EMP: Effective Multidimensional Persistence for Graph Representation Learning

Ignacio Segovia-Dominguez*
West Virginia University
Ignacio.SegoviaDominguez@mail.wvu.edu

Yuzhou Chen*
Temple University
yuzhou.chen@temple.edu

Cuneyt G. Akcora
University of Central Florida
cuneyt.akcora@ucf.edu

Zhiwei Zhen
University of Texas at Dallas
zhiwei.zhen@utdallas.edu

Murat Kantarcioglu
University of Texas at Dallas
muratk@utdallas.edu

Yulia R. Gel
University of Texas at Dallas
National Science Foundation, USA
ygl@utdallas.edu

Baris Coskunuzer
University of Texas at Dallas
coskunuz@utdallas.edu

Abstract

Topological data analysis (TDA) is gaining prominence across a wide spectrum of machine learning tasks that spans from manifold learning to graph classification. A pivotal technique within TDA is persistent homology (PH), which furnishes an exclusive topological imprint of data by tracing the evolution of latent structures as a scale parameter changes. Present PH tools are confined to analyzing data through a single filter parameter. However, many scenarios necessitate the consideration of multiple relevant parameters to attain finer insights into the data. We address this issue by introducing the Effective Multidimensional Persistence (EMP) framework. This framework empowers the exploration of data by simultaneously varying multiple scale parameters. The framework integrates descriptor functions into the analysis process, yielding a highly expressive data summary. It seamlessly integrates established single PH summaries into multidimensional counterparts like EMP Landscapes, Silhouettes, Images, and Surfaces. These summaries represent data's multidimensional aspects as matrices and arrays, aligning effectively with diverse ML models. We provide theoretical guarantees and stability proofs for EMP summaries. We demonstrate EMP's utility in graph classification tasks, showing its effectiveness. Results reveal that EMP enhances various single PH descriptors, outperforming cutting-edge methods on multiple benchmark datasets.

1 Introduction

In the past decade, topological data analysis proved to be a powerful machinery to discover hidden patterns in various forms of data that are otherwise inaccessible with more traditional methods [1, 2]. In particular, for graph machine learning (ML) tasks, while many traditional methods fail, TDA and, specifically, tools of persistent homology (PH), have demonstrated a high potential to detect local and global patterns and to produce a unique topological fingerprint to be used in various ML tasks [3]. This makes PH particularly attractive for capturing various characteristics of the complex data which may play the key role behind the learning task performance.

In turn, multiparameter persistence (multipersistence or MP) is a novel idea to further advance the PH machinery by analyzing the data in a much finer way simultaneously along multiple dimensions.

*Equal contribution.

EMP: Effective Multidimensional Persistence for Graph Representation Learning. *Proceedings of the Second Learning on Graphs Conference (LoG 2023)*, PMLR 231, Virtual Event, November 27–30, 2023.

However, due to technical issues stemming from its multidimensional structure in commutative algebra, a general definition for MP has not yet been established. In this paper, we develop an alternative approach to utilize multipersistence ideas efficiently for various types of data, with a main focus on graph representation. In particular, we bypass technical issues with the MP (Appendix B.5) by extracting very practical summaries via the slicing approach in a structured way. We then obtain computationally efficient multidimensional topological fingerprints of the data in the form of matrices and linear arrays which can serve as input to ML models.

Our main contribution lies in employing interpretable persistence features to achieve accurate graph classification. This is crucial because existing graph neural networks (GNNs) demand extensive data and costly training, yet lack insights into specific classification outcomes. Unlike GNNs, our single and multi-parameter persistence features offer insights into the evolving underlying graphs as the scale parameter changes, enabling the creation of efficient machine learning models. In domains with limited graph data, the predictive models we construct using topological summaries from single and multi-parameter persistence achieve accuracy levels comparable to the top GNN models.

Our contributions can be summarized as follows:

- ➢ We develop a computationally efficient and highly expressive EMP framework that provides multidimensional topological fingerprints of the data. EMP expands popular summaries of single persistence to multidimensions by adapting an effective slicing direction.
- ➢ We derive theoretical stability guarantees of the new topological summaries.
- ➢ We illustrate the utility of EMP summaries in various settings and compare our results to state-of-the-art (SOTA) methods. Our numerical experiments demonstrate that EMP summaries outperform SOTA in a broad range of benchmark datasets for graph classification tasks.

2 Related Work

2.1 Persistent Homology

Persistent homology is a key tool in topological data analysis to deliver invaluable and complementary information on the intrinsic properties of data that are inaccessible with conventional methods [4, 5]. In the past decade, PH has become quite popular in various ML tasks, ranging from manifold learning to medical image analysis, and material science to finance (TDA applications library [6]).

Multipersistence is a highly promising approach with the potential to significantly improve the success of single parameter persistence (SP) in applications of TDA, but there exist some fundamental challenges related to converting this novel idea into an effective feature extraction method, as follows. Except for some special cases, MP theory tends to suffer from the problem of the nonexistence of barcode decomposition because of the partially ordered structure of the index set $\{(\alpha_i, \beta_j)\}$ [7]. Lesnick et al. [8] proposed a solution to circumvent this problem using the *slicing technique*. This method involves examining one-dimensional fibers within the multiparameter domain. In these fibers, the multidimensional persistence module is constrained to a single direction (referred to as a *slice*), and single persistence analysis is applied to this one-dimensional slice. Later, by using this novel idea, Carrière et al. [9] combined several slicing directions (vineyards) and obtained a vectorization by summarizing the persistence diagrams (PDs) in these directions. There are several promising recent studies in this direction [10, 11]. However, these methods often come with several drawbacks, primarily related to computational expenses. Consequently, their practical use for real-world problems is constrained. Very recently, a couple of very successful MP vectorizations proved the potential of MP in several settings [12, 13]. Here, we aim to add a practical and highly efficient way to use MP approach for various forms of data and provide a multidimensional topological vectorization with EMP summaries.

2.2 Graph Neural Networks

After the success of convolutional neural networks (CNN) on image-based tasks, graph neural networks (GNNs) have emerged as powerful tools for graph-level classification and representation learning. Based on the spectral graph theory, Bruna et al. introduced a graph-based convolution in the Fourier domain [14] . However, the complexity of this model is high since all Laplacian eigenvectors are needed. To tackle this problem, ChebNet [15] integrated spectral graph convolution

with Chebyshev polynomials. Then, Graph Convolutional Networks (GCNs) of [16] simplified the graph convolution with a localized first-order approximation. More recently, various approaches proposed accumulating graph information from a wider neighborhood, using diffusion aggregation and random walks. Such higher-order methods include approximate personalized propagation of neural predictions (APPNP) [17], higher-order graph convolutional architectures (MixHop) [18], multi-scale graph convolution (N-GCN) [19], and Lévy Flights Graph Convolutional Networks (LFGCN) [20]. In addition to random walks, other recent approaches include GNNs on directed graphs (MotifNet) [21], graph convolutional networks with attention mechanism (GAT, SPAGAN) [22, 23], and graph Markov neural network (GMNN) [24]. While GNNs produce great performances in many graph learning tasks, they tend to suffer from over-smoothing problems and are vulnerable to graph perturbations.

3 Background

We start with providing the basic background for our framework. Since our primary focus pertains to graph representation learning in this study, we elucidate our methodology within the context of graphs. For a comprehensive treatment of the classical persistent homology construction in various contexts, consult [25]. Due to space limitations, our methodology on alternative data forms, such as point clouds and image data are explained in Appendices B.2 and B.3.

3.1 Persistent Homology

Persistent Homology is a mathematical machinery to capture the hidden shape patterns in the data by using algebraic topology tools. PH extracts this information by keeping track of evolution of the topological features (components, loops, cavities) created in the data while looking at it in different resolutions [4].

For a given graph $\mathcal{G}$, consider a nested sequence of subgraphs $\mathcal{G}_1 \subseteq \ldots \subseteq \mathcal{G}_N = \mathcal{G}$. For each $\mathcal{G}_i$, define an abstract simplicial complex $\widehat{\mathcal{G}}_i$, $1 \leq i \leq N$, yielding a filtration of complexes $\widehat{\mathcal{G}}_1 \subseteq \ldots \subseteq \widehat{\mathcal{G}}_N$. Here, clique complexes are among the most common ones, i.e., a clique complex $\widehat{\mathcal{G}}$ is obtained by assigning (filling with) a k-simplex to each complete $(k+1)$-complete subgraph in $\mathcal{G}$. For example, a 3-clique in $\mathcal{G}$, which is a complete 3-subgraph, will be filled with a 2-simplex (triangle). Then, in this sequence of simplicial complexes, we can systematically keep track of the evolution of the topological patterns. A k-dimensional topological feature (or k-hole) represent connected components (0-hole), loops (1-hole) and cavities (2-hole). For each k-hole σ, PH records its first appearance (birth) in the filtration sequence, say $\widehat{\mathcal{G}}_{b_\sigma}$, and disappearance (death) in later complexes, $\widehat{\mathcal{G}}_{d_\sigma}$ with a unique pair (b_σ, d_σ), where $1 \leq b_\sigma < d_\sigma \leq N$. PH records all these birth and death times of the topological features in *persistence diagrams* (PDs). Let $0 \leq k \leq D$ where D is the highest dimension in the simplicial complex $\widehat{\mathcal{G}}_N$. Then the kth persistence diagram $\mathrm{PD_k}(\mathcal{G}) = \{(b_\sigma, d_\sigma) \mid \sigma \in H_k(\widehat{\mathcal{G}}_i) \text{ for } b_\sigma \leq i < d_\sigma\}$. Here, $H_k(\widehat{\mathcal{G}}_i)$ represents the kth homology group of $\widehat{\mathcal{G}}_i$ which keeps the information of the k-holes in the simplicial complex $\widehat{\mathcal{G}}_i$. For the sake of notation, we skip the dimension (subscript k). With the intuition that the topological features with a long life span (persistent features) describe the hidden shape patterns in the data, these PDs provide a unique topological fingerprint of the graph $\mathcal{G}$.

As one can easily notice, the most important step in the PH machinery is the construction of the nested sequence of subgraphs $\mathcal{G}_1 \subseteq \ldots \subseteq \mathcal{G}_N = \mathcal{G}$. For a given unweighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V} = \{v_1, \ldots, v_N\}$ the set of nodes and $\mathcal{E} \subset \{\{v_i, v_j\} \in \mathcal{V} \times \mathcal{V}, i \neq j\}$ the set of edges, the most common technique is to use a filtering function $f : \mathcal{V} \to \mathbb{R}$ with a choice of thresholds $\mathcal{I} = \{\alpha_i\}$ where $\alpha_1 = \min_{v \in \mathcal{V}} f(v) < \alpha_2 < \ldots < \alpha_N = \max_{v \in \mathcal{V}} f(v)$. For $\alpha_i \in \mathcal{I}$, let $\mathcal{V}_i = \{v_r \in \mathcal{V} \mid f(v_r) \leq \alpha_i\}$. Let $\mathcal{G}_i$ be the induced subgraph of $\mathcal{G}$ by $\mathcal{V}_i$, i.e., $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i)$ where $\mathcal{E}_i = \{e_{rs} \in \mathcal{E} \mid v_r, v_s \in \mathcal{V}_i\}$. This process yields a nested sequence of subgraphs $\mathcal{G}_1 \subset \mathcal{G}_2 \subset \ldots \subset \mathcal{G}_N = \mathcal{G}$, called *the sublevel filtration* induced by the filtering function f. We denote PDs obtained via sublevel filtration for a filtering function f as $\mathrm{PD}(\mathcal{G}, \mathrm{f})$. The choice of f is crucial here, and in most cases, f is either an important function from the domain of the data, e.g., atomic number for chemical compounds, or a function defined from intrinsic properties of the graph, e.g., degree and betweenness. Similarly, for a weighted graph, one can use sublevel filtration on the weights of the edges and obtain a suitable filtration reflecting the domain information stored in the edge weights.

3.2 Single Persistence Vectorizations

While PH extracts hidden shape patterns from data as persistence diagrams (PD), PDs being collections of points in $\mathbb{R}^2$ by itself are not very practical for statistical and machine learning purposes. Instead, the common technique is to faithfully represent PDs as kernels [26] or vectorizations [27]. This provides a practical way to use the outputs of PH in real-life applications. *Single Persistence Vectorizations* transform obtained PH information (i.e., PDs) into a function or a feature vector form which are much more suitable for machine learning tools than PDs. Common single persistence (SP) vectorization methods are Persistence Images [28], Persistence Landscapes [29], Silhouettes [30], and various Persistence Curves [31]. These vectorizations define a single variable or multivariable functions out of PDs, which can be used as fixed size $1D$ or $2D$ vectors in applications, i.e. $1 \times n$ vectors or $m \times n$ vectors. For example, a Betti curve for a PD with n thresholds can also be expressed as $1 \times n$ size vectors. Similarly, Persistence Images is an example of $2D$ vectors with the chosen resolution (grid) size.

3.3 Multidimensional Persistence

MultiPersistence introduces a novel concept that holds the potential to significantly enhance the performance of single-parameter persistence. The term *single* is applied because we filter the data in just one direction, $\mathcal{G}_1 \subset \cdots \subset \mathcal{G}_N = \mathcal{G}$. The filtration's construction is pivotal in achieving a detailed analysis of the data to uncover concealed patterns. When utilizing a single function $f : \mathcal{V} \to \mathbb{R}$ containing crucial domain information (e.g., value for blockchain networks, atomic number for protein networks), it induces a single-parameter filtration as described earlier.

However, numerous datasets possess multiple highly relevant domain functions for data analysis. Employing these functions concurrently would yield a more comprehensive grasp of the concealed patterns. This insight led to the suggestion of the MP theory as a natural extension of single persistence (SP).

In simpler terms, if we use only one filtering function, sublevel sets induce a single parameter filtration $\widehat{\mathcal{G}}_1 \subset \cdots \subset \widehat{\mathcal{G}}_N = \widehat{\mathcal{G}}$. Instead, if we use two or more functions, then it would give a way to study the data in a much finer resolution. For example, if we have two node functions $f : \mathcal{V} \to \mathbb{R}$ and $g : \mathcal{V} \to \mathbb{R}$ with valuable complementary information of the network, MP is presumed to produce a unique topological fingerprint combining the information from both functions. These pair of functions f, g induce a multivariate filtering function $F : \mathcal{V} \to \mathbb{R}^2$ with $F(v) = (f(v), g(v))$. Again, we can define a set of non-decreasing thresholds $\{\alpha_i\}_1^m$ and $\{\beta_j\}_1^n$ for f and g respectively. Then, $\mathcal{V}_{ij} = \{v_r \in \mathcal{V} \mid f(v_r) \leq \alpha_i, g(v_r) \leq \beta_j\}$, i.e., $\mathcal{V}_{ij} = F^{-1}((-\infty, \alpha_i] \times (-\infty, \beta_j])$. Then, let $\mathcal{G}_{ij}$ be the induced subgraph of $\mathcal{G}$ by $\mathcal{V}_{ij}$, i.e., the smallest subgraph of $\mathcal{G}$ generated by $\mathcal{V}_{ij}$. This induces a bifiltration of complexes $\{\widehat{\mathcal{G}}_{ij} \mid 1 \leq i \leq m, 1 \leq j \leq n\}$. We can imagine $\{\widehat{\mathcal{G}}_{ij}\}$ as a rectangular grid of size $m \times n$ (See Figure 3).

By computing the homology groups of these complexes, $\{H_k(\widehat{\mathcal{G}}_{ij})\}$, we obtain the induced bi-graded persistence module (a rectangular grid of size $m \times n$). Again, the idea is to keep track of the k-dimensional topological features via the homology groups $\{H_k(\widehat{\mathcal{G}}_{ij})\}$ in this grid. As we explained in Appendix B.5, because of the technical issues related to commutative algebra, converting the multipersistence module into a mathematical representation like a "Multipersistence Diagram" is unfeasible. As a result, we do not have an effective vectorization of the MP module. These technical obstacles prevent this promising approach to reach its full potential in real-life applications.

In this paper, we overcome this problem by producing practical and computationally efficient vectorizations by utilizing the slicing idea in the multipersistence grid in a structured way, as we describe next.

4 Effective Multidimensional Persistence Summaries

We now introduce our Effective MultiPersistence framework which describes a way to expand most single persistence vectorizations (Section 3.2) as multidimensional vectorizations by utilizing the MP approach. In particular, by using the existing single-parameter persistence vectorizations, we produce multidimensional vectors by effectively using one of the filtering directions as the *slicing direction* in

the multipersistence module. We give the basic setup in this section, and the generalization directions and further details in Appendix B.

4.1 EMP Framework

To keep the exposition simple, we describe our framework for 2-parameter multipersistence ($d = 2$). For $d > 2$, the construction is similar and given below. The outline is as follows: For two given filtering functions f, g for a graph $\mathcal{G}$, we use the first function f to get a single parameter filtering of the data, i.e., $\mathcal{G}_1 \subseteq \ldots \subseteq \mathcal{G}_m = \mathcal{G}$. Then, we use the second function in each subgraph $\mathcal{G}_i$ to obtain persistence diagram $\mathrm{PD}(\mathcal{G}_\mathrm{i}, \mathrm{g})$ for $1 \leq i \leq m$. Hence, we obtain m persistence diagrams $\{\mathrm{PD}(\mathcal{G}_\mathrm{i}, \mathrm{g})\}_{\mathrm{i}=1}^{\mathrm{m}}$. Next, by applying the chosen SP vectorization φ to each PD, we obtain m different same length vector $\vec{\varphi}(\mathrm{PD}(\mathcal{G}_\mathrm{i}, \mathrm{g}))$, say $1 \times k$ vector. By combining all m $1D$-vectors, we obtain EMP vectorization $\mathbf{M}_\varphi$ with $\mathbf{M}_\varphi^i = \vec{\varphi}(\mathrm{PD}(\mathcal{G}_\mathrm{i}, \mathrm{g}))$ where $\mathbf{M}_\varphi^i$ represents the ith row of $\mathbf{M}_\varphi$ which is a $2D$-vector (matrix) of size $m \times k$ (See Figure 1).

Here, we give the details for sublevel filtration for two filtering functions. In Appendix B.3, we explain how to modify the construction for weight filtrations or Vietoris-Rips filtrations. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph, and let $f, g : \mathcal{V} \to \mathbb{R}$ be two filtering functions with threshold sets $\{\alpha_i\}_{i=1}^m$ and $\{\beta_j\}_{j=1}^n$ respectively. Let $\mathcal{V}_i = \{v_r \in \mathcal{V} \mid f(v_r) \leq \alpha_i\}$. Let $\mathcal{G}_i$ be the induced subgraph of $\mathcal{G}$ by $\mathcal{V}_i$. This gives a filtering of the graph (nested subgraphs) as $\mathcal{G}_1 \subseteq \ldots \subseteq \mathcal{G}_m = \mathcal{G}$. Recall that $g : \mathcal{V} \to \mathbb{R}$ is another filtering function for $\mathcal{G}$. Now, we fix $1 \leq i_0 \leq m$, and consider $\mathcal{G}_{i_0}$. By restricting g on $\mathcal{G}_{i_0}$, we get persistence diagram $\mathrm{PD}(\mathcal{G}_{\mathrm{i}_0}, \mathrm{g})$ as follows. Let $\mathcal{V}_{i_0 j} = \{v_r \in \mathcal{V}_{i_0} \mid f(v_r) \leq \beta_j\}$, and let $\mathcal{G}_{i_0 j}$ be the induced subgraph of $\mathcal{G}_{i_0}$ by $\mathcal{V}_{i_0 j}$. This defines a finer filtering of the graph $\mathcal{G}_{i_0}$ as $\mathcal{G}_{i_0 1} \subseteq \mathcal{G}_{i_0 2} \ldots \subseteq \mathcal{G}_{i_0 n} = \mathcal{G}_{i_0}$ (See Figure 3 in the appendix). Corresponding clique complexes defines a filtration $\widehat{\mathcal{G}}_{i_0 1} \subseteq \widehat{\mathcal{G}}_{i_0 2} \ldots \subseteq \widehat{\mathcal{G}}_{i_0 n} = \widehat{\mathcal{G}}_{i_0}$. This filtration gives the persistence diagram $\mathrm{PD}(\mathcal{G}_{\mathrm{i}_0}, \mathrm{g})$. Hence, for each $1 \leq i \leq m$, we obtain a persistence diagram $\mathrm{PD}(\mathcal{G}_\mathrm{i}, \mathrm{g})$.

The next step is to use vectorization on these m persistence diagrams. Let φ be a single persistence vectorization, e.g., Persistence Landscape, Silhouette, Entropy, Betti, Persistence Image. By applying the chosen SP vectorization φ to each PD, we obtain a function $\varphi_i = \varphi(\mathrm{PD}(\mathcal{G}_\mathrm{i}, \mathrm{g}))$ where in most cases it is a single variable function on the threshold domain $[\beta_1, \beta_n]$, i.e., $\varphi_i : [\beta_1, \beta_n] \to \mathbb{R}$. For the multivariable case (e.g., Persistence Image), we give an explicit description in the examples section below. Most such vectorizations being induced from a discrete set of points $\mathrm{PD}(\mathcal{G})$, they naturally can be expressed as a $1D$ vector of length k. In the examples below, we explain this conversion in detail. Then, let $\vec{\varphi}_i$ be the corresponding $1 \times k$ vector for the function φ_i. Now, we are ready to define our EMP summary $\mathbf{M}_\varphi$ which is a $2D$-vector (a matrix)

$$\mathbf{M}_\varphi^i = \vec{\varphi}_i \quad \text{for} \quad 1 \leq i \leq m,$$

where $\mathbf{M}_\varphi^i$ is the ith-row of $\mathbf{M}_\varphi$. Hence, $\mathbf{M}_\varphi$ is a $2D$-vector of size $m \times k$. Each row $\mathbf{M}_\varphi^i$ is the vectorization of the persistence diagram $\mathrm{PD}(\mathcal{G}_\mathrm{i}, \mathrm{g})$ via the SP vectorization method φ. We use the first filtering function f to get a finer look at the graph as $\mathcal{G}_1 \subseteq \ldots \subseteq \mathcal{G}_m = \mathcal{G}$. Then, we use the second filtering function g to obtain the persistence diagrams $\mathrm{PD}(\mathcal{G}_\mathrm{i}, \mathrm{g})$ of these finer pieces. In a way, we look at $\mathcal{G}$ with a two-dimensional resolution $\{\mathcal{G}_{ij}\}$ and we keep track of the evolution of topological features in the induced bifiltration $\{\widehat{\mathcal{G}}_{ij}\}$. The main advantage of this technique is that the outputs are fixed-size matrices (or arrays) for each dataset which is highly suitable for various machine learning models.

Order of the filtering functions. Since we only use horizontal slices, the first function is only used for finer filtration, and the second function gives the persistence diagrams. This makes our method a-symmetric (the order is important). Hence, one can change the order and get different performance results for the model. We discuss the order choice in detail and give experiments in Appendix B.4.

Higher Dimensional Multipersistence. Similarly, for $d = 3$, let f, g, h be filtering functions and let $\{\mathcal{G}_{ij}\}$ be the bifiltering of the data, e.g., sublevel filtration for two functions f, g. Then, again by using the third function h, we find $m \cdot n$ persistence diagrams $\{\mathrm{PD}(\mathcal{G}_{\mathrm{ij}}, \mathrm{h})\}_{\mathrm{i,j}=1}^{\mathrm{m,n}}$. Similarly, for a given SP vectorization φ, one obtains a $1D$-vector $\vec{\varphi}(\mathrm{PD}(\mathcal{G}_{\mathrm{ij}}, \mathrm{g}))$ of size $1 \times k$ for each i, j. This produces a $3D$-vector (array) $\mathbf{M}_\varphi$ of size $m \times n \times k$ where $\mathbf{M}_\varphi^{ij} = \vec{\varphi}(\mathrm{PD}(\mathcal{G}_{\mathrm{ij}}, \mathrm{g}))$. For $d > 3$, one could follow a similar route.

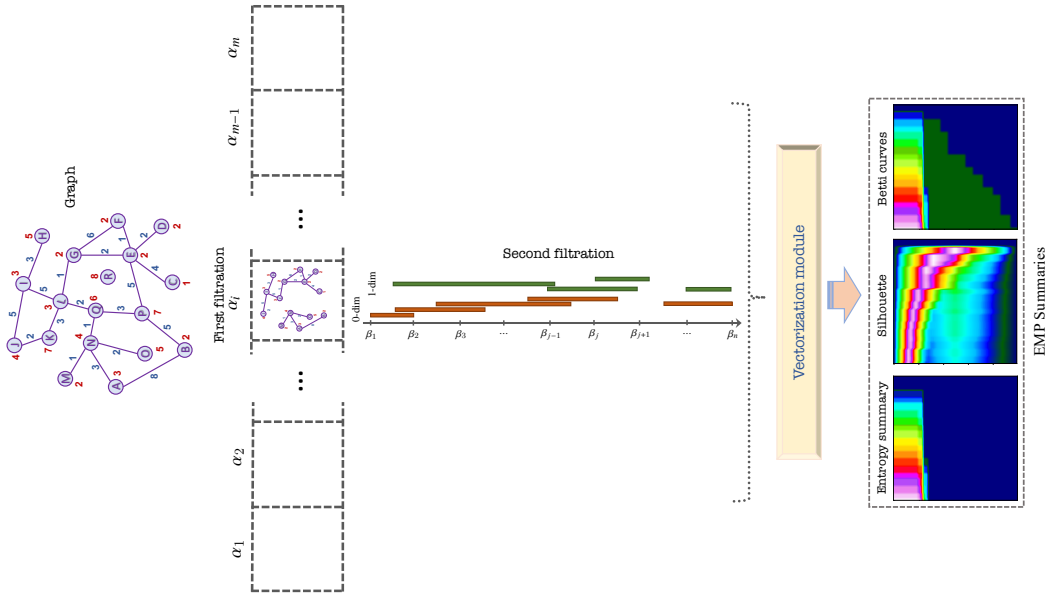

Figure 1: Illustration of the EMP framework for networks. Using the pair of filtering functions f, g we define non-decreasing thresholds $\{\alpha_i\}_1^m$ and $\{\beta_j\}_1^n$, respectively, based on node features, red, and edge features, blue. Both, filtrations and vectorizations run in parallel to better use computational resources and produce EMP representations in a timely manner.

4.2 Examples of EMP Summaries

Here, we discuss explicit constructions of EMP summaries for most common SP vectorizations. As noted above, the framework is very general and it can be applied to most SP vectorization methods. In all the examples below, we use the following setup: Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph, and let $f, g : \mathcal{V} \to \mathbb{R}$ be two filtering functions with threshold sets $\{\alpha_i\}_{i=1}^m$ and $\{\beta_j\}_{j=1}^n$ respectively. As explained above, we first apply sublevel filtering with f to get a sequence of nested subgraphs, $\mathcal{G}_1 \subseteq \ldots \subseteq \mathcal{G}_m = \mathcal{G}$. Then, for each $\mathcal{G}_i$, we apply sublevel filtration with g to get persistence diagram $\mathrm{PD}(\mathcal{G}_\mathrm{i}, \mathrm{g})$. Therefore, we will have m PDs. In the examples below, for a given SP vectorization φ, we explain how to obtain a vector $\vec{\varphi}(\mathrm{PD}(\mathcal{G}_\mathrm{i}, \mathrm{g}))$, and define the corresponding EMP $\mathbf{M}_\varphi$. Note that we skip the homology dimension (subscript k for $PD_k(\mathcal{G})$) in the discussion. In particular, for each dimension $k = 0, 1, \ldots,$ we will have one EMP $\mathbf{M}_\varphi(\mathcal{G})$ (a matrix or array) corresponding to $\{\vec{\varphi}(PD_k(\mathcal{G}_i, g))\}$. The most common dimensions are $k = 0$ and $k = 1$ in applications. Recently, Demir et al. [32] has successfully applied a similar vectorization in drug discovery problem.

EMP Landscapes. Persistence Landscapes λ are one of the most common SP vectorizations introduced by [29]. For a given persistence diagram $\mathrm{PD}(\mathcal{G}) = \{(\mathrm{b_i}, \mathrm{d_i})\}$, λ produces a function $\lambda(\mathcal{G})$ by using generating functions Λ_i for each $(b_i, d_i) \in \mathrm{PD}(\mathcal{G})$, i.e., $\Lambda_i : [b_i, d_i] \to \mathbb{R}$ is a piecewise linear function obtained by two line segments starting from $(b_i, 0)$ and $(d_i, 0)$ connecting to the same point $(\frac{b_i+d_i}{2}, \frac{d_i-b_i}{2})$. Then, the *Persistence Landscape* function $\lambda(\mathcal{G}) : [\epsilon_1, \epsilon_q] \to \mathbb{R}$ for $t \in [\epsilon_1, \epsilon_q]$ is defined as

$$\lambda(\mathcal{G})(t) = \max_i \Lambda_i(t)$$

where $\{\epsilon_k\}_1^q$ are thresholds for the filtration used.

Considering the piecewise linear structure of the function, $\lambda(\mathcal{G})$ is completely determined by its values at $2q - 1$ points, i.e., $\frac{b_i \pm d_i}{2} \in \{\epsilon_1, \epsilon_{1.5}, \epsilon_2, \epsilon_{2.5}, \ldots, \epsilon_q\}$ where $\epsilon_{k.5} = (\epsilon_k + \epsilon_{k+1})/2$. Hence, a vector of size $1 \times (2q - 1)$ whose entries the values of this function would suffice to capture all the information needed, i.e. $\vec{\lambda} = [\lambda(\epsilon_1)\ \lambda(\epsilon_{1.5})\ \lambda(\epsilon_2)\ \lambda(\epsilon_{2.5})\ \lambda(\epsilon_3)\ \ldots\ \lambda(\epsilon_q)]$

Considering we have threshold set $\{\beta_j\}_{j=1}^n$ for the second filtering function g, $\vec{\lambda}_i = \vec{\lambda}(\mathrm{PD}(\mathcal{G}_\mathrm{i}, \mathrm{g}))$ will be a vector of size $1 \times 2n - 1$. Then, as $\mathbf{M}_\lambda^i = \vec{\lambda}_i$ for each $1 \leq i \leq m$, EMP Landscape $\mathbf{M}_\lambda(\mathcal{G})$ would be a $2D$-vector (matrix) of size $m \times (2n - 1)$.

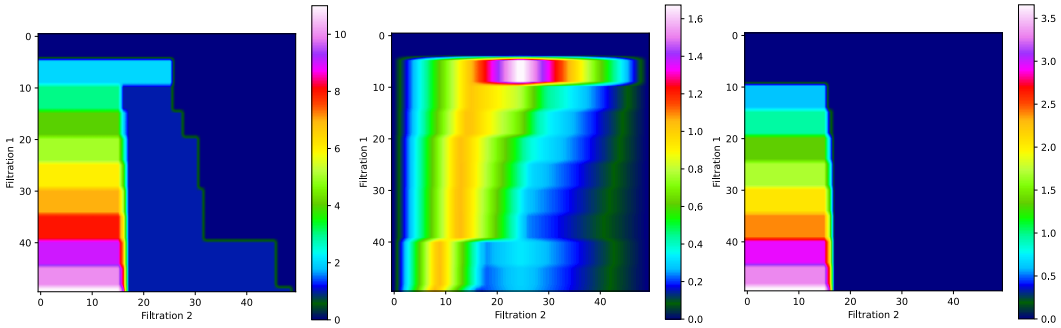

Figure 2: For the same network and the same filtering functions, EMP Betti Summary (left), EMP Silhouette (center), and EMP Entropy Summary (right) can produce highly different topological summaries emphasizing different information in persistence diagrams.

EMP Surfaces. Next, we give an important family of SP vectorizations, Persistence Curves [31]. This is an umbrella term for several different SP vectorizations, i.e., Betti Curves, Life Entropy, Landscapes, et al. Our EMP framework naturally adapts to all Persistence Curves to produce multidimensional vectorizations. As Persistence Curves produce a single variable function in general, they all can be represented as $1D$-vectors by choosing a suitable mesh size depending on the number of thresholds used. Here, we describe one of the most common Persistence Curves in detail, i.e., Betti Curves. It is straightforward to generalize the construction to other Persistence Curves.

Betti curves are one of the simplest SP vectorizations as it gives the count of the topological features at a given threshold interval. In particular, $\beta_k(\Delta)$ is the total count of k-dimensional topological feature in the simplicial complex Δ, i.e., $\beta_k(\Delta) = rank(H_k(\Delta))$ (See Figure 3 in the Appendix). In particular, $\beta_k(\mathcal{G}) : [\epsilon_1, \epsilon_{q+1}] \to \mathbb{R}$ is a step function defined as $\beta_k(\mathcal{G})(t) = rank(H_k(\widehat{\mathcal{G}}_i))$ for $t \in [\epsilon_i, \epsilon_{i+1})$, where $\{\epsilon_i\}_1^q$ represents the thresholds for the filtration used. Considering this is a step function where the function is constant for each interval $[\epsilon_i, \epsilon_{i+1})$, it can be perfectly represented by a vector of size $1 \times q$ as $\vec{\beta}(\mathcal{G}) = [\beta(1)\ \beta(2)\ \beta(3)\ \ldots\ \beta(q)]$.

Then, with the threshold set $\{\beta_j\}_{j=1}^n$ for the second filtering function g, $\vec{\beta}_i = \vec{\beta}(\mathrm{PD}(\mathcal{G}_\mathrm{i}, \mathrm{g}))$ will be a vector of size $1 \times n$. Then, as $\mathbf{M}_\beta^i = \vec{\beta}_i$ for each $1 \leq i \leq m$, *EMP Betti Summary* $\mathbf{M}_\beta(\mathcal{G})$ would be a $2D$-vector (matrix) of size $m \times n$ (Figure 2). In particular, each entry $\mathbf{M}_\beta = [m_{ij}]$ is just the Betti number of the corresponding clique complex in the bifiltration $\{\widehat{\mathcal{G}}_{ij}\}$, i.e., $m_{ij} = \beta(\widehat{\mathcal{G}}_{ij})$. This matrix $\mathbf{M}_\beta$ is also called bigraded Betti numbers in the literature, and computationally much faster than other vectorizations [33].

We give further EMP examples (*EMP Silhouettes* and *EMP Images*) in Appendix B.1.

4.3 Stability of EMP Summaries

We now demonstrate that when the original single-parameter vectorization φ is stable, the resulting EMP vectorization $\mathbf{M}_\varphi$ also maintains stability. The specifics of the stability concept in persistence theory are outlined, along with examples of stable SP vectorizations, in Appendix C.1.

We give generalizations of these stability notions and metrics in multidimensions and the proof of the following stability theorem in Appendix C.2.

Theorem 1. *Let φ be a stable SP vectorization. Then, the induced EMP Vectorization $\mathbf{M}_\varphi$ is also stable, i.e., with the notation given in Appendix C.2, there exists $\widehat{C}_\varphi > 0$ such that for any pair of graphs $\mathcal{G}^+$ and $\mathcal{G}^-$, we have the following inequality.*

$$\mathfrak{D}(\mathbf{M}_\varphi(\mathcal{G}^+), \mathbf{M}_\varphi(\mathcal{G}^-)) \leq \widehat{C}_\varphi \cdot \mathbf{D}_{p_\varphi}(\{\mathrm{PD}(\mathcal{G}^+)\}, \{\mathrm{PD}(\mathcal{G}^-)\}).$$

5 Experiments

In our experiments, we used nine widely used benchmark datasets in graph classification tasks. In particular, we consider (i) three molecule graphs [34]: BZR_MD, COX2_MD, and DHFR_MD;

Table 1: Accuracy. Classification accuracy (in % ± standard deviation) of EMP summary on nine benchmark datasets. The best results are in **bold** font and the second best results are marked *underlined*.

Model	BZR_MD	COX2_MD	DHFR_MD	MUTAG	PROTEINS	IMDB-B	IMDB-M	REDDIT-B	REDDIT-5K
CSM [35]	77.63±1.29	OOT	OOT	87.29±1.25	OOT	OOT	OOT	OOT	OOT
H-SP [37]	60.08±0.88	59.92±0.66	67.95±0.00	80.90±0.48	74.53±0.35	73.34±0.47	**51.58±0.42**	OOM	OOM
H-WL [37]	52.64±1.20	57.15±1.20	66.08±1.02	75.51±1.34	74.53±0.35	72.75±1.02	50.73±0.63	OOM	OOM
MLG [38]	51.46±0.61	51.15±0.00	67.95±0.00	78.53±2.25	75.55±0.71	52.56±0.42	34.27±0.33	OOM	OOM
WL [39]	67.45±1.40	60.07±2.22	62.56±1.51	85.75±1.96	73.06±0.47	71.15±0.47	50.25±0.72	77.95±0.60	51.63±0.37
WL-OA [40]	68.19±1.09	62.37±2.11	64.10±1.70	86.10±1.95	73.50±0.87	74.01±0.66	49.95±0.46	87.60±0.33	OOM
FC-V [41]	75.61±1.13	**73.41±0.79**	76.78±0.69	87.31±0.66	74.54±0.48	73.84±0.36	46.80±0.37	89.41±0.24	52.36±0.37
GNNs [42]	69.87±1.29	66.05±3.16	73.11±1.59	80.42±2.07	**75.80±3.70**	71.20±3.90	49.10±3.50	89.90±1.90	**56.10±1.60**
EMP	**77.76±0.95**	70.12±0.81	**80.13±0.94**	**88.79±0.63**	72.78±0.54	**74.44±0.45**	48.01±0.42	**91.03±0.22**	54.41±0.32

(ii) two biological graphs [35, 36]: MUTAG and PROTEINS; and (iii) four social graphs: IMDB-Binary (IMDB-B), IMDB-Multi (IMDB-M), REDDIT-Binary (REDDIT-B), and REDDIT-Multi-5K (REDDIT-M-5K). The dataset statistics are given in appendix Table 4.

5.1 Experimental Setup

To assess the effectiveness of our EMP summaries in graph representation learning, we assess them using the random forest (RF) classifier in a graph classification task. We select RF to underline EMP's adaptability, although EMP can seamlessly integrate with advanced DL models as a trainable component.

For the RF classifier, we fix the forest's tree count at 1000, the minimum sample requirement at 2, and the Gini impurity for split quality measurement. All EMP representations we introduce are vectorized.

We apply filtrations based on the available information of each dataset, either using their graph structure or their provided node/edge features. Our pool of filtering functions include: atomic weight, closeness, edge-betweenness, weighted degree, Katz centrality, and Ricci curvatures. We also use power filtration as the last filtration direction. To test our EMP framework we use three vectorizations: Betti curves, Silhouettes and Entropy Summary functions, thus producing EMP matrix representations that can be embedded in classic and modern machine learning algorithms.

We give further details on our experimental setup in Appendix A.1. Furthermore, Appendix A.2 includes an analysis of computational complexity. The source code is available in Python[2].

5.2 Experimental Results

We compare our model with three types of state-of-the-art baselines, covering six graph kernels, six graph neural networks (GNNs), and one topological method. Graph kernels: (i) comprised of the subgraph matching kernel (CSM) [35], (ii) Shortest Path Hash Graph Kernel (HGK-SP) [37], (iii) WL Hash Graph Kernel (HGK-WL) [37], (iv) Multiscale Laplacian Graph Kernel (MLG) [38], (v) Weisfeiler– Lehman (WL) [39], and (vi) WL Optimal Assignment (WL-OA) [40]; topological method: filtration curves (FC-V) [41], six graph neural networks including GCN, DGCNN, Diffpool, ECC, GIN, GraphSage which are compared in [42]. We report the best results of these six GNNs in the GNNs row in Table 1. For each dataset, we report our best performing model (Table 3). For all methods, we report the average accuracy of 10 runs of 10-fold CV along with the standard deviation.

Table 1 shows the results of different methods on nine graph datasets. Out-of-time (OOT) results indicate that a method could not complete the classification task within 12 hours, and OOM means "out-of-memory" (from an allocation of 128 GB RAM). In Table 2, we further compare our EMP model with other existing MultiPersistence vectorizations in the literature. Namely, MP landscapes (MP-L) [11], MP Images (MP-I) [9], multiparameter persistence kernel (MP-K) [43], the generalized rank invariant landscape (GRIL) [44], and MP Hilbert and Euler characteristic functions (MP-H and MP-E) [45] and MP Signed Barcodes (MP-SB) [13] where we reported the best of the four different vectorizations namely, MP-HSM-C, MP-ESM-C, MP-ESM-SW, MP-HSM-SW.

[2] https://www.dropbox.com/scl/fo/5kydyx2ivu1vpqi8hd0ob/h?rlkey=5u46k0x6p4hewpq5lhk9zpxl5&dl=0

We observe the following phenomena:

- ◇ Compared with all baselines, out of 9 benchmark datasets, the proposed EMP summaries achieve the best performance on 5 datasets (BZR_MD, DFHR_MD, MUTAG, IMDB-B, REDDIT-B), and second best in 2 datasets (COX2_MD, REDDIT-5K).
- ◇ EMP summary consistently outperforms Filtration Curves on all datasets except COX2_MD and PROTEINS, indicating that the multiparameter structure of the EMP summaries can better capture the complex structural properties and local topological information in heterogeneous graphs.
- ◇ When compared with other Multipersistence Vectorizations, EMP consistently ranks among top two performances.
- ◇ Moreover, EMP Summaries consistently deliver competitive results with GNNs and kernel methods in all benchmark datasets. This indicates that EMP summary introduces powerful topological and geometric schemes for node features and graph representation learning.

Table 2: Comparison with other Multipersistence Vectorizations

Dataset	MP-Kernel	MP-Landscape	MP-Images	GRIL	MP-Hilbert	MP-Euler	MP-SB	EMP
COX2	79.9±1.8	79.0±3.3	77.9±2.7	79.8±2.9	78.2±1.7	77.3±1.1	78.4±0.7	**79.9±0.8**
IMDB-B	68.2±1.2	71.2±2.0	71.1±2.1	65.2±2.6	73.0±4.5	72.0±1.9	**75.1±3.4**	74.4±0.5
IMDB-M	46.9±2.6	46.2±2.3	46.7±2.7	NA	49.1±1.6	50.0±0.8	**51.1±1.3**	48.0±0.4
MUTAG	86.1±5.2	84.0±6.8	85.6±7.3	87.8±4.2	87.2±5.8	87.2±4.3	**89.9±4.3**	88.8±0.6
PROTEINS	67.5±3.1	65.8±3.3	67.3±3.5	70.9±3.1	70.2±2.1	70.7±1.9	**73.9±1.7**	72.8±0.5

5.3 Ablation Study

To further investigate the effectiveness of the EMP vectorization function in graph representation learning, we have conducted ablation studies of various EMP summaries on the benchmark datasets. In Table 3, we give the performances of different types of EMP summaries (i.e., EMP Silhouette (EMP-S), EMP Entropy (EMP-E), and EMP Betti (EMP-B) using only 0-dimensional topological features (H_0), only 1-dimensional topological features (H_1) and using both ($H_0 + H_1$). Note that in these models, we also included graph-based features (i.e., $f_{\mathcal{V}}$: node features, and $f_{\mathcal{E}}$: edge features; see more details in Section 5.1 and Appendix A.1).

These results suggest that:

- ◇ The choice of the EMP summary can significantly affect the performance (e.g., BZR_MD, DFHR_MD),
- ◇ Using both dimensions ($H_0 + H_1$) may not be better than using only one of the dimensions (e.g., PROTEINS, IMDB-B, REDDIT-5K)
- ◇ The choice of topological dimension (H_0 or H_1) for EMP Summary can be crucial for some datasets (e.g. BZR_MD, REDDIT-5K).

Table 3: Ablation Study. Comparison of the performances of EMP summaries on nine benchmark datasets. The accuracies are given in % (± standard deviation) and **bold numbers** represent the best results.

Model	BZR_MD	COX2_MD	DHFR_MD	MUTAG	PROTEINS	IMDB-B	IMDB-M	REDDIT-B	REDDIT-5K
EMP-B H_0	68.04±2.04	69.39±1.36	72.12±0.94	88.56±0.66	71.54±0.43	73.13±0.44	46.33±0.43	90.52±0.20	**54.41±0.32**
EMP-B H_1	69.22±1.06	**70.12±0.81**	68.77±1.03	86.30±0.72	71.61±0.48	**74.44±0.45**	**48.01±0.42**	88.23±0.28	51.96±0.30
EMP-B $H_0 + H_1$	69.08±1.47	69.27±1.15	74.21±0.91	**88.79±0.63**	71.52±0.53	73.20±0.36	46.82±0.53	**91.03±0.22**	54.34±0.31
EMP-E H_0	65.29±1.75	67.29±1.19	74.53±1.03	86.74±0.68	**72.78±0.54**	73.15±0.63	46.93±0.49	89.95±0.18	53.47±0.36
EMP-E H_1	**77.76±0.95**	67.55±0.85	66.40±0.89	86.57±0.70	71.93±0.54	72.95±0.49	47.03±0.23	88.64±0.16	51.97±0.36
EMP-E $H_0 + H_1$	73.32±1.09	69.18±1.11	74.80±0.96	86.84±0.62	71.86±0.36	72.97±0.58	47.03±0.41	90.05±0.22	53.95±0.35
EMP-S H_0	68.23±0.97	69.07±1.24	78.31±0.68	87.89±0.57	71.51±0.60	73.83±0.39	47.38±0.53	88.42±0.24	52.29±0.19
EMP-S H_1	70.74±1.28	63.72±1.24	75.26±1.05	84.71±0.99	70.74±0.57	73.99±0.55	47.71±0.34	86.96±0.35	50.71±0.28
EMP-S $H_0 + H_1$	72.90±0.91	67.89±1.70	**80.13±0.94**	88.10±0.83	71.61±0.56	74.29±0.28	47.80±0.51	88.59±0.38	52.75±0.22

Limitations: The main limitation of the EMP approach comes with the choice of filtering functions and pairing them. While these choices give flexibility to the model, the best function pairs to use may require domain information of the dataset. One way to bypass this issue is to use self-supervised methods to learn effective node and edge functions from the data.

6 Conclusion

We have introduced an innovative and computationally efficient summary approach for multidimensional persistence across various forms of data, with a specific focus on applications in graph-based machine learning. This novel framework, called Effective Multidimensional Persistence (EMP), offers a practical and effective method to integrate the concept of multidimensional persistence into real-world scenarios. The EMP approach seamlessly integrates with ML models, providing a unified enhancement to existing single persistent summaries. In graph classification tasks, EMP summaries have demonstrated superior performance compared to state-of-the-art techniques across multiple benchmark datasets. Moreover, we have shown that EMP maintains important stability guarantees. This signifies a significant stride in bridging theoretical multipersistence concepts with the machine learning community, thus advancing the utilization of persistent homology in diverse contexts. Looking ahead, our future endeavors aim to enrich the EMP framework by incorporating multiple slicing directions within the multipersistence grid. This involves leveraging deep learning methodologies to effectively combine outputs from various slicing directions.

Acknowledgements

This work was partially supported by the National Science Foundation (NSF) under grants DMS-2202584, DMS-2220613, OAC-1828467, DMS-1925346, CNS-2029661, DMS-2335846/2335847, TIP-2333703, OAC-2115094 Simons Foundation under grant # 579977, a gift from Cisco Inc. and the grant from the Office of Naval Research (ONR) award N00014-21-1-2530. Part of this material is also based upon work supported by (while Y.R.G. was serving at) the NSF. The views expressed in the article do not necessarily represent the views of NSF or ONR.

References

[1] Yuzhou Chen, Yulia Gel, and H Vincent Poor. Time-conditioned dances with simplicial complexes: Zigzag filtration curve based supra-hodge convolution networks for time-series forecasting. *NeurIPS*, 35:8940–8953, 2022.

[2] C. G. Akcora, M. Kantarcioglu, and Y. R. Gel. Blockchain networks: Data structures of Bitcoin, Monero, Zcash, Ethereum, Ripple and Iota. *arXiv:2103.08712*, 2021.

[3] Yuzhou Chen, Ignacio Segovia-Dominguez, Baris Coskunuzer, and Yulia Gel. TAMP-S2GCNets: coupling time-aware multipersistence knowledge representation with spatio-supra graph convolutional networks for time-series forecasting. In *ICLR*, 2021.

[4] Frédéric Chazal and Bertrand Michel. An introduction to topological data analysis: fundamental and practical aspects for data scientists. *Frontiers in Artificial Intelligence*, 4, 2021.

[5] Felix Hensel, Michael Moor, and Bastian Rieck. A survey of topological machine learning methods. *Frontiers in Artificial Intelligence*, 4:52, 2021.

[6] Barbara Giunti. TDA applications library, 2022. https://www.zotero.org/groups/2425412/tda-applications/library.

[7] Magnus Bakke Botnan and Michael Lesnick. An introduction to multiparameter persistence. *arXiv preprint arXiv:2203.14289*, 2022.

[8] Michael Lesnick and Matthew Wright. Interactive visualization of 2-D persistence modules. *arXiv preprint arXiv:1512.00180*, 2015.

[9] Mathieu Carrière and Andrew Blumberg. Multiparameter persistence image for topological machine learning. *NeurIPS*, 33, 2020.

[10] Magnus Bakke Botnan, Steffen Oppermann, and Steve Oudot. Signed barcodes for multiparameter persistence via rank decompositions. In *38th International Symposium on Computational Geometry (SoCG 2022)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2022.

[11] Oliver Vipond. Multiparameter persistence landscapes. *J. Mach. Learn. Res.*, 21:61–1, 2020.

[12] David Loiseaux, Mathieu Carrière, and Andrew J Blumberg. A framework for fast and stable representations of multiparameter persistent homology decompositions. *NeurIPS*, 2023.

[13] David Loiseaux, Luis Scoccola, Mathieu Carrière, Magnus Bakke Botnan, and Steve Oudot. Stable vectorization of multiparameter persistent homology using signed barcodes as measures. *NeurIPS*, 2023.

[14] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. In *ICLR*, 2014.

[15] M. Defferrard, X. Bresson, and P. Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *NeurIPS*, pages 3844–3852, 2016.

[16] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *ICLR*, 2017.

[17] Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. In *ICLR*, 2019.

[18] Sami Abu-El-Haija, Bryan Perozzi, Amol Kapoor, Nazanin Alipourfard, Kristina Lerman, Hrayr Harutyunyan, Greg Ver Steeg, and Aram Galstyan. Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing. In *ICML*, 2019.

[19] Sami Abu-El-Haija, Amol Kapoor, Bryan Perozzi, and Joonseok Lee. N-GCN: Multi-scale graph convolution for semi-supervised node classification. In *Uncertainty in Artificial Intelligence*, pages 841–851, 2020.

[20] Y. Chen, Y.R. Gel, and K. Avrachenkov. LFGCN: Levitating over graphs with levy flights. In *IEEE ICDM*, 2020.

[21] Federico Monti, Karl Otness, and Michael M Bronstein. MotifNet: a motif-based graph convolutional network for directed graphs. In *IEEE DSW*, pages 225–228, 2018.

[22] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *ICLR*, 2018.

[23] Yiding Yang, Xinchao Wang, Mingli Song, Junsong Yuan, and Dacheng Tao. Spagan: shortest path graph attention network. In *International Joint Conference on Artificial Intelligence*, pages 4099–4105, 2019.

[24] Meng Qu, Yoshua Bengio, and Jian Tang. Gmnn: Graph markov neural networks. In *ICML*, pages 5241–5250. PMLR, 2019.

[25] Tamal Krishna Dey and Yusu Wang. *Computational Topology for Data Analysis*. Cambridge University Press, 2022.

[26] Nils M Kriege, Fredrik D Johansson, and Christopher Morris. A survey on graph kernels. *Applied Network Science*, 5(1):1–42, 2020.

[27] Dashti Ali, Aras Asaad, Maria-Jose Jimenez, Vidit Nanda, Eduardo Paluzo-Hidalgo, and Manuel Soriano-Trigueros. A survey of vectorization methods in topological data analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[28] 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.

[29] P. Bubenik. Statistical topological data analysis using persistence landscapes. *JMLR*, 16(1): 77–102, 2015.

[30] Frédéric Chazal, Brittany Terese Fasy, Fabrizio Lecci, Alessandro Rinaldo, and Larry Wasserman. Stochastic convergence of persistence landscapes and silhouettes. In *Proceedings of the thirtieth annual symposium on Computational geometry*, pages 474–483, 2014.

[31] Yu-Min Chung and Austin Lawson. Persistence curves: A canonical framework for summarizing persistence diagrams. *Advances in Computational Mathematics*, 48(1):6, 2022.

[32] Andac Demir, Baris Coskunuzer, Yulia Gel, Ignacio Segovia-Dominguez, Yuzhou Chen, and Bulent Kiziltan. ToDD: Topological compound fingerprinting in computer-aided drug discovery. *NeurIPS*, 35:27978–27993, 2022.

[33] Michael Lesnick and Matthew Wright. Computing minimal presentations and bigraded Betti numbers of 2-parameter persistent homology. *SIAM Journal on Applied Algebra and Geometry*, 6(2):267–298, 2022.

[34] Jeffrey J Sutherland, Lee A O'brien, and Donald F Weaver. Spline-fitting with a genetic algorithm: A method for developing classification structure-activity relationships. *Journal of chemical information and computer sciences*, 43(6):1906–1915, 2003.

[35] Nils Kriege and Petra Mutzel. Subgraph matching kernels for attributed graphs. In *ICML*, pages 291–298, 2012.

[36] Karsten M Borgwardt, Cheng Soon Ong, Stefan Schönauer, SVN Vishwanathan, Alex J Smola, and Hans-Peter Kriegel. Protein function prediction via graph kernels. *Bioinformatics*, 21 (suppl_1):i47–i56, 2005.

[37] Christopher Morris, Nils M Kriege, Kristian Kersting, and Petra Mutzel. Faster kernels for graphs with continuous attributes via hashing. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 1095–1100. IEEE, 2016.

[38] Risi Kondor and Horace Pan. The multiscale laplacian graph kernel. *NeurIPS*, 29, 2016.

[39] Nino Shervashidze, Pascal Schweitzer, Erik Jan Van Leeuwen, Kurt Mehlhorn, and Karsten M Borgwardt. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12(9), 2011.

[40] Nils M Kriege, Pierre-Louis Giscard, and Richard Wilson. On valid optimal assignment kernels and applications to graph classification. *NeurIPS*, 29, 2016.

[41] Leslie O'Bray, Bastian Rieck, and Karsten Borgwardt. Filtration curves for graph representation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1267–1275, 2021.

[42] Federico Errica, Marco Podda, Davide Bacciu, and Alessio Micheli. A fair comparison of graph neural networks for graph classification. In *ICLR*, 2020.

[43] René Corbet, Ulderico Fugacci, Michael Kerber, Claudia Landi, and Bei Wang. A kernel for multi-parameter persistent homology. *Computers & graphics: X*, 2:100005, 2019.

[44] Cheng Xin, Soham Mukherjee, Shreyas N Samaga, and Tamal K Dey. Gril: A 2-parameter persistence based vectorization for machine learning. *arXiv preprint arXiv:2304.04970*, 2023.

[45] Steve Oudot and Luis Scoccola. On the stability of multigraded Betti numbers and hilbert functions. *arXiv preprint arXiv:2112.11901*, 2021.

[46] 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.

[47] Herbert Edelsbrunner and Salman Parsa. On the computational complexity of Betti numbers: reductions from matrix rank. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on discrete algorithms*, pages 152–160. SIAM, 2014.

[48] Mehmet E Aktas, Esra Akbas, and Ahmed El Fatmaoui. Persistence homology of networks: methods and applications. *Applied Network Science*, 4(1):1–28, 2019.

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

[50] David Eisenbud. *Commutative algebra: with a view toward algebraic geometry*, volume 150. Springer Science & Business Media, 2013.

[51] Michael Lesnick. The theory of the interleaving distance on multidimensional persistence modules. *Foundations of Computational Mathematics*, 15(3):613–650, 2015.

[52] Nieves Atienza, Rocío González-Díaz, and Manuel Soriano-Trigueros. On the stability of persistent entropy and new summary functions for topological data analysis. *Pattern Recognition*, 107:107509, 2020.

[53] Megan Johnson and Jae-Hun Jung. Instability of the Betti sequence for persistent homology and a stabilized version of the betti sequence. *Journal of the Korean Society for Industrial and Applied Mathematics*, 25(4):296–311, 2021.