TopoAttn-Nets: Topological Attention in Graph Representation Learning

Yuzhou Chen[1,4], Elena Sizikova[2], Yulia R. Gel[3,5]

[1]Department of Electrical and Computer Engineering, Princeton University
[2]Center for Data Science, New York University
[3]Department of Mathematical Sciences, University of Texas at Dallas
[4]Lawrence Berkeley National Laboratory
[5]National Science Foundation
yc0774@princeton.edu
es5223@nyu.edu
ygl@utdallas.edu

Abstract. Topological characteristics of graphs, that is, properties that are invariant under continuous transformations, have recently emerged as a new alternative form of graph descriptors which tend boost performance of graph neural networks (GNNs) on a wide range of graph learning tasks, from node classification to link prediction. Furthermore, GNNs coupled with such topological information tend to be more robust to attacks and perturbations. However, all prevailing topological methods for GNNs consider a scenario of a fixed learning approach and do not allow for distinguishing between topological noise and topological signatures of the graph which might be the most valuable for the current learning task. To exploit the inherent task-specific topological graph descriptors, we propose a new versatile framework known as Topological Attention Neural Networks (TopoAttn-Nets)[1]. As the first meta-representation of topological knowledge, TopoAttn-Nets employs the attention operation on both local and global data properties and offers their geometric augmentation. We derive theoretical guarantees of the proposed topological learning framework and evaluate TopoAttn-Nets in conjunction with graph classification. TopoAttn-Nets delivers the highest accuracy, outperforming 26 state-of-the-art classifiers on benchmark datasets.

Keywords: Meta-representation · Topological signatures · Representation learning · Graph classification

1 Introduction

Accurately classifying graphs by inferring their geometric and topological properties has recently witnessed an ever increasing interest in many data science applications [6, 10, 33]. In particular, an emerging sub-field of geometric deep learning (GDL) aims to generalize the concept of deep learning (DL) to data in

[1] Our code is available at https://github.com/TopoAttn-Nets/TopoAttn-Nets.git

non-Euclidean spaces by bridging the gap between graph theory and deep neural networks [3]. In turn, many recent studies indicate that integration of topological descriptors, i.e., systematic shape characteristics, into graph learning often results in noticeable performance gains in such tasks as graph classification, link prediction, and anomaly detection [6,12,18,33,46,47]. Furthermore, incorporating the topological signatures into GDL enhances robustness of graph learning to perturbations and attacks. This phenomenon can be explained by important complementary information and deeper insight into the intrinsic graph organizational structure provided by topological data summaries, as compared to conventional non-topological descriptors. Here we aim to further advance topological approaches to graph learning by offering a systematic and versatile framework for extracting the essential *task-specific shape* information.

In particular, topological data analysis (TDA) offers rigorous mathematical tools to explore structural shape properties of the graph-structured data [4,9,14]. Here by shape we broadly understand data properties which are invariant under continuous transformations such as stretching, bending, and twisting. Persistence homology (PH) is a methodology under the TDA framework that analyzes evolution of various patterns in a graph $\mathcal{G}$ as we vary certain user-selected (dis)similarity threshold (i.e., a scale). As such, we can say that PH studies the observed graph $\mathcal{G}$ at multiple resolutions or evaluates its structural properties through multiple lenses. All extracted shape patterns can be then summarized in a form of multi-set in $\mathbb{R}$, known as a persistence diagram (PD). PDs record a type of the topological patterns we detect as well as how long we observe each topological feature as a function of the scale parameter. We are particularly interested in topological features with a longer lifespan, since such features tend to contain valuable information about hidden mechanisms behind graph organization and as such, play a more important role in graph learning. Features with a longer lifespan are said to persist. In turn, features with shorter lifespans are likely to be attributed to topological noise. However, there exists a number of interlinked fundamental challenges on the way of successful integration of topological information into graph learning. The first key problem is how to distinguish important topological features from topological noise [8, 9, 15]. Second, since PDs are point multi-sets, there exists no straightforward approach to combine the extracted topological summaries in a form of PDs with DL models, as DL often requires input data in vector form. As such, there are multiple approaches to make PDs compatible with DL inputs [1, 18, 24]. One of the most popular PD representations allowing for construction of a fully trainable topological layer is adaptively kernelization of PDs. However, existing kernel representations of PDs assume that influence of persistent features on the learning process is *fixed*. Furthermore, typically only a single PD is computed from the graph $\mathcal{G}$–, either upon extracting topological features directly from $\mathcal{G}$, referred to as the *topological domain*, or from the spectral signatures of $\mathcal{G}$, (e.g., Heat Kernel Signatures (HKS) with a single (fixed) diffusion parameter t), referred to as the *spectral domain*. As such, the current kernel representations of PDs do not allow for dis-

tinguishing topological graph characteristics which are *the most valuable for the current learning task*, from topological noise.

New Topological Meta-Representation Paradigm We propose a new flexible and unified framework, TopoAttn-Nets, for meta-representation topological signatures of the graph $\mathcal{G}$ extracted from its PDs. That is, we instill topological signatures from different domains and embed them into meta-representation with attention mechanism which shows an end-to-end learning approach that in turn can be used to learn multiple persistence representations. Furthermore, inspired by the recent meta-learning mechanisms in deep neural networks [20], we combine all kernel-based representation of PDs in various domains into a joint aggregated attention layer, where attention mechanism is used to explicitly encode the structural information of $\mathcal{G}$ from a global perspective. The resulting TopoAttn-Nets represents a trainable, task-specific framework to extract the most informative topological signatures of graph $\mathcal{G}$ from multiple domains in an efficient and provably stable manner.

Contributions. Contrary to all conventional TDA methods in DL where a given task is tackled using a *fixed* learning approach, this paper aims to enhance the topological learning algorithm itself, thereby being the first step toward the paradigm of *topological meta-learning*. The ultimate idea of TopoAttn-Nets is to systematically integrate joint topological features, persistence-based information from multiple domains, and PD transform learning. Specifically, compared to all previous approaches for topological features/kernels/layers, our meta-representation: (1) is not restricted to a particular type of input data and a fixed parametrization map of topological summaries, (2) is more robust to perturbations, (3) allows for learning relationships among topological signatures by providing their geometric augmentation. As a part of the new topological meta-representation, the attention mechanism learns to focus on the most essential topological characteristics of the data and learning algorithms. This is particularly important for web-based data, e.g., usage graphs from social media or other web sources, that exhibit variation at different scales. Capturing both finer scale and larger scale variations using a fixed learning model is challenging. In contrast, TopoAttn-Nets offers a representation that captures both local and global properties, and as a result, improved tractability and generalization performance. Our extensive numerical results indicate that TopoAttn-Nets is competitive in graph classification in comparison to the state of the art: it outperforms 26 top methods in accuracy and is more robust under graph perturbations.

2 Related Work

Kernels for Graph Classification Traditionally, one of the most popular graph classification tools over the past two decades were graph kernel approaches. There is a wide variety of graph kernel frameworks, including marginalized kernel [21], shortest-path kernel [2], graphlet kernel [35], Weisfeiler-Lehman graph kernel [34], and Weisfeiler–Lehman hash graph kernel [29]. These more classical graph-based kernels only consider generating graph level features through

aggregating node representations. While powerful and expressive, the existing kernel-based techniques suffer from limited ability to capture similarities among higher order graph properties of local neighborhoods which in contrast can be inferred from topological structures. To address this limitation, we propose a new flexible topological meta-representation neural network model which coupled with attention mechanism, enables the graph-based learning framework to systematically incorporate higher order graph information both at the local and global levels.

Neural Networks for Graph Classification There generally exist three neural network-based approaches for graph classification: (i) GNN architectures that encode both local graph structure and features of nodes [22, 26, 28, 39, 41], (ii) stable vectorizations of PDs within GNNs [1, 46] or embedding multiple graph filtrations [19], and (iii) kernelization of topological information within GNNs [19,24,45,47]. In contrast, our approach is built upon meta-representation of *multiple* kernelized PDs, that is, choice of topological meta-knowledge to meta-learn. Armed with the proposed meta-representation machinery, we can then exploit the relations between tasks or domains, and learning algorithms.

3 Background on Persistent Homology

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be the observed graph, where $\mathcal{V}$ denotes the set of nodes, $\mathcal{E}$ denotes the set of edges, and $e_{uv} \in \mathcal{E}$ denoting an edge between nodes $u, v \in \mathcal{V}$. The fundamental postulate is to view $\mathcal{G}$ as a sample from some metric space $\mathbb{M}$ whose intrinsic topological structure has been lost due to sampling. Our goal is then to regain knowledge on the lost structural properties of $\mathbb{M}$ via characterizing shape of the observed graph $\mathcal{G}$. The key approach here is to first associate $\mathcal{G}$ with some filtration of $\mathcal{G}$: let $\mathcal{G}_1 \subseteq \mathcal{G}_2 \subseteq \ldots \subseteq \mathcal{G}_k = \mathcal{G}$ be a nested sequence of subgraphs, and let $\mathcal{C}_i$ be the simplicial complex induced by the subgraph $\mathcal{G}_i$ (e.g., clique complex). Then, the nested sequence of these simplicial complexes $\mathcal{C}_1 \subseteq \mathcal{C}_2 \subseteq \ldots \subseteq \mathcal{C}_k$ is called *a filtration* of $\mathcal{G}$. We then can track lifespan of shape characteristics of $\mathcal{G}$ throughout this nested sequence of simplicial complexes. Such shape features include connected components, loops, cavities, and more generally k-dimensional holes. We detect them by means of *a homology*, an algebraic topological invariant. To define the lifespan of a topological feature, we say that the feature is born at $\mathcal{G}_b$ if it does not come from $\mathcal{G}_{b-1}$, and it dies at $\mathcal{G}_d$ ($d \geq b$) if the feature disappears entering $\mathcal{G}_d$ [5]. Hence, its corresponding lifespan, or *persistence* is $d - b$. The resulting persistent homology can be then coded as a multi-set $\mathcal{D}$ of points in $\mathbb{R}^2$, called a PD, with x and y coordinates being the birth and death of each topological feature, respectively. Since $d \geq b$, all points in $\mathcal{D}$ are in the half-space on or above $y = x$. The multiplicity of a point $(b, d) \in \Omega = \{(x, y) \in \mathbb{R}^2 : y > x\}$ is the number of k-dimensional topological features that are born at b and die at d, while points at the diagonal $\Delta = \{(b, b) | b \in \mathbb{R}\}$ have infinite multiplicities. Finally, there exist multiple approaches to construct a filtration of $\mathcal{G}$ [9]. One common method is to use a descriptor function (usually conveys domain information) $f : \mathcal{V} \to \mathbb{R}$ and a sequence of real numbers $a_1 <$

$a_2 < \cdots < a_k$, one can define a nested sequence of subgraphs with $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i)$ where $\mathcal{V}_i = \{v \in \mathcal{V} | f(v) \leq a_i\}$ and $\mathcal{G}_i$ is the induced subgraph of $\mathcal{G}$ by $\mathcal{V}_i$, i.e., $\mathcal{E}_i = \{e_{uv} \in \mathcal{E} | u, v \in \mathcal{V}_i\}$. Similarly, for a weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$ and a sequence of real numbers $a'_1 < a'_2 < \cdots < a'_s$, one can use the weights to define $\mathcal{G}_j = (\mathcal{V}_j, \mathcal{E}_j)$ with $\mathcal{E}_j = \{e_{uv} \in \mathcal{E} | w_{uv} \leq a'_j\}$ and $\mathcal{V}_j = \{v \in \mathcal{V} | e_{uv} \in \mathcal{E}_j\}$.

4 Learnable Topological Meta-Representation for Deep Attention Networks

4.1 Persistence Meta-Representation

In spirit of recent approaches to learnable PD vectorizations [6,18,24], we define an individual representation function s of $\mathcal{D}$ as a composite function of three point transformations in $\mathbb{R}^2$: $s = k \circ \tau_\eta \circ \rho : \Theta \to \big\{ f : \Omega \cup \Delta \to \mathbb{R} \big\}$, where k is a parametrized functional (e.g., the Gaussian kernel) such that $k(x, -\infty) = 0$, $\rho : \mathbb{R}^2 \to \mathbb{R}^2$ is a linear birth-lifetime coordinate transform such that $\rho(x, y) = (x, y - x)$, τ_η is a rationally stretched birth-lifetime, or spike point transform $\tau_\eta : \mathbb{R} \times [0, \infty] \to \mathbb{R} \times (\mathbb{R} \cup \{-\infty\})$, $\eta > 0$, and Θ is a parameter space. Representation of s as a composite function allows us to study PD parametrization over $\mathbb{R}^2$ and, hence, enables a more tractable mathematical formalism and application of a broader range of weighting functions to distinguish topological features in terms of their contribution to the learning task.

Based on the PH framework, we can obtain a set of different representation of topological signatures for the same input graph $\mathcal{G}$ by (i) considering different choices of simplicial complexes, (ii) using different filtering functions, and (iii) defining $\mathcal{G}$ on different domains. Our idea is to harness complementary information from multiple PDs and their learnable representations, hence, capitalizing on the concepts of *meta-analysis*. In particular, here we focus on representation learning of persistence diagrams with respect to two domains. Armed with the set of learnable representations $\boldsymbol{s} = \{s_1, s_2, \ldots, s_{\mathcal{Q}}\}$ and a collection of PDs $\boldsymbol{\mathcal{D}} = \{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_{\mathcal{Q}}\}$, we propose an aggregated, i.e., a meta-representation, of multiple PDs. We first assign each dimension $i \in \{1, \cdots, \mathcal{Q}\}$ a 2-dimensional base representation $s_i(x, y)$ (where (x, y) belongs to $\mathcal{D}_i$) and construct an aggregated representation with n-th order as: $\mathfrak{s}_{agg_n}(x, y) = \underset{1 \leq i_1 < i_2 < \cdots < i_n \leq \mathcal{Q}}{\mathcal{I}} \big[\omega_{i_1, \ldots, i_n} \times \phi(s_{i_1}, \ldots, s_{i_n}) \big]$, where $\mathcal{Q}$ is the dimension of the input space; $\mathcal{I}\big[\,\cdot\,\big]$ refers to the aggregation scheme such as sum and average; function $\phi(\cdot)$ takes multiple base representations as input and outputs to a new representation – *meta-representation*; ω is a weight controlling the effect of corresponding *meta-representation*. For the sake of notation, we omit indices of the base representation as $s_i(\cdot)$. In particular, when $n = \mathcal{Q}$, the $\mathcal{Q}$-th order *meta-representation* can be written of the form: $\mathfrak{s}_{agg_{\mathcal{Q}}}(x, y) = \phi(s_1, \ldots, s_{\mathcal{Q}})$.

To extract topological signatures from a graph $\mathcal{G}$, we can compute persistent homology directly from the observed graph $\mathcal{G}$ and from spectral descriptors of $\mathcal{G}$. The resulting persistence-based summaries contain complementary information

and can be plugged into compatible learning representations via different kernel types.

Option 1. *Spectral domain:* Following [6, 32], we compute $\mathcal{D}$ for graph by replacing original filtration with HKS for fixed diffusion parameter t as the feature function. Given a real-valued function $h(\cdot;\cdot) : \mathbb{R}^2_+ \to \mathbb{R}$, set $h(t;\lambda_k) = e^{-t\lambda_k}$. The HKS $p(\cdot,\cdot) : \mathbb{R}^2 \to \mathbb{R}$ is defined as $p_t(x,y) = \sum_{i=0}^{\infty} e^{-\lambda_i t}\varphi_i(x)\varphi_i(y)$, where λ_i and φ_i are the i-th eigenvalue and the i-th eigenfunction of the Laplace-Beltrami operator, respectively. HKS on a graph $\mathcal{G}$ can be represented as $p : v \to \sum_{i=1}^{n} e^{-\lambda_i t}\varphi_i^2(v)$, where v is a node of $\mathcal{G}$, λ_i and φ_i are eigenvalues and eigenvectors of the normalized graph Laplacian. Since the heat kernel can be viewed as a *low-pass filter*, HKS contains information mainly from low frequencies (and hence higher frequencies are suppressed by increasing t). To capture all the low and high frequencies in $\mathcal{G}$, we use a *meta-representation* to include multiple PDs extracted from HKS with various diffusion parameters.

Option 2. *Topological domain:* To make the model invariant to changes in position and orientation, rotation has been shown to significantly increase classification and segmentation performance [13, 23]. The key operation in the topological domain is to produce transformed training samples of PDs and feed them to the DL model. For a persistence diagram $\mathcal{D}$, rotation augmentations are done by rotating the points on the x- and y-coordinates by θ degrees. In machine learning terminology, these coordinates can be referred to as features. This allows us to characterize the $\mathcal{D}$ generated by each data point as a compact feature vector. The application of rotation augmentation to $\mathcal{D}$ allows us to encode importance of different topological summaries in a vector representation.

Learnable PD Representation in the Topological Domain Let $R_\theta : \mathbb{R}^2 \to \mathbb{R}^2$ be a *rotational* operator for rotation by an angle θ. Applying R_θ to $\mathcal{D}$ results in $R_\theta(\mathcal{D}) = \mathcal{D}_\theta = \big\{\big(\cos(\theta)x + \sin(\theta)y, \cos(\theta)y - \sin(\theta)x\big) \in \mathbb{R}^2 | (x,y) \in \mathcal{D}\big\}$. A PD can be rotated by multiple angles $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_\aleph\}$, $\aleph \geq 2$, where either θ_i is sampled from the uniform distribution $U(0,\pi)$ or $\boldsymbol{\theta}$ is a deterministic sequence of angles. Number of rotated angles $\aleph$ is user-specified to meet computational constraints. This rotational procedure provides a set of candidate latent features for meta-learning [7, 20].

What Are Advantages of the PD Random Rotation? Random rotation of a PD achieves two interlinked goals: (i) improves the extraction of prominent topological information from PDs and (ii) enhances learning the ring of algebraic functions on PDs. On (i), topological features near the diagonal Δ exhibit a higher level of uncertainty but may still contain useful information for classification tasks [27]. Indeed, since we cannot explicitly define how close a feature ought to be to Δ in order to be viewed as topological noise, we aim to extract the signal out of such features under uncertainty. Note that since $\theta \sim U(0,\pi)$, the range of topological feature lifespan in the rotation image $R_\theta(\mathcal{D})$ is $(y-x, -y+x)$. As a result, features with a shorter lifespan in the original unrotated space are stretched in the rotated space and may have a longer lifespan. That is, intuitively, while we still give a higher weight to more persistent features in the original space, upon rotation with a random angle θ, we attempt

to assign topological features whose original lifespan may be shorter due, e.g., to various uncertainties, a chance to contribute to the topological learning. Since $\mathbb{E}[\theta] = \pi/2$ and, hence, the expected rotated lifespan of each topological feature translates to its mean point in the unrotated space, and vice versa, we still incorporate the conventional lifespan characterization of PD. As a result, we extract more signal out of all available topological features than the standard TDA tools (i.e., in (i)), while the attention mechanism mitigates the impact of including the potential topological noise. On (ii), random sampling of θ in the rotation operator R_θ allows us to enrich the set of elements of the affine coordinate ring (i.e., functions on the coordinatized PD space), thereby improving learning of the associated algebraic variety under uncertainties. Such random rotation may be also viewed as a semi-parametric bootstrap of lifespans of each topological feature. To infer potential long-range and periodic relations in the *rotational* transformation of PDs, we propose the generalized locally periodic (GLP) kernel for rotated PDs.

Definition 1. *Let $p_i, l_i, \mu_i, \alpha_i \in \mathbb{R}$, $i = 1, 2$. Then the generalized locally periodic (GLP) kernel is nonnegative function $\mathbb{R}^2 \to \mathbb{R}_+$ is defined as:*

$$\begin{aligned} k_{GLP}(x, y) = \sigma^2 e^{\left\{-2\sin^2\left(\frac{\pi(x-\alpha_1)^2}{p_1}\right) - \frac{(x-\mu_1)^2}{2l_1^2}\right\}} \\ \times e^{\left\{-2\sin^2\left(\frac{\pi(y-\alpha_2)^2}{p_2}\right) - \frac{(y-\mu_2)^2}{2l_2^2}\right\}}. \end{aligned} \tag{1}$$

The advantages of the generalized locally periodic (GLP) kernel are as follows: (i) compared to the Gaussian kernel, it is more appropriate to adopt a periodic kernel that can reflect the similarities between different PDs and (ii) strict periodicity is too rigid (i.e., the purely periodic kernel) since variance exists.

Lemma 1. *The GLP kernel $k_{GLP}(x, y)$ is (a) Lipschitz continuous on $\mathbb{R}^2$, and (b) positive semidefinite.*

Proof. See Appendix A.1.

Furthermore, here we extend the rationally stretched birth-lifetime transform of [18] and consider a generalized spike transform:

$$\tau_\eta^m(x, y) = \begin{cases} (x, y), & y \in [\eta, \infty), \\ (x, \frac{m}{m-1}\eta - \frac{1}{m-1}\frac{\eta^m}{y^{m-1}}), & y \in (0, \eta), \\ (x, -\infty), & y = 0, \end{cases} \tag{2}$$

where $m \in \mathbb{Z}, m \geq 2$.

Lemma 2. *Let $m \in \mathbb{Z}$ and $m \geq 2$, then τ_η is continuous on $\mathbb{R} \times \mathbb{R}_+$ and belongs to a class $\mathcal{C}^1$ of continuously differentiable functions on $\mathbb{R} \times \mathbb{R}_+$.*

Proof. See Appendix A.2.

Armed with Lemmas 1 and 2, we now show the key result needed to derive stability of $s_{ROT} = k_{GLP} \circ \tau_\eta^m \circ \rho$.

Lemma 3. $\lim_{y\to 0}\left|\left(k_{GLP}\circ\tau_\eta\right)_y^{'}\right| < C$ *for* $\mathbb{R}\times[0,\epsilon)$, $C>0$.

Proof. See Appendix A.3.

Lemma 3 implies that $k_{GLP}\circ\tau_\eta$ is Lipschitz continuous and, hence, we can derive stability of rotationally transformed PD representations.

Corollary 1 (Stability of Rotationally Transformed PD Representations). *Following the rotational operator procedure, let* $\mathcal{D}_{\theta_1}$ *and* $\mathcal{D}_{\theta_2}$ *be two rotated persistence diagrams by two angles (i.e.,* θ_1, θ_2*) and let* $s_{ROT}=k_{GLP}\circ\tau_\eta^m\circ\rho$ *where* τ_η^m *is defined by (2) and* $\rho:\Omega\cup\Delta\to\mathbb{R}\times\mathbb{R}_{\geq 0}$. *Then* $\left|s_{ROT}(\mathcal{D}_{\theta_1})-s_{ROT}(\mathcal{D}_{\theta_2})\right|\leq CW_1^q(\mathcal{D}_{\theta_1},\mathcal{D}_{\theta_2})$, *where* $C>0$ *and*

$$W_1^q(\mathcal{D}_{\theta_1},\mathcal{D}_{\theta_2})=\inf_{\gamma}\Big(\sum_{x\in\mathcal{D}_{\theta_1}}\|x-\gamma(x)\|_q\Big)$$

is 1-Wasserstein distance with $q\in\mathbb{Z}^+$, γ *ranging over all bijections between* $\mathcal{D}_{\theta_1}\cup\Delta$ *and* $\mathcal{D}_{\theta_2}\cup\Delta$, *and* $||z||_\infty=\max_i|z_i|$.

Proof. See Appendix A.4.

Learnable PD Representation in the Spectral Domain Based on the multi-scale property of the heat kernel, for small values of t, the function $p_i(t)$ is mainly determined by small neighborhoods of node i, and heat diffuses to larger and larger neighborhoods as t increases. This means $p_i(t)$ can capture both local and global information from the view point of node i when varying t. Let $\mathcal{D}_t$ be a PD obtained from graph $\mathcal{G}$ by using the multiscale heat kernel $p(t)$ with diffusion parameter t. Similar to [17,24,32], we consider the Gaussian-based kernel as a representation for PD, but we utilize a higher-order Gaussian kernel which can be beneficial for better distinguishing topological signals from topological noise [36].

Definition 2. *Let* $\boldsymbol{\mu}=(\mu_1,\mu_2)^\top\in\mathbb{R}^2$, $\boldsymbol{\sigma}=(\sigma_1,\sigma_2)\in\mathbb{R}^2$, *and* $\boldsymbol{\rho}=(\rho_1,\rho_2)\in\mathbb{R}_+^2$. *We define the higher-order Gaussian (HOG) kernel through the following equation:*

$$k_{HOG}(x,y)=e^{\left(-\left(\frac{(x-\mu_1)^2}{\sigma_1^2}\right)^{\rho_1}-\left(\frac{(y-\mu_2)^2}{\sigma_2^2}\right)^{\rho_2}\right)}. \tag{3}$$

Note that $k_{HOG}(x,y)$ *belongs to class* $\mathcal{C}^\infty(\mathbb{R}^2)$ *and is Lipschitz continuous on* $\mathbb{R}^2$.

Similar to s_{ROT}, we derive the following theoretical properties on the learnable PD representation in the spectral domain, i.e., Lipschitz continuity in Lemma 4 and stability of the PD representation using the HOG kernel.

Lemma 4. $\lim_{y\to 0}\left|\left(k_{HOG}\circ\tau_\eta\right)_y^{'}\right| < C$ *for* $\mathbb{R}\times[0,\epsilon)$, $C>0$.

Proof. See Appendix A.5.

Corollary 2 (Stability of PD Representations in the Spectral Domain). *Let $\mathcal{D}_{t_1}$ and $\mathcal{D}_{t_2}$ be two persistence diagrams over two diffusion parameters (e.g., t_1, t_2) and let $s_{TOP} = k_{HOG} \circ \tau_\eta^m \circ \rho$, where τ_η^m is defined by (2) and $\rho : \Omega \cup \Delta \to \mathbb{R} \times \mathbb{R}_{\geq 0}$. Then $\big|s_{TOP}(\mathcal{D}_{t_1}) - s_{TOP}(\mathcal{D}_{t_2})\big| \leq CW_1^q(\mathcal{D}_{t_1}, \mathcal{D}_{t_2})$.*

Proof. See Appendix A.6.

Persistence-Based Weight Mechanism Recall that points $d = (x, y) \in \mathcal{D}$ with a longer persistence $(y - x)$ are likelier to contain intrinsic structural information on the graph $\mathcal{G}$, while points with shorter persistence tend to be topological noise [**?**]. As such, assigning a higher weight to more persistent points in $\mathcal{D}$ tends to improve classification performance. Here we consider a weighting function $\mathcal{F}(x, y) = \arctan\big(C((y - x))^\zeta\big)$, where $C > 0$ and $\zeta \in \mathbb{Z}^+$.

Theorem 1 (Stability of the Weighted Kernel Embedding). *Let $\mathcal{D}_1$ and $\mathcal{D}_2$ be two persistence diagrams. Let $h(x, y) = \mathcal{F}(x, y)s(x, y)$, where $\mathcal{F}(x, y) = arctan\big(C((y - x))^\zeta\big)$, $C > 0$ and $\zeta \in \mathbb{Z}^+$, and $s : \Omega \bigcup \Delta \to \mathbb{R}$ where s is either s_{ROT} (1) or s_{TOP} (2). Then, for $\zeta = 1$,*

$$|| \sum_{(x,y)\in\mathcal{D}_1} h(x, y) - \sum_{(x',y')\in\mathcal{D}_2} h(x', y')|| \leq CW_1^q(\mathcal{D}_1, \mathcal{D}_2).$$

Proof. See Appendix A.7.

4.2 Aggregated Attention Layer

We now proceed to construction of TopoAttn-Nets. First, note that HKS at lower and higher values of t capture high- and low-frequency information, respectively. Since higher frequencies are more sensitive to changes of t than lower frequencies, in a bid to capture the global and local information of input graph $\mathcal{G}$, we propose a new model, TopoAttn-Nets, that can learn relationships between spectral and geometric information, including mixing feature representations of different frequencies and transformations. As discussed earlier, aggregated representations in machine learning constitute a powerful architecture allowing for automatic combination of multi-source information. Contrary to [18, 24, 32], all key constituents in the proposed TopoAttn-Nets framework – kernel locations, kernel lengths, kernel scales, and the stretched parameter (i.e., parameters defined for a *meta-representation*) are learnable during training. For any domain, we use $\boldsymbol{\mathcal{D}}_{\vartheta_i} = \{\mathcal{D}_{\vartheta_i}^1, \cdots, \mathcal{D}_{\vartheta_i}^N\}$, where $i = \{1, 2, \cdots, I\}$ and N is the number of PDs, to represent a set of PDs over HKS diffusion scale t_i (i.e., $\vartheta_i \leftarrow t_i$) or rotation angle θ_i (i.e., $\vartheta_i \leftarrow \theta_i$). Finally, the TopoAttn-Nets can be formulated as:

$$H^{(l+1)} = \begin{cases} \oplus_i \sigma(\alpha_i s(\boldsymbol{\mathcal{D}}_{\vartheta_i}) \cdot \Theta_i^{(l)}), & \text{1st-order} \\ \oplus_{\substack{i\neq j \\ i<j}} \sigma\big(\alpha_{ij}\big[s(\boldsymbol{\mathcal{D}}_{\vartheta_i}); s(\boldsymbol{\mathcal{D}}_{\vartheta_j})\big]\Theta_{ij}^{(l)}\big), & \text{2nd-order} \end{cases} \tag{4}$$

where $\oplus$ denotes concatenation of vectors, $H^{(l+1)}$ is the first-order feature vector, $\Theta_i^{(l)}$ and $\Theta_{ij}^{(l)}$ are trainable weights in the layer, and $\sigma(\cdot)$ is the activation function, e.g., $\mathrm{ReLU}(\cdot) = \max(0, \cdot)$. Notice that function $s(\cdot)$ is either $s_{ROT}(\cdot)$ or $s_{TOP}(\cdot)$, which depends on the type of $\boldsymbol{\mathcal{D}}_{\vartheta_i}$. To make learnable weights comparable across different components, we normalize them by a softmax operation. That is, (i) 1st-order: $\alpha_i = \exp(\omega_i) / \sum_i \exp(\omega_i)$, where $\omega_i = \mathrm{diag}\big(\mathfrak{F}(\mathcal{D}^1_{\vartheta_i}), \cdots, \mathfrak{F}(\mathcal{D}^N_{\vartheta_i})\big)$; (ii) 2nd-order: $\alpha_{ij} = \exp(\omega_{ij}) / \sum_j \exp(\omega_{ij})$, where $\omega_{ij} = \mathrm{diag}\big(\sum_{\kappa=1}^{2} \mathfrak{F}(\mathcal{D}^\kappa_{\vartheta_i}), \cdots, \sum_{\kappa=N-1}^{N} \mathfrak{F}(\mathcal{D}^\kappa_{\vartheta_i})\big)$ and $\mathfrak{F}(\mathcal{D}^\kappa_{\vartheta_i}) = \big(\mathcal{F}(x_1, y_1)^\kappa_{\vartheta_i}, \mathcal{F}(x_2, y_2)^\kappa_{\vartheta_i}, \cdots, \mathcal{F}(x_m, y_m)^\kappa_{\vartheta_i}\big)$ (where $\mathfrak{F}(\mathcal{D}^\kappa_{\vartheta_i})$ is the arctangent function for k-th PD $\mathcal{D}^\kappa_{\vartheta_i}$ and $\mathcal{F}(x, y) = \arctan\big(C((y-x))^\zeta\big)$ (every point $(x, y) \in \mathcal{D}^\kappa_{\vartheta_i}$, $C > 0$, $\zeta \in \mathbb{Z}^+$)). Here m is the number of points in $\mathcal{D}^\kappa_{\vartheta_i}$. The relative architectures of the feature vectors based on HKS at various diffusion parameters with the A-th order and rotation by different angles with the B-th order can be written as $H^{(l+1)}_{\mathrm{hks}_A}$ and $H^{(l+1)}_{\mathrm{rot}_B}$, respectively (where $A, B \in \{1, 2\}$). We can now rewrite the output $Z^{l+1} = \{H^{(l+1)}_{\mathrm{hks}_A}, H^{(l+1)}_{\mathrm{rot}_B}\}$ of the TopoAttn-Nets using column-wise concatenation as $Z^{(l+1)} = \oplus_j H^{(l+1)}_j$, where $j \in \{\mathrm{hks}_A, \mathrm{rot}_B\}$.

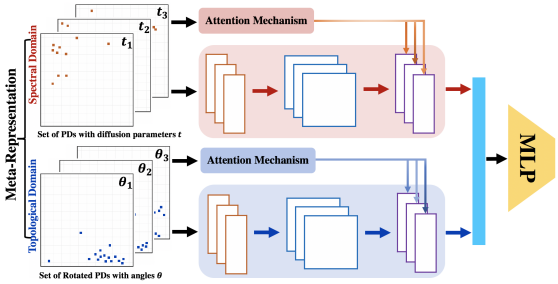

Fig. 1. Architecture of TopoAttn-Nets. A detailed description is given in Appendix C.

5 Experiments

For graph classification, we validate our method on the following standard graph benchmarks: (i) biological frameworks MUTAG and PTC, where nodes represent mutable and carcinogenic molecules, (ii) internet movie collaborations IMDB-B and IMDB-M, where nodes are actors/actresses and edges are common movie occurrences, and (iii) Reddit (an online aggregation and discussion website) discussion threads REDDIT-5K and REDDIT-12K, 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 the split setting of [18], that is, a 90/10 random training/test split. Furthermore, we perform a one-sided two-sample t-test between the best result and

Table 1. Performance summary (accuracy with standard deviation) on the graph classification tasks.

Method	MUTAG	PTC	IMDB-B	IMDB-M	REDDIT-5K	REDDIT-12K
GK [35]	83.5 (0.6)	59.2 (0.5)	65.9 (0.3)	43.9 (0.4)	41.0 (0.2)	31.8 (0.1)
RetGK [44]	90.3 (1.1)	62.5 (1.6)	71.9 (1.0)	47.7 (0.3)	56.1 (0.5)	**48.7** (**0.2**)
DGK [42]	87.4 (2.7)	60.1 (2.5)	67.0 (0.6)	44.6 (0.4)	41.3 (0.2)	32.2 (0.1)
RF [17]	89.0 (3.8)	61.5 (2.7)	71.5 (0.8)	50.7 (0.7)	50.9 (0.3)	42.7 (0.3)
WL [34]	84.4 (1.5)	55.4 (1.5)	70.8 (0.5)	49.8 (0.5)	51.2 (0.3)	32.6 (0.3)
Deep-WL [42]	82.9 (2.7)	60.1 (2.5)	-	-	-	-
WWL [37]	87.3 (1.5)	66.3 (1.2)	-	-	-	-
P-WL [33]	86.3 (1.4)	63.1 (1.7)	72.8 (0.5)	-	-	-
P-WL-C [33]	90.5 (1.3)	64.0 (0.8)	73.2 (0.8)	-	-	-
P-WL-UC [33]	85.2 (0.3)	63.5 (1.6)	73.0 (1.0)	-	-	-
PF [25]	85.6 (1.7)	62.4 (1.8)	71.2 (1.0)	48.6 (0.7)	56.2 (1.1)	47.6 (0.5)
WKPI [45]	88.3 (2.6)	68.1 (2.4)	75.1 (1.1)	49.5 (0.4)	59.5 (0.6)	48.4 (0.5)
TopoGNN [19]	-	-	72.0 (2.3)	-	-	-
$\text{TopoGNN}_{(\text{stat})}$ [19]	-	-	72.8 (5.4)	-	-	-
sPBoW [47]	-	-	-	-	45.6 (5.4)	31.6 (2.8)
$\text{PI}_{(\text{NN})}$ [18]	89.8 (2.5)	63.5 (2.6)	71.2 (2.5)	48.8 (2.8)	46.7 (0.5)	35.1 (0.5)
$\text{Essential}_{(\text{NN})}$ [18]	90.0 (1.7)	63.0 (2.3)	73.5 (2.0)	52.0 (1.8)	54.5 (0.6)	44.5 (0.4)
DGCNN [43]	85.8 (5.5)	58.6 (7.1)	70.0 (0.8)	47.8 (3.4)	48.7 (4.5)	-
GAT [38]	87.4 (5.3)	63.7 (8.2)	72.3 (5.1)	50.1 (3.6)	57.2 (2.2)	-
GraphSAGE [16]	85.7 (4.7)	63.9 (7.7)	72.3 (5.3)	50.9 (2.2)	-	-
CapsGNN [40]	86.7 (6.9)	66.0 (5.9)	71.7 (3.4)	48.5 (4.1)	52.9 (2.2)	-
PSCN [30]	89.0 (4.4)	62.3 (5.7)	71.0 (2.3)	45.2 (2.8)	49.1 (0.7)	41.3 (0.4)
GIN [41]	90.0 (8.8)	66.6 (6.9)	75.1 (5.1)	52.3 (2.8)	57.5 (1.5)	-
GCN [22]	85.6 (5.8)	64.2 (4.3)	74.0 (3.4)	51.9 (3.8)	56.7 (1.7)	-
PersLay [6]	89.8 (1.5)	-	71.2 (2.5)	48.8 (1.0)	55.6 (1.1)	47.7 (0.9)
FC [31]	87.3 (0.7)	65.1 (3.9)	73.8 (0.4)	46.8 (0.4)	52.4 (0.4)	-
TopoAttn-Nets (ours)	*****92.4** (**1.5**)	**68.3** (**5.1**)	**75.2** (**2.1**)	*****54.2** (**0.6**)	**59.5** (**0.5**)	45.0 (0.5)

the best performance achieved by the runner-up, where *, **, *** denote significant, statistically significant, highly statistically significant results, respectively. The statistics of data we used in the Experiments section are summarized in Appendix B, Table 1.

Baselines For graph classification, we perform an expansive evaluation the performance of TopoAttn-Nets with respect to the 26 most recent state-of-the-art (SOA) approaches: (i) graph kernel-based approaches: graphlet kernel (GK) [35], deep graphlet kernel (DGK) [42], Weisfeiler-Lehman kernel (WL) [34], deep variant of subtree features (Deep-WL) [42], graph-feature + random forest approach (RF) [17], Wasserstein Weisfeiler–Lehman (WWL) [37], probability-based graph kernel (RetGK) [44], persistent Weisfeiler-Lehman kernels (P-WL, P-WL-C, P-WL-UC) [33], and Persistence Fisher kernel (PF) [25]; (ii) topological information in kernel-based methods: Stable Persistence Bag of Words (sP-BoW) [47], weighted-kernel for persistence images (WKPI) [45], and Filtration Curves (FC) [31]; (iii) graph neural networks: PATCHYSAN (PSCN) [30], Graph

Convolutional Network (GCN) [22], Graph attention networks (GAT) [38], GraphSAGE [16], Deep Graph CNN (DGCNN) [43], Graph Isomorphism Network (GIN) [41], and Capsule Graph Neural Network (CapsGNN) [40]; (iv) topological-based deep neural networks: persistence images (PI) combined with a convolutional neural network [17, 18], essential features (Essential) combined with a convolutional neural network [18], GNN augmented with global graph persistence yielded from multiple filtrations (TopoGNN) [19], and the generic neural network layer for persistence diagrams (PersLay) [6].

Parameters Setting In our experiments, We adopt the Adam optimizer for our TopoAttn-Nets model training with an initial learning rate $lr = 1 \times 10^{-3}$. We fix the number of training epoch to 500 for all datasets. We train the model using early stopping with a window size of 200. To prevent over-fitting, we use 1×10^{-4} L_2 regularization on the weights, and dropout input and hidden layers. To analyze behavior of HKS, i.e., $p(t,x) = \sum_{i=0}^{\infty} e^{-\lambda_i t}\varphi_i(x)^2$ (where λ_i and $\varphi_i(x)$ are the i-th eigenvalue and the i-th eigenfunction of the Laplace-Beltrami operator, respectively) under different time values t and to capture all of the information contained in the heat kernel, we set $t = \{0.1, 1, 5, 10, 50, 100, 150, 200, 1000\}$. We then conduct a random combination method to determine the best combination of local and global information. For topological signature rotation, the rotation could be implemented by infinite angles among the range $[0°, 180°]$. To avoid repetition and redundancy, we rotate topological signatures at the set of angles $\boldsymbol{\theta}$, i.e., $\boldsymbol{\theta} = [0°, 30°, 45°, 60°, 90°, 120°, 135°, 150°, 180°]$ and find an optimal combination of topological information through random combination method. Since points near the diagonal in the persistence diagram $\mathcal{D}$ have shorter lifetimes (i.e., $y - x$) and are considered "topological noise", we determine the number of persistent pairs for model training through $argsort(f(\mathcal{D}))[-num\_pairs :]$, where $f(\mathcal{D}) = (y_1 - x_1, y_2 - x_2, \cdots, y_m - x_m)$ and num_pairs is the minimum number of persistent pairs in PDs $\{\mathcal{D}_1, \cdots, \mathcal{D}_N\}$ for each graph in the dataset.

Table 2. Analysis of kernel hyperparameters, attention mechanism, numbers of PDs as input, and rotation angles. Classification accuracy (st. dev.) on IMDB-B.

	k_{HOG}	
Kernel	$\rho = 1.0$	$\rho = 2.0$
	72.0 (3.4)	**75.2 (2.1)**
	Attention mechanism	
	W/o Attn	**With Attn**
Framework	74.0 (3.7)	**76.2 (2.1)**
	The number of PDs	
	1 PD	**3 PDs**
	71.0 (2.2)	**75.2 (2.1)**
	Rotation angles	
Rotation	$\theta = 45°$	$\theta = 90°$
	71.1 (1.1)	**71.2 (4.6)**

Graph Classification Table 1 reports results of mean accuracy and standard deviation across all models tested. The proposed model outperforms 26 SOAs on 5 benchmark datasets, except for REDDIT-12K. Compared to baseline methods, which extract PDs from only one domain, TopoAttn-Nets combines multi-frequencies and topological information across different domains in a single framework. RetGK outperforms our proposed model on the REDDIT-12K

dataset may be due to REDDIT-12K has the weakest structural information, i.e., with very few links per node on average (its average density $\approx 2 \times 10^{-6}$ which is too sparse to deliver sufficient information on higher-order topological properties. In addition, for attributed graphs (i.e., MUTAG and PTC), TopoAttn-Nets still outperforms GCN-based approaches which use additional node features/labels, because kernel-based *meta-representation* equipped with neural network architecture can extract aggregated information from different scales that greatly benefits graph classification tasks.

Ablation Study To better evaluate the performance of TopoAttn-Nets, we conduct a comprehensive ablation study on IMDB-B (see Table 2) by testing (i) kernel hyperparameters, (ii) attention mechanism (Attn), (iii) the number of PDs as input to our TopoAttn-Nets model, and (iv) rotation angle. The performances of TopoAttn-Nets with different (kernel) hyperparameters indicate that kernel hyperparameters enable control the effect of persistence, i.e., extracting meaningful information via a good approximation of the kernel. The comparison between with and without attention mechanism shows that adding attention mechanism can help capture importance of different PDs. Examining the results of different PDs as input, we can observe that a large improvement brought by applying multiple PDs to the input of TopoAttn-Nets. Comparison among different rotation angles underscores contribution of rotations to variability.

Table 3. Learned attention weights α_{hks} and α_{rot} of TopoAttn-Nets for *multi-frequency* and *topological* features.

Dataset	**Learned value**	
Attention weights	$\boldsymbol{\alpha}_{\text{hks}}$	$\boldsymbol{\alpha}_{\text{rot}}$
IMDB-B	**0.53**	0.47
IMDB-M	**0.60**	0.40
REDDIT-5K	0.42	**0.58**
REDDIT-12K	0.32	**0.68**

Sensitivity and Robustness We evaluate robustness of TopoAttn-Nets w.r.t. adversarial attacks on REDDIT-5K. Here we consider graph structural perturbations of [48]) and present a comparison against two runner-ups which are the closest competitors of TopoAttn-Nets, namely, WKPI [45] and GIN [41]. Table 4 indicates that TopoAttn-Nets outperforms SOAs both in terms of accuracy and standard deviation under all attacks. Hence, TopoAttn-Nets may be viewed as the most reliable and accurate alternative under perturbations.

Table 4. Classification accuracy (st. dev.) under adversarial attack on REDDIT-5K.

Method	**Perturbation Rate**			
	0%	**5%**	**10%**	**15%**
WKPI [45]	59.5 (0.6)	51.3 (3.3)	50.5 (2.2)	50.0 (2.0)
GIN [41]	57.5 (1.5)	51.2 (3.5)	49.0 (1.5)	47.7 (1.6)
TopoAttn-Nets	**59.5 (0.5)**	**51.9 (2.9)**	**51.2 (2.3)**	**50.1 (1.4)**

Relative Importance of Features Table 3 reports the TopoAttn-Nets learned attention weights. Interestingly, we notice the attention weight of the

multi-frequency feature is larger than that of *topological* feature for smaller graphs (i.e., biological and internet movie collaboration graphs). That is, the attention component reveals the relative importance of intrinsic finer- or coarser-grain variability in the data shape. For example, in learning tasks for sparser graphs, local variability often tends to be the key factor. Table 3 shows that indeed topological features addressing finer-grain shape properties of very sparse REDDIT-5K and REDDIT-12K, with average diameters of 11.96 and 10.91 and densities of 0.90 and 1.79, respectively, tend to be more valuable for classification. This also implies that importance of multi-frequency or topological information might depend more on the graph size rather than the specific type of data.

Computational Costs Complexity of computing distances among PDs is $\mathcal{O}(m^3)$, where m is the number of points. All experiments are compiled and tested on a Tesla V100-SXM2-16GB GPU. Table 5 reports average running time to generate PDs and mean training time per epoch of TopoAttn-Nets on IMDB-B and REDDIT-5K, respectively.

Table 5. Complexity of TopoAttn-Nets: average time (in sec). to generate PD and training time per epoch.

Dataset	**Avg. points in PD**	**Avg. Time Taken**	
		PD generation	**Train per epoch**
IMDB-B	84.51	6×10^{-3}	1.15
REDDIT-5K	521.35	5×10^{-1}	0.53

6 Conclusion

We have developed a new flexible framework for meta-representation of persistence information in graphs, which may be viewed as the first step toward topological meta-learning on graphs. We have derived stability guarantees of the proposed approach and assessed its robustness to perturbations. The exhaustive experimental validation has indicated high competitiveness of the proposed meta-representation ideas in respect to the benchmarks. Future research include multiple directions. First, we will explore few shot concepts for topological meta-learning on graphs. Second, we will investigate utility of topological meta-representation for link prediction. Third, we will explore the proposed meta-representation and attention ideas in conjunction with multiparameter persistence [11] and local topological algorithms [10, 46].

Acknowledgements

This work is sponsored by the National Science Foundation under award numbers ECCS 2039701, INTERN supplement for ECCS 1824716, DMS 1925346 and

the Department of the Navy, Office of Naval Research under ONR award number N00014-21-1-2530. Part of this material is also based upon work supported by (while serving at) the National Science Foundation. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation and/or the Office of Naval Research. The authors are grateful to Baris Coskunuzer for insightful discussions.

References

1. Adams, H., Emerson, T., Kirby, M., Neville, R., Peterson, C., Shipman, P., Chepushtanova, S., Hanson, E., Motta, F., Ziegelmeier, L.: Persistence images: A stable vector representation of persistent homology. JMLR **18**(1), 218–252 (2017)
2. Borgwardt, K.M., Kriegel, H.P.: Shortest-path kernels on graphs. In: ICDM (2005)
3. Bronstein, M.M., Bruna, J., LeCun, Y., Szlam, A., Vandergheynst, P.: Geometric deep learning: going beyond euclidean data. IEEE Signal Process. Mag. **34** (2017)
4. Carlsson, G.: Topology and data. BAMS **46**(2), 255–308 (2009)
5. Carlsson, G.: Topological pattern recognition for point cloud data. Acta Numerica **23**, 289–368 (2014)
6. Carrière, M., Chazal, F., Ike, Y., Lacombe, T., Royer, M., Umeda, Y.: Perslay: A simple and versatile neural network layer for persistence diagrams. In: AISTATS (2020)
7. Charles, C.K., Taylor, C., Keller, J.: Meta-analysis: From data characterisation for meta-learning to meta-regression. In: PKDD Workshop on data mining, decision support, meta-learning and ILP (2000)
8. Chazal, F., Fasy, B., Lecci, F., Michel, B., Rinaldo, A., Wasserman, L.: Robust topological inference: Distance to a measure and kernel distance. JMLR **18** (2017)
9. Chazal, F., Michel, B.: An introduction to topological data analysis: fundamental and practical aspects for data scientists. Frontiers in Artificial Intelligence **4** (2021)
10. Chen, Y., Coskunuzer, B., Gel, Y.: Topological relational learning on graphs. In: NeurIPS. vol. 34, pp. 27029–27042 (2021)
11. Chen, Y., Segovia-Dominguez, I., Coskunuzer, B., Gel, Y.: TAMP-S2GCNets: coupling time-aware multipersistence knowledge representation with spatio-supra graph convolutional networks for time-series forecasting. In: ICLR (2022)
12. Chen, Y., Segovia-Dominguez, I., Gel, Y.R.: Z-GCNETs: Time zigzags at graph convolutional networks for time series forecasting. In: ICML (2021)
13. Dieleman, S., Willett, K.W., Dambre, J.: Rotation-invariant convolutional neural networks for galaxy morphology prediction. MNRAS **450**(2), 1441–1459 (2015)
14. Edelsbrunner, H., Letscher, D., Zomorodian, A.: Topological persistence and simplification. Discrete Comput. Geom. **28**, 511–533 (2002)
15. Fasy, B.T., Lecci, F., Rinaldo, A., Wasserman, L., Balakrishnan, S., Singh, A.: Confidence sets for persistence diagrams. AoS **42**(6), 2301–2339 (2014)
16. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: NeurIPS. pp. 1024–1034 (2017)
17. Hofer, C., Kwitt, R., Niethammer, M., Uhl, A.: Deep learning with topological signatures. In: NeurIPS. pp. 1634–1644 (2017)
18. Hofer, C.D., Kwitt, R., Niethammer, M.: Learning representations of persistence barcodes. JMLR **20**(126), 1–45 (2019)

19. Horn, M., De Brouwer, E., Moor, M., Moreau, Y., Rieck, B., Borgwardt, K.: Topological graph neural networks. In: ICLR (2022)
20. Hospedales, T., Antoniou, A., Micaelli, P., Storkey, A.: Meta-learning in neural networks: A survey. arXiv:2004.05439 (2020)
21. Kashima, H., Tsuda, K., Inokuchi, A.: Marginalized kernels between labeled graphs. In: ICML. pp. 321–328 (2003)
22. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: ICLR (2017)
23. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NeurIPS. pp. 1097–1105 (2012)
24. Kusano, G., Fukumizu, K., Hiraoka, Y.: Kernel method for persistence diagrams via kernel embedding and weight factor. JMLR **18** (2017)
25. Le, T., Yamada, M.: Persistence fisher kernel: A Riemannian manifold kernel for persistence diagrams. In: NeurIPS. pp. 10007–10018 (2018)
26. Levie, R., Monti, F., Bresson, X., Bronstein, M.M.: Cayleynets: Graph convolutional neural networks with complex rational spectral filters. IEEE Signal Process. Mag. **67**(1), 97–109 (2018)
27. Maroulas, V., Mike, J.L., Oballe, C.: Nonparametric estimation of probability density functions of random persistence diagrams. JMLR **20**(151), 1–49 (2019)
28. Monti, F., Boscaini, D., Masci, J., Rodola, E., Svoboda, J., Bronstein, M.M.: Geometric deep learning on graphs and manifolds using mixture model CNNs. In: CVPR. pp. 5115–5124 (2017)
29. Morris, C., Kriege, N.M., Kersting, K., Mutzel, P.: Faster kernels for graphs with continuous attributes via hashing. In: IEEE ICDM. pp. 1095–1100 (2016)
30. Niepert, M., Ahmed, M., Kutzkov, K.: Learning convolutional neural networks for graphs. In: ICML. pp. 2014–2023 (2016)
31. O'Bray, L., Rieck, B., Borgwardt, K.: Filtration curves for graph representation. In: ACM SIGKDD. pp. 1267–1275 (2021)
32. Reininghaus, J., Huber, S., Bauer, U., Kwitt, R.: A stable multi-scale kernel for topological machine learning. In: CVPR. pp. 4741–4748 (2015)
33. Rieck, B., Bock, C., Borgwardt, K.: A persistent Weisfeiler-Lehman procedure for graph classification. In: ICML. pp. 5448–5458 (2019)
34. Shervashidze, N., Schweitzer, P., Van Leeuwen, E.J., Mehlhorn, K., Borgwardt, K.M.: Weisfeiler-Lehman graph kernels. JMLR **12**(77), 2539–2561 (2011)
35. Shervashidze, N., Vishwanathan, S., Petri, T., Mehlhorn, K., Borgwardt, K.: Efficient graphlet kernels for large graph comparison. In: AISTATS. pp. 488–495 (2009)
36. Tashev, I., Acero, A.: Statistical modeling of the speech signal. In: IWAENC (2010)
37. Togninalli, M., Ghisu, E., Llinares-López, F., Rieck, B., Borgwardt, K.: Wasserstein Weisfeiler-Lehman graph kernels. In: NeurIPS. pp. 6436–6446 (2019)
38. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. In: ICLR (2018)
39. Verma, S., Zhang, Z.L.: Hunt for the unique, stable, sparse and fast feature learning on graphs. In: NeurIPS. pp. 88–98 (2017)
40. Xinyi, Z., Chen, L.: Capsule graph neural network. In: ICLR (2018)
41. Xu, K., Hu, W., Leskovec, J., Jegelka, S.: How powerful are graph neural networks? In: ICLR (2019)
42. Yanardag, P., Vishwanathan, S.: Deep graph kernels. In: ACM SIGKDD. pp. 1365–1374 (2015)
43. Zhang, M., Cui, Z., Neumann, M., Chen, Y.: An end-to-end deep learning architecture for graph classification. In: AAAI (2018)

44. Zhang, Z., Wang, M., Xiang, Y., Huang, Y., Nehorai, A.: Retgk: Graph kernels based on return probabilities of random walks. In: NeurIPS. pp. 3964–3974 (2018)
45. Zhao, Q., Wang, Y.: Learning metrics for persistence-based summaries and applications for graph classification. In: NeurIPS. pp. 9855–9866 (2019)
46. Zhao, Q., Ye, Z., Chen, C., Wang, Y.: Persistence enhanced graph neural network. In: AISTATS. pp. 2896–2906 (2020)
47. Zieliński, B., Lipiński, M., Juda, M., Zeppelzauer, M., Dłotko, P.: Persistence bag-of-words for topological data analysis. In: IJCAI. pp. 4489–4495 (2019)
48. Zügner, D., Akbarnejad, A., Günnemann, S.: Adversarial attacks on neural networks for graph data. In: ACM SIGKDD. pp. 2847–2856 (2018)