_v^{(\ell+1)}||^2)$), and then we preserve the connections between top similar pairs of nodes (e.g., $\boldsymbol{S}_{uv} \leq \zeta$, where $\zeta \geq 0$) and hence obtain a new graph structure $\hat{\mathcal{G}} = (\mathcal{V}, \hat{\mathcal{E}})$ (where $\hat{\mathcal{E}}$ depends on similarity matrix $\boldsymbol{S}$ and threshold ζ); (ii) then armed with $\hat{\mathcal{G}}$, we extract persistent topological features and summarize them as persistence diagram $\mathcal{WD}_{\hat{\mathcal{G}}}$ of $\hat{\mathcal{G}}$, i.e., $\mathcal{WD}_{\hat{\mathcal{G}}} = \text{Dg}(\mathcal{W}(\mathfrak{L}, \hat{\mathcal{G}}))$.

To input the global topological information summarized

by $\mathcal{WD}_{\hat{\mathcal{G}}}$ into neural network architecture, we convert $\mathcal{WD}_{\hat{\mathcal{G}}}$ to its finite-dimensional vector representation, i.e., witness complex persistence image of resolution p, i.e., $\mathcal{W}\text{PI}_{\hat{\mathcal{G}}} \in \mathbb{R}^{p\times p}$ (see Appendix A for more details of $\mathcal{W}$PI) and then feed the $\mathcal{W}\text{PI}_{\hat{\mathcal{G}}}$ into multi-layer perceptron (MLP) to learn global topological information for graph embedding

$$\boldsymbol{H}_w = \text{MLP}(\text{Flatten}(\mathcal{W}\text{PI}_{\hat{\mathcal{G}}})) \tag{7}$$

where $\text{Flatten}(\cdot)$ flattens $\mathcal{W}\text{PI}_{\hat{\mathcal{G}}}$ into an p^2-dimensional vector representation and $\boldsymbol{H}_w \in \mathbb{R}^{d_w}$ is the output of witness complex-based topological layer. Finally, we concatenate the outputs of GNN-based topological pooling layer (see Eq. 6) and witness complex-based topological layer (see Eq. 7), and feed the concatenated vector into a single-layer MLP for classification as

$$\boldsymbol{H}_o = \text{MLP}([\hat{\boldsymbol{H}}_r, \boldsymbol{H}_w]),$$

where $[\cdot,\cdot]$ denotes the concatenation of the outputs of two layers and $\boldsymbol{H}_o$ is the final classification score.

Experiments

Datasets and Baselines We validate Wit-TopoPool on graph classification tasks using the following 11 real-world graph datasets (for further details, please refer to Appendix B): (i) 3 chemical compound datasets: MUTAG, BZR, and COX2, where graphs represent chemical compounds, nodes are different atoms, and edges are chemical bonds; (ii) 5 molecular compound datasets: PROTEINS, PTC_MR, PTC_MM, PTC_FM, and PTC_FR, where nodes are secondary structure elements and edge existence between two nodes implies that the nodes are adjacent nodes in an amino acid sequence or three nearest-neighbor interactions; (iii) 2 internet movie databases: IMDB-BINARY (IMDB-B) and IMDB-MULTI (IMDB-M), where nodes are actors/actresses and there is an edge if the two people appear in the same movie, and (iv) 1 Reddit (an online aggregation and discussion website) discussion threads dataset: REDDIT-BINARY (REDDIT-B), where nodes are Reddit users and edges are direct replies in the discussion threads. Each dataset includes multiple graphs of each class, and we aim to classify graph classes. For all graphs, we use different random seeds for 90/10 random training/test split. Furthermore, we perform a one-sided two-sample t-test between the best result and the best performance achieved by the runner-up, where *, **, *** denote significant, statistically significant, highly statistically significant results, respectively. We evaluate the performances of our Wit-TopoPool on 11 graph datasets versus 18 state-of-the-art baselines (including 4 types of approaches): (i) 6 graph kernel-based methods: (1) comprised of the subgraph matching kernel (CSM) (Kriege and Mutzel 2012), (2) Shortest Path Hash Graph Kernel (HGK-SP) (Morris et al. 2016), (3) Weisfeiler–Lehman Hash Graph Kernel (HGK-WL) (Morris et al. 2016), (4) Weisfeiler–Lehman (WL) (Shervashidze et al. 2011), and (5) Weisfeiler-Lehman Optimal Assignment (WL-OA) (Kriege, Giscard, and Wilson 2016); (ii) 3 GNNs: (6) Graph Convolutional Network (GCN) (Kipf and Welling 2017), (7) Graph Isomorphism Network (GIN) (Xu et al. 2018), and (8) Deep Graph Convolutional Neural Network (DGCNN) (Zhang et al. 2018); (iii) 4 topological and simplicial complex-based methods: (9) Neural Networks for Persistence Diagrams (PersLay) (Carrière et al. 2020), (10) Filtration Curves with a Random Forest (FC-V) (O'Bray, Rieck, and Borgwardt 2021), (11) Deep Graph Mapper (MPR) (Bodnar, Cangea, and Liò 2021), and (12) Message Passing Simplicial Networks (SIN) (Bodnar et al. 2021); (iv) 6 graph pooling methods: (13) GNNs with Differentiable Pooling (DiffPool) (Ying et al. 2018), (14) TopKPooling with Graph U-Nets (Top-K) (Gao and Ji 2019), (15) GCNs with Eigen Pooling (EigenGCN) (Ma et al. 2019), (16) Self-attention Graph Pooling (SAGPool) (Lee, Lee, and Kang 2019), (17) Spectral Clustering for Graph Pooling (MinCut-Pool) (Bianchi, Grattarola, and Alippi 2020), and (18) Haar Graph Pooling (HaarPool) (Wang et al. 2020).

Experiment Settings We conduct our experiments on two NVIDIA GeForce RTX 3090 GPU cards with 24GB memory. Wit-TopoPool is trained end-to-end by using Adam optimizer and the optimal trainable weight matrices are trained by minimizing the cross-entropy loss function. The tuning of Wit-TopoPool on each dataset is done via grid hyperparameter configuration search over a fixed set of choices and the same cross-validation setup is used to tune baselines. In our experiments, for all datasets, we set the grid size of $\mathcal{W}$PI to 5×5, and the MLP is of 2 layers where Batchnorm and Dropout with dropout ratio of $p_{drop} \in \{0, 0.1, \ldots, 0.5\}$ applied after the fist layer of MLP. For further details, please refer to Appendix A.2. The source code is available at https://github.com/topologicalpooling/TopologicalPool.git.

Experiment Results

The evaluation results on 11 graph datasets are summarized in Tables 1 and 2. We also conduct ablation studies to assess contributions of the key Wit-TopoPool components. Moreover, we perform sensitivity analysis to examine the impact of different choices for topological score functions and landmark sets. OOM indicates out of memory (from an allocation of 128 GB RAM) and OOT indicates out of time (within 120 hours).

Molecular and Chemical Graphs Table 1 shows the performance comparison among 18 baselines on BZR, COX2, MUTAG, PROTEINS, and four PTC datasets with different carcinogenicities on rodents (i.e., PTC_MR, PTC_MM, PTC_FM, and PTC_FR) for graph classification. Our Wit-TopoPool consistently outperforms baseline models on all 8 datasets. In particular, the average relative gain of Wit-TopoPool over the runner-ups is 5.10%. The results demonstrate the effectiveness of Wit-TopoPool. In terms of baseline models, graph kernels only account for the graph structure information and tend to suffer from higher computational costs. In turn, GNN-based models, e.g., GIN, capture both local graph structures and information of the neighborhood for each node, hence, resulting in improvement over graph kernels. Comparing with GNN-based models, graph pooling methods such as SAGPool and HaarPool utilize the hierarchical structure of the graph and extract important geometric information on the observed graphs. Finally, Per-

Model	BZR	COX2	MUTAG	PROTEINS	PTC_MR	PTC_MM	PTC_FM	PTC_FR
CSM	84.54±0.65	79.78±1.04	87.29±1.25	OOT	58.24±2.44	63.30±1.70	63.80±1.00	65.51±9.82
HGK-SP	81.99±0.30	78.16±0.00	80.90±0.48	74.53±0.35	57.26±1.41	57.52±9.98	52.41±1.79	66.91±1.46
HGK-WL	81.42±0.60	78.16±0.00	75.51±1.34	74.53±0.35	59.90±4.30	67.22±5.98	64.72±1.66	67.90±1.81
WL	86.16±0.97	79.67±1.32	85.75±1.96	73.06±0.47	57.97±0.49	67.28±0.97	64.80±0.85	67.64±0.74
WL-OA	87.43±0.81	81.08±0.89	86.10±1.95	73.50±0.87	62.70±1.40	66.60±1.16	66.28±1.83	67.82±5.03
DGCNN	79.40±1.71	79.85±2.64	85.83±1.66	75.54±0.94	58.59±2.47	62.10±14.09	60.28±6.67	65.43±11.30
GCN	79.34±2.43	76.53±1.82	80.42±2.07	70.31±1.93	62.26±4.80	67.80±4.00	62.39±0.85	69.80±4.40
GIN	85.60±2.00	80.30±5.17	89.39±5.60	76.16±2.76	64.60±7.00	67.18±7.35	64.19±2.43	66.97±6.17
Top-K	79.40±1.20	80.30±4.21	67.61±3.36	69.60±3.50	64.70±6.80	67.51±5.96	65.88±4.26	66.28±3.71
MinCutPool	82.64±5.05	80.07±3.85	79.17±1.64	76.52±2.58	64.16±3.47	N/A	N/A	N/A
DiffPool	83.93±4.41	79.66±2.64	79.22±1.02	73.63±3.60	64.85±4.30	66.00±5.36	63.00±3.40	69.80±4.40
EigenGCN	83.05±6.00	80.16±5.80	79.50±0.66	74.10±3.10	N/A	N/A	N/A	N/A
SAGPool	82.95±4.91	79.45±2.98	76.78±2.12	71.86±0.97	69.41±4.40	66.67±8.57	67.65±3.72	65.71±10.69
HaarPool	83.95±5.68	82.61±2.69	90.00±3.60	73.23±2.51	66.68±3.22	69.69±5.10	65.59±5.00	69.40±5.21
PersLay	82.16±3.18	80.90±1.00	89.80±0.90	74.80±0.30	N/A	N/A	N/A	N/A
FC-V	85.61±0.59	81.01±0.88	87.31±0.66	74.54±0.48	N/A	N/A	N/A	N/A
MPR	N/A	N/A	84.00±8.60	75.20±2.20	66.36±6.55	68.60±6.30	63.94±5.19	64.27±3.78
SIN	N/A	N/A	N/A	76.50±3.40	66.80±4.56	70.55±4.79	68.68±6.80	69.80±4.36
Wit-TopoPool	**87.80±2.44** ***	**87.24±3.15**	**93.16±4.11** **	**80.00±3.22** *	**70.57±4.43** ***	**79.12±4.45**	**71.71±4.86**	**75.00±3.51** ***

Table 1: Performance on molecular and chemical graphs. The best results are given in bold while the best performances achieved by the runner-ups are underlined.

sLay, FC-V, MPR, and SIN are the state-of-the-art topological and simplicial complex-based models, specialized on extracting topological information and higher-order structures from the observed graphs. A common limitation of these approaches is that they do not simultaneously capture both local and global topological properties of the graph. Hence, it is not surprising that performance of Wit-TopoPool which systematically integrates all types of the above information on the observed graphs is substantially higher than that of the benchmark models.

Social Graphs Table 2 shows the performance comparison on 3 social graph datasets. Similarly, Table 2 indicates that our Wit-TopoPool model is always better than baselines for all social graph datasets. We find that, even compared to the baselines (which feeds neural networks with topological summaries (i.e., PersLay) or integrates higher-order structures into GNNs (i.e., SIN)), Wit-TopoPool is highly competitive, revealing that global and local topological representation learning modules can enhance the model expressiveness.

Model	IMDB-B	IMDB-M	REDDIT-B
CSM	OOT	OOT	OOT
HGK-SP	73.34±0.47	51.58±0.42	OOM
HGK-WL	72.75±1.02	50.73±0.63	OOM
WL	71.15±0.47	50.25±0.72	77.95±0.60
WL-OA	74.01±0.66	49.95±0.46	87.60±0.33
DGCNN	70.00±0.90	47.80±0.90	76.00±1.70
GCN	66.53±2.33	48.93±0.88	89.90±1.90
GIN	75.10±5.10	52.30±2.80	92.40±2.50
Top-K	73.17±4.84	48.80±3.19	79.40±7.40
MinCutPool	70.77±4.89	49.00±2.83	87.20±5.00
DiffPool	68.60±3.10	45.70±3.40	79.00±1.10
EigenGCN	70.40±3.30	47.20±3.00	N/A
SAGPool	74.87±4.09	49.33±4.90	84.70±4.40
HaarPool	73.29±3.40	49.98±5.70	N/A
PersLay	71.20±0.70	48.80±0.60	N/A
FC-V	73.84±0.36	46.80±0.37	89.41±0.24
MPR	73.80±4.50	50.90±2.50	86.20±6.80
SIN	75.60±3.20	52.50±3.00	92.20±1.00
Wit-TopoPool	*****78.40±1.50**	**53.33±2.47**	***92.82±1.10**

Table 2: Performance on social graphs. The best results are given in bold while the best performances achieved by the runner-ups are underlined.

Ablation Study

To evaluate the contributions of the different components in our Wit-TopoPool model, we perform exhaustive ablation studies on COX2, PTC_MM, and IMDB-B datasets. We use Wit-TopoPool as the baseline architecture and consider three ablated variants: (i) Wit-TopoPool without topological pooling enhanced graph convolutional layer (W/o TPGCL), (ii) Wit-TopoPool without witness complex-based topological layer (W/o Wit-TL), and (iii) Wit-TopoPool without attention mechanism (W/o Attention Mechanism). The experimental results are shown in Table 3 and we prove the validity of each component. As Table 3 suggest, we find that (i) ablating each of above component leads to the performance drops in comparison with the full Wit-TopoPool model, thereby, indicating that each of the designed components contributes to the success of Wit-TopoPool, (ii) on

all three datasets, TPGCL module significantly improves the classification results, i.e., Wit-TopoPool outperforms Wit-TopoPool w/o TPGCL with an average relative gain 5.30% over three datasets – this phenomenon implies that learning both local topological information and node features are critical for successful graph learning, (iii) in comparison to Wit-TopoPool and Wit-TopoPool w/o Wit-TL, Wit-TopoPool always outperforms because the Wit-TL module enables the model to effectively incorporate more global topological information, demonstrating the significance of the proposed global topological representation learning module for graph classification, and (iv) Wit-TopoPool consistently outperforms Wit-TopoPool w/o Attention Mechanism on all 3 datasets, indicating the attention mechanism can successfully extract the most correlated information and, hence, improves the generalization of unseen graph structures. Moreover, we also compare Wit-TopoPool with VR-TopoPool (i.e., replacing witness complex in global information learning with Vietoris-Rips complex) (see Appendix B for a discussion).

	Architecture	Accuracy mean±std
COX2	**Wit-TopoPool**	***87.24±3.15**
	W/o TPGCL	82.67±3.26
	W/o Wit-TL	85.21±3.20
	W/o Attention mechanism	85.58±3.53
PTC_MM	**Wit-TopoPool**	**76.76±5.78**
	W/o TPGCL	67.38±5.33
	W/o Wit-TL	70.58±5.29
	W/o Attention mechanism	75.12±5.59
IMDB-B	**Wit-TopoPool**	****78.40±1.50**
	W/o TPGCL	73.93±1.83
	W/o Wit-TL	77.00±1.69
	W/o Attention mechanism	76.20±1.98

Table 3: Ablation study of the Wit-TopoPool architecture.

Sensitivity Analysis We perform sensitivity analysis of (i) landmark set selection and (ii) topological score function to explore the effect of above two components on our Wit-TopoPool performance. The optimal choice of landmark set selection and topological score function can be obtained via cross-validation. We first explore the effect of landmark set selection. We consider 3 types of landmark set selections, i.e., (i) randomly ($\mathfrak{L}_r$), (ii) node betweenness centrality ($\mathfrak{L}_b$), and (iii) node degree centrality ($\mathfrak{L}_d$), and report results on COX2 and PTC_MM datasets. As Table 4 shows, we observe that the landmark set selection based on either node betweenness or degree centrality helps to improve the graph classification performance, whereas the landmark set based on randomly results in the performance drop. We also explore the effect of topological score function for the importance measurement of the persistence diagram (see Eq. 2). As the results in Table 5 suggest, summing over lifespans of topological features (points) in persistence diagrams can significantly improve performance, but applying piecewise linear weighting function on topological features may result in deterioration of performance.

Dataset	Landmark set	Accuracy mean±std
COX2	$\mathfrak{L}_r$	82.98±3.88
	$\mathfrak{L}_b$	85.10±2.52
	$\mathfrak{L}_d$	**87.24±3.15**
PTC_MM	$\mathfrak{L}_r$	71.53±6.17
	$\mathfrak{L}_b$	**79.12±4.45**
	$\mathfrak{L}_d$	76.76±5.78

Table 4: Sensitivity analysis with respect to the landmark set selection for Wit-TopoPool on COX2 and PTC_MM.

Dataset	Weighting function	Accuracy mean±std
COX2	$d_\rho - b_\rho$	*****87.24±3.15**
	$\arctan\left(C \times ((d_\rho - b_\rho)^\eta\right)$	79.78±1.06
PTC_MM	$d_\rho - b_\rho$	***79.12±4.45**
	$\arctan\left(C \times ((d_\rho - b_\rho)^\eta\right)$	76.18±5.00

Table 5: Sensitivity analysis with respect to selection of weighting functions within the topological score for Wit-TopoPool on COX2 and PTC_MM.

Computational Complexity Computational complexity of the standard persistent homology matrix reduction algorithm (Edelsbrunner, Letscher, and Zomorodian 2000) (i.e., based on column operations over boundary matrix of the complex) runs in cubic time in the worst case, i.e., $\mathcal{O}(m^3)$, where m is the number of simplices in the filtration. For 0-dimensional PH, it can be computed efficiently using disjoint sets with complexity $\mathcal{O}(m\alpha^{-1}m)$, where $\alpha^{-1}(\cdot)$ is the inverse Ackermann function (Cormen et al. 2022). Computational complexity of the witness complex construction is $\mathcal{O}(\mathfrak{L}\log(n))$ (where n is the number of data points and $\mathfrak{L}$ is the landmark set), involving calculating the distance between data points and landmark points.

Conclusion

In this paper, we have proposed Wit-TopoPool, a differentiable and comprehensive pooling operator for graph classification that simultaneously extracts the key topological characteristics of graphs at both local and global levels, using the notions of persistence, landmarks, and witnesses. In the future, we will expand the ideas of learnable topological representations and adaptive similarity learning among nodes to dynamic and multilayer networks.

Acknowledgements

This work was supported by the NSF grant # ECCS 2039701 and ONR grant # N00014-21-1-2530. Part of this material is also based upon work supported by (while serving at) the NSF. The views expressed in the article do not necessarily represent the views of NSF and ONR.

References

Bianchi, F. M.; Grattarola, D.; and Alippi, C. 2020. Spectral clustering with graph neural networks for graph pooling. In *ICML*, 874–883.

Bodnar, C.; Cangea, C.; and Liò, P. 2021. Deep graph mapper: Seeing graphs through the neural lens. *Frontiers in Big Data*, 38.

Bodnar, C.; Frasca, F.; Wang, Y.; Otter, N.; Montufar, G. F.; Lio, P.; and Bronstein, M. 2021. Weisfeiler and lehman go topological: Message passing simplicial networks. In *ICML*, 1026–1037.

Boureau, Y.-L.; Ponce, J.; and LeCun, Y. 2010. A theoretical analysis of feature pooling in visual recognition. In *ICML*, 111–118.

Cangea, C.; Veličković, P.; Jovanović, N.; Kipf, T.; and Liò, P. 2018. Towards sparse hierarchical graph classifiers. *Workshop on Relational Representation Learning, NeurIPS*.

Carlsson, G. 2020. Topological methods for data modelling. *Nature Reviews Physics*, 2(12): 697–708.

Carlsson, G.; and Vejdemo-Johansson, M. 2021. *Topological Data Analysis with Applications*. Cambridge University Press.

Carrière, M.; Chazal, F.; Ike, Y.; Lacombe, T.; Royer, M.; and Umeda, Y. 2020. Perslay: A neural network layer for persistence diagrams and new graph topological signatures. In *AISTATS*, 2786–2796.

Cormen, T. H.; Leiserson, C. E.; Rivest, R. L.; and Stein, C. 2022. *Introduction to algorithms*. MIT press.

De Silva, V.; and Carlsson, G. E. 2004. Topological estimation using witness complexes. In *PBG*, 157–166.

Defferrard, M.; Bresson, X.; and Vandergheynst, P. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. *NIPS*, 29.

Edelsbrunner, H.; Letscher, D.; and Zomorodian, A. 2000. Topological persistence and simplification. In *FOCS*, 454–463.

Gao, H.; and Ji, S. 2019. Graph U-nets. In *ICML*, 2083–2092.

Girdhar, R.; and Ramanan, D. 2017. Attentional pooling for action recognition. In *NIPS*, volume 30.

Hofer, C.; Graf, F.; Rieck, B.; Niethammer, M.; and Kwitt, R. 2020. Graph filtration learning. In *ICML*, 4314–4323.

Huang, J.; Li, Z.; Li, N.; Liu, S.; and Li, G. 2019. Attpool: Towards hierarchical feature representation in graph convolutional networks via attention mechanism. In *IEEE/CVF ICCV*, 6480–6489.

Kipf, T. N.; and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.

Kriege, N.; and Mutzel, P. 2012. Subgraph matching kernels for attributed graphs. In *ICML*, 291–298.

Kriege, N. M.; Giscard, P.-L.; and Wilson, R. 2016. On valid optimal assignment kernels and applications to graph classification. In *NeurIPS*.

Lee, J.; Lee, I.; and Kang, J. 2019. Self-attention graph pooling. In *ICML*, 3734–3743.

Ma, Y.; Wang, S.; Aggarwal, C. C.; and Tang, J. 2019. Graph convolutional networks with eigenpooling. In *SIGKDD*, 723–731.

Morris, C.; Kriege, N. M.; Kersting, K.; and Mutzel, P. 2016. Faster kernels for graphs with continuous attributes via hashing. In *ICDM*, 1095–1100.

O'Bray, L.; Rieck, B.; and Borgwardt, K. 2021. Filtration Curves for Graph Representation. In *SIGKDD*, 1267–1275.

Otter, N.; Porter, M. A.; Tillmann, U.; Grindrod, P.; and Harrington, H. A. 2017. A roadmap for the computation of persistent homology. *EPJ Data Science*, 6: 1–38.

Poklukar, P.; Varava, A.; and Kragic, D. 2021. Geomca: Geometric evaluation of data representations. In *ICML*, 8588–8598.

Schönenberger, S. T.; Varava, A.; Polianskii, V.; Chung, J. J.; Kragic, D.; and Siegwart, R. 2020. Witness autoencoder: Shaping the latent space with witness complexes. In *NeurIPS 2020 Workshop on TDA and Beyond*.

Shen, Y.; Feng, C.; Yang, Y.; and Tian, D. 2018. Mining point cloud local structures by kernel correlation and graph pooling. In *CVPR*, 4548–4557.

Shervashidze, N.; Schweitzer, P.; Van Leeuwen, E. J.; Mehlhorn, K.; and Borgwardt, K. M. 2011. Weisfeiler-lehman graph kernels. *JMLR*, 12(9).

Tauzin, G.; Lupo, U.; Tunstall, L.; Pérez, J.; Caorsi, M.; Medina-Mardones, A.; Dassatti, A.; and Hess, K. 2021. giotto-tda:: A Topological Data Analysis Toolkit for Machine Learning and Data Exploration. *JMLR*, 22: 39–1.

Wang, Y. G.; Li, M.; Ma, Z.; Montufar, G.; Zhuang, X.; and Fan, Y. 2020. Haar graph pooling. In *ICML*, 9952–9962.

Xia, F.; Sun, K.; Yu, S.; Aziz, A.; Wan, L.; Pan, S.; and Liu, H. 2021. Graph learning: A survey. *IEEE Trans AI*, 2(2): 109–127.

Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2018. How Powerful are Graph Neural Networks? In *ICLR*.

Yang, J.; Zhao, P.; Rong, Y.; Yan, C.; Li, C.; Ma, H.; and Huang, J. 2021. Hierarchical graph capsule network. In *AAAI*.

Ying, Z.; You, J.; Morris, C.; Ren, X.; Hamilton, W.; and Leskovec, J. 2018. Hierarchical graph representation learning with differentiable pooling. In *NeurIPS*, volume 31.

Yu, F.; and Koltun, V. 2016. Multi-scale context aggregation by dilated convolutions. In *ICLR*.

Zhang, M.; Cui, Z.; Neumann, M.; and Chen, Y. 2018. An end-to-end deep learning architecture for graph classification. In *AAAI*, volume 32.

Zhou, J.; Cui, G.; Hu, S.; Zhang, Z.; Yang, C.; Liu, Z.; Wang, L.; Li, C.; and Sun, M. 2020. Graph neural networks: A review of methods and applications. *AI Open*, 1: 57–81.

Zomorodian, A.; and Carlsson, G. 2005. Computing persistent homology. *Discrete & Computational Geometry*, 33(2): 249–274.