TopoGCL: Topological Graph Contrastive Learning

Yuzhou Chen[1], Jose Frias[2], Yulia R. Gel[3,4]

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

Abstract

Graph contrastive learning (GCL) has recently emerged as a new concept which allows for capitalizing on the strengths of graph neural networks (GNNs) to learn rich representations in a wide variety of applications which involve abundant unlabeled information. However, existing GCL approaches largely tend to overlook the important latent information on higher-order graph substructures. We address this limitation by introducing the concepts of topological invariance and extended persistence on graphs to GCL. In particular, we propose a new contrastive mode which targets topological representations of the two augmented views from the same graph, yielded by extracting latent shape properties of the graph at multiple resolutions. Along with the extended topological layer, we introduce a new extended persistence summary, namely, extended persistence landscapes (EPL) and derive its theoretical stability guarantees. Our extensive numerical results on biological, chemical, and social interaction graphs show that the new Topological Graph Contrastive Learning (TopoGCL) model delivers significant performance gains in unsupervised graph classification for 8 out of 12 considered datasets and also exhibits robustness under noisy scenarios.

Introduction

In the last couple of years self-supervised contrastive learning (CL) has emerged as a new promising trend in graph learning which has brought the power of Graph Neural Networks (GNNs) to a broad range of applications without annotated supervisory data, from prediction of molecular properties in biochemistry to discovery of new crystalline materials (Koker et al. 2022; Xu et al. 2021; Fang et al. 2022; Stärk et al. 2022). Indeed, until very recently GNNs have been limited in their representation learning capabilities due to the over-reliance on the existence of task-dependent labels. However, such supervisory information is often handcrafted and may be both scarce and notoriously hard to obtain in many real-life applications of graph learning. For instance, labeling e-crime activity on blockchain transaction graphs typically involves a highly resource-intensive process of manual annotation by the law enforcement agencies, while in bioinformatics and material research graph labeling requires costly and time-consuming wet-lab experiments.

The emerging GCL paradigm rests on the two major components: 1) *augmentation* which constructs multiple views of the graph by exploiting invariance under various transformations such as subgraph sampling, perturbations of nodes and edges, and attribute masking (You et al. 2020; Zeng and Xie 2021); and 2) *contrastive learning (CL)* itself which maximizes mutual information (MI) among the views generated from the resulting graph augmentations, such that positive pairs are contrasted with their negative counterparts (Sun et al. 2019; Velickovic et al. 2019; Hassani and Khasahmadi 2020). The CL step is performed by contrasting node-level representations and graph-level representations, and the three traditional contrasting modes are local-local CL, global-local CL, and global-global CL. However, since the agreement analysis among representations is typically assessed using cosine similarity of the related embeddings, these contrasting approaches cannot systematically account for similarity of higher-order graph properties, for instance, simultaneous matching among subgraphs of varying sizes and orders. In turn, such polyadic node interactions, including various network motifs and other multi-node graph substructures, often play the key role in graph learning tasks, especially, in conjunction with prediction of protein functions in protein-protein interactions and fraud detection in financial networks (Benson, Gleich, and Leskovec 2016; Chen, Gel, and Poor 2022). Interestingly, as shown by (You et al. 2020), subgraphs also tend to play the uniformly consistent role in the data augmentation step of GCL across all types of the considered graphs, from bioinformatics to social networks.

Motivated by the recent success of the computational algebraic topology in graph representation learning, we introduce the concepts of topological representation invariance and extended persistent homology (PH) to GCL. In particular, PH is a tool in computational topology which retrieves evolution of the shape patterns in the observed data along various user-defined geometric dimensions (Hofer, Kwitt, and Niethammer 2019; Edelsbrunner, Letscher, and Zomorodian 2000; Zomorodian and Carlsson 2005). By "shape" here, we broadly refer to the data properties which are invariant under continuous transformations such as bending, stretching, and twisting. PH tools, often in a form of a fully trainable topological layer (Carrière et al. 2020; Chen, Coskunuzer, and Gel 2021; Yan et al. 2021; Horn et al.

2022), have proliferated into a wide range of graph learning tasks, from node and graph classification to anomaly detection to link prediction, often resulting not only in bolstering GNN performance but also in enhancing robustness against random perturbations and adversarial attacks. Nevertheless, despite the well-documented PH utility in semi-supervised graph learning and the intuitive premise of topological invariance to better assess similarity among graph augmentations (see a visual experiment with proteins in Figure 4 and Appendix D.2), PH concepts have never been explored in conjunction with GCL. Here we bridge this gap and introduce the new contrastive mode which targets topological representations of the two augmented views from the same graph. We specifically focus on the notion of extended persistence (EP) which, despite being under-explored in ML applications, has been shown to retrieve a richer topological structure from the observed data and, hence, be particularly suitable for shape matching within GCL. Furthermore, we introduce a new EP summary, extended persistence landscapes (EPL) and prove its stability guarantees. We also contrast theoretical and numerical properties of EPL with respect to another EP summary, extended persistence images (EPI). While EPI has been used in applications before (Yan et al. 2021), it has neither been formally defined for the EP case nor its properties have been evaluated. Armed with EPL, EPI and the associated extended topological layer, we develop a new Topological Graph Contrastive Learning (TopoGCL) model, equipped with contrastive mode on extended topological representations (topological-topological CL (*aka* topo-topo CL)) that allows us to capture not only the critical topological and geometric graph information but also to enhance its latent representation learning.

Significance of our contributions can be summarized as follows:

- TopoGCL is the first approach introducing the concepts of persistent homology to graph contrastive learning.

- We propose a new summary of extended persistence, namely, extended persistence landscapes, and prove its theoretical stability guarantees.

- We validate the utility of TopoGCL in conjunction with unsupervised graph classifications on 12 benchmark datasets from biology, chemistry, and social sciences. Our findings indicate that in addition to outperforming state-of-the-art baselines on 11 out of 12 benchmarks and delivering (statistically) significant gains on 8 datasets, TopoGCL also yields highly promising results in terms of robustness to noise.

Related Work

Graph Representation Learning and Graph Contrastive Learning. Recently, inspired by the success of convolutional neural networks (CNN) on image-based tasks, graph neural networks (GNNs) have emerged as a powerful tool for graph representation learning. Based on the spectral graph theory, (Bruna et al. 2014) introduces a graph-based convolution in Fourier domain. However, the complexity of this model is very high since all Laplacian eigenvectors are needed. To tackle this problem, ChebNet (Defferrard, Bresson, and Vandergheynst 2016) integrates spectral graph convolution with Chebyshev polynomials. Then, Graph Convolutional Networks (GCNs) of (Kipf and Welling 2017) simplify the graph convolution with a localized first-order approximation. More recently, there have been proposed various approaches based on accumulation of the graph information from a wider neighborhood, using diffusion aggregation and random walks. Such higher-order methods include approximate personalized propagation of neural predictions (APPNP) (Klicpera, Bojchevski, and Günnemann 2019), and higher-order graph convolutional architectures (MixHop) (Abu-El-Haija et al. 2019). Moreover, other recent approaches include GNNs with the attention mechanism (GAT, SPAGAN) (Veličković et al. 2018; Yang et al. 2019), and GNNs based on graph diffusion convolution (Gasteiger, Weißenberger, and Günnemann 2019; Zhao et al. 2021). Furthermore, there has appeared a number of approaches introducing a pooling mechanism into GNNs to capture graph (sub)structural information (Cangea et al. 2018; Gao and Ji 2019; Lee, Lee, and Kang 2019; Du et al. 2021). However, such tools mainly focus on supervised and semi-supervised settings and differ from our unsupervised representation learning scheme. Graph contrastive learning is a self-supervised learning approach to learn an encoder (e.g., GNNs without the final classifier) for extracting embeddings from the unlabeled input data. Existing graph contrastive learning approaches mainly focus on three modes, i.e., local-local CL (Zhu et al. 2021; Thakoor et al. 2021), global-local CL (Velickovic et al. 2019; Sun et al. 2019), and global-global CL (You et al. 2020; Li et al. 2022). For instance, GCC (Qiu et al. 2020) proposes a pretraining framework based on local-local CL which constructs multiple graph views by sampling subgraphs based on random walks. For global-local CL, the works (Velickovic et al. 2019; Hassani and Khasahmadi 2020; Asano, Rupprecht, and Vedaldi 2020; Hassani and Khasahmadi 2020) follow the InfoMax principle (Linsker 1988) to maximize the Mutual Information (MI) between the representation of local features and global features. Moreover, another graph contrastive learning mode, i.e., global-global CL (You et al. 2020; Fang et al. 2022) studies the relationships between the global context representations of different samples, which performs better on graph-level tasks. Different from these methods, we propose a novel model Topological Graph Contrastive Learning with the topo-topo CL contrasting mode that not only captures crucial topological and geometrical information but enhances the latent graph representation learning.

Extended Persistence for Machine Learning. Extended persistence (EP) has been introduced by (Cohen-Steiner, Edelsbrunner, and Harer 2009) who show that, by assessing the evolution of shape properties in both upward and downward filtration direction, EP allows us to capture some important topological properties of the underlying object that ordinary persistence cannot. This makes EP particularly attractive for shape matching and CL. However, EP remains substantially less explored in the ML literature, comparing with the ordinary persistence (Carlsson and Vejdemo-Johansson 2021; Adams and Moy 2021; Pun, Lee, and Xia

2022). Some prominent applications of EP in graph learning include link prediction (Yan et al. 2021), node classification (Zhao et al. 2020; Yan et al. 2022), and graph classification (Carrière et al. 2020). To our best knowledge neither EP, nor ordinary persistence or any other tools of computational topology have been ever applied in conjunction with contrastive learning. TopoGCL is the first approach to bridge this knowledge gap.

Notations and Preliminaries

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

Preliminaries on Extended Persistent Homology. PH is a subfield in computational topology whose main goal is to detect, track and encode the evolution of shape patterns in the observed object along various user-selected geometric dimensions (Edelsbrunner, Letscher, and Zomorodian 2000; Zomorodian and Carlsson 2005; Carlsson and Vejdemo-Johansson 2021). These shape patterns represent topological properties such as connected components, loops, and, in general, n-dimensional "holes", that is, the characteristics of the graph $\mathcal{G}$ that remain preserved at different resolutions under continuous transformations. By employing such a multi-resolution approach, PH addresses the intrinsic limitations of classical homology and allows for retrieving the latent shape properties of $\mathcal{G}$ which may play the essential role in a given learning task. The key approach here is to select some suitable scale parameters ν and then to study changes in shape of $\mathcal{G}$ that occur as $\mathcal{G}$ evolves with respect to ν. That is, we no longer study $\mathcal{G}$ as a single object but as a *filtration* $\mathcal{G}_{\nu_1} \subseteq \ldots \subseteq \mathcal{G}_{\nu_n} = \mathcal{G}$, induced by monotonic changes of ν. To ensure that the process of pattern selection and counting is objective and efficient, we build an abstract simplicial complex $\mathscr{K}(\mathcal{G}_{\nu_j})$ on each $\mathcal{G}_{\nu_j}$, which results in a filtration of complexes $\mathscr{K}(\mathcal{G}_{\nu_1}) \subseteq \ldots \subseteq \mathscr{K}(\mathcal{G}_{\nu_n})$. For example, for an edge-weighted graph $(\mathcal{V}, \mathcal{E}, w)$, with the edge-weight function $w : \mathcal{E} \to \mathbb{R}$, we can set $\mathcal{G}_{\leq \nu_j} = (\mathcal{V}, \mathcal{E}, w^{-1}(-\infty, \nu_j])$ for each ν_j, $j = 1, \ldots, n$, yielding the induced sublevel edge-weighted filtration. Similarly, we can consider a function on a node set $\mathcal{V}$, for example, node degree, which results in a sequence of induced subgraphs of $\mathcal{G}$ with a maximal degree of ν_j for each $j = 1, \ldots, n$ and the associated degree sublevel set filtration. We can then record scales b_i (birth) and d_i (death) at which each topological feature first and last appear in the sublevel filtration $\mathcal{G}_{\nu_1} \subseteq \mathcal{G}_{\nu_2} \subseteq \mathcal{G}_{\nu_3} \ldots$. However, in such sublevel filtration, some topological features may never disappear (i.e., persist forever), resulting in a loss of the important information on the underlying latent topological properties of $\mathcal{G}$ and, hence, making it more difficult to use the extracted topological information for shape matching among objects. An alternative approach is to complement the sublevel filtration by its superlevel counterpart, that is, to consider also the sequence of superlevel subgraphs $\mathcal{G}^{\leq \nu_j} = (\mathcal{V}, \mathcal{E}, w^{-1}[\nu_j, \infty))$ for each ν_j, $j = 1, \ldots, n$. That is, we know simultaneously assess evolution of topological features of $\mathcal{G}$ observed over filtrations in both upward and downward directions. This mechanism results in the *extended* persistence, which encompassed information obtained both from sublevel and superlevel filtrations, and the death times can be now also defined as indices at which the topological feature reappears in the superlevel sequence of graphs $\mathcal{G}^{\leq \nu_j}$. The extracted extended topological information can be then summarized as a multiset in $\mathbb{R}$ called *extended persistence diagram (EPD)* $\text{EDg} = \{(b_\rho, d_\rho) \in \mathbb{R}^2\}$. (For further details, see Appendix C.) Note that the extended persistence ensures that no topological feature persist forever and is hence particularly suitable for latent shape matching, opening new perspectives for topological contrastive learning.

What New Does Topological Invariance Bring to GCL? Figure 4 in Appendix D.2 shows 4 networks PROTEINS dataset, along with their corresponding EPDs. These protein networks are hard to discern visually and their traditional network summaries are also virtually indistinguishable. Furthermore, the state-of-the-art CL models also do not correctly classify these 4 proteins. However, we find that the Wasserstein distances between two EPDs of protein networks are always very high if the two protein networks do not belong to the same class and low, otherwise. This phenomenon underlines that persistence and topological invariance play important roles in CL. (For more details see Appendix D.2.)

Topological Graph Contrastive Learning

We now introduce our topological graph contrastive learning (TopoGCL) model which incorporates both graph and topological representations learning into the CL module. In this section, we first briefly recap graph contrastive learning (GCL). Then we discuss the details of the proposed topological contrastive learning (TopoCL). The overall architecture is demonstrated in Figure 1. To facilitate the reading, the frequently used mathematical notations are summarized in Table 1 in Appendix A.1.

Graph Contrastive Learning

Consider a set of graphs $\boldsymbol{\mathcal{G}} = \{\mathcal{G}_1, \ldots, \mathcal{G}_\Upsilon\}$. Following the InfoMax principle (Linsker 1988), GCL aims to perform pre-training through maximizing the mutual information between two augmented views of the same graph via a contrastive loss in the learned latent space (Hassani and Khasahmadi 2020; You et al. 2020). For a better illustration, let us start, as an example, with a case of one graph $\mathcal{G}_i$, where $i \in \{1, \ldots, \Upsilon\}$. That is, given $\mathcal{G}_i = \{\mathcal{V}_i, \mathcal{E}_i, \boldsymbol{X}_i\}$, we first corrupt the original $\mathcal{G}_i$ by an explicit corruption pipeline $\mathcal{T}(\cdot)$ (e.g., node perturbation, edge perturbation, or node feature shuffling) to convert the graph into the two perturbed versions $\tilde{\mathcal{G}}_i = \mathcal{T}_i(\mathcal{G}_i) = \{\tilde{\mathcal{V}}_i, \tilde{\mathcal{E}}_i, \tilde{\boldsymbol{X}}_i\}$ and $\tilde{\mathcal{G}}'_i = \mathcal{T}'_i(\mathcal{G}_i) = \{\tilde{\mathcal{V}}'_i, \tilde{\mathcal{E}}'_i, \tilde{\boldsymbol{X}}'_i\}$. Then, both $\tilde{\mathcal{G}}_i$ and $\tilde{\mathcal{G}}'_i$ are fed into a shared $f_{\text{ENCODER}}(\cdot)$ for graph representation learning (see the blue box in Figure 1). Here $\tilde{\boldsymbol{H}}_i = f_{\text{ENCODER}}(\tilde{\mathcal{G}}_i)$ and $\tilde{\boldsymbol{H}}'_i = f_{\text{ENCODER}}(\tilde{\mathcal{G}}'_i)$ are the learned representations

of the two augmented views of the original graph $\mathcal{G}_i$. The contrastive loss function for the positive pair of samples $\ell_G(\tilde{\mathcal{G}}_i, \tilde{\mathcal{G}}'_i)$ is formulated as:

$$\ell_{i,G}(\tilde{\mathcal{G}}_i, \tilde{\mathcal{G}}'_i) = -\log \frac{\exp\left(\mathrm{sim}(\tilde{\boldsymbol{H}}_i, \tilde{\boldsymbol{H}}'_i)/\zeta\right)}{\sum_{\gamma, \gamma \neq i}^{2\Upsilon} \exp\left(\mathrm{sim}(\tilde{\boldsymbol{H}}_i, \tilde{\boldsymbol{H}}_\gamma)/\zeta\right)}, \quad (1)$$

where $\mathrm{sim}(\tilde{\boldsymbol{H}}_i, \tilde{\boldsymbol{H}}'_i) = \tilde{\boldsymbol{H}}_i^\top \tilde{\boldsymbol{H}}'_i / \|\tilde{\boldsymbol{H}}_i\|\|\tilde{\boldsymbol{H}}'_i\|$, $\tilde{\boldsymbol{H}}_\gamma = f_{\mathrm{ENCODER}}(\tilde{\mathcal{G}}_\gamma)$ denotes the graph representation of $\tilde{\mathcal{G}}_\gamma$, and ζ is the temperature hyperparameter.

Topological Contrastive Learning

The ultimate goal of TopoCL is to contrast latent shape properties of the two augmented views from the same graph, assessed at different resolution scales, by contrasting their respective extended topological representations. Below we introduce the two key components of our method, i.e., extraction of the extended topological features and extended topological representation learning. We focus our discussion on a perturbed graph $\tilde{\mathcal{G}}$ for the sake of simplicity (omitting the subscript i). Since an extended persistence diagram EDg is a multiset in $\mathbb{R}^2$, it cannot be directly used as an input into ML models. To facilitate effective and flexible downstream applications, we utilize the representations of EPD in a functional Hilbert space, i.e., the extended persistence landscape (EPL) and the extended persistence image (EPI). We also design a novel extended topological layer based on the given EDg. As such, broadly speaking, TopoCL consists of three steps: (i) extracting the latent shape properties of the graph using extended persistence in a form of extended persistence diagram EDg and then converting EDg into either EPL or EPI, (ii) constructing the extended topological layer, and (iii) then contrasting the derived topological representations.

Extended Persistence Vectorization. Here we focus on two summaries for EP, Extended Persistence Landscapes and Extended Persistence Images. Both of them are motivated by their respective counterparts as summaries of ordinary persistence. However, while EPI has appeared in applications before (Yan et al. 2021), it has not been formally defined. EPL is a new summary, and we derive its theoretical stability guarantees and discuss its properties in comparison with EPI.

Extended Persistence Landscape (EPL). Inspired by persistence landscapes for ordinary persistence of (Bubenik 2015), we propose a new computationally efficient EP summary called Extended Persistence Landscape. Consider the generating functions Λ_i for each $(b_i, d_i) \in$ EDg, i.e., $\Lambda_i : \mathbb{R} \to \mathbb{R}$ is the piecewise linear function obtained by two line segments starting from $(b_i, 0)$ and $(d_i, 0)$ connecting to the same point $(\frac{b_i + d_i}{2}, \frac{d_i - b_i}{2})$ and 0 in $\mathbb{R} \setminus [b_i, d_i]$. Please see Appendix C for an extended exposition on Extended Homology.

Definition 0.1 (Extended Persistence Landscape). Decompose extended persistence diagram EDg as

$$\mathrm{EDg} = \mathrm{EDg}^+ \cup \mathrm{EDg}^-,$$

where the $+$ and $-$ sets contain the points (b_i, d_i) with $b_i < d_i$ and $d_i < b_i$, respectively. Define the kth landscape function $\lambda_k(\mathcal{G})(t)$ as the kth largest value of $\{\Lambda_i(t) : (b_i, d_i) \in \mathrm{EDg}^+\}$. Similarly, define the $(-j)^{th}$ landscape function $\lambda_{-j}(\mathcal{G})(t)$ as the jth smallest value of $\{\Lambda_i(t) : (b_i, d_i) \in \mathrm{EDg}^-\}$. Then the *Extended Persistence Landscape* is the set of landscape functions

$$\lambda(\mathcal{G}) = \{\lambda_k(\mathcal{G})\} \cup \{\lambda_{-j}(\mathcal{G})\}.$$

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

Definition 0.2 (Distances between EPLs). Let EDg_1 and EDg_2 be two EPDs with corresponding extended persistence landscapes $\lambda(\mathcal{G}_1)$ and $\lambda(\mathcal{G}_2)$. For $1 \leq p \leq \infty$, the *p-landscape distance* between EDg_1 and EDg_2 is defined as

$$\Lambda_p(\mathrm{EDg}_1, \mathrm{EDg}_2) = \|\lambda(\mathcal{G}_1) - \lambda(\mathcal{G}_2)\|_p,$$

where $\|\cdot\|_p$ is a ℓ_p-norm. Analogously, if EM_1 and EM_2 are the persistence modules corresponding to EDg_1 and EDg_2, the *p-landscape distance* $(1 \leq p \leq \infty)$ between EM_1 and EM_2 is defined as

$$\Lambda_p(\mathrm{EM}_1, \mathrm{EM}_2) = \|\lambda(\mathcal{G}_1) - \lambda(\mathcal{G}_2)\|_p.$$

Now consider the case that we have piecewise linear functions f and g inducing filtrations on the simplicial complex $\mathscr{K}$ and taking values in $\mathbb{R}$. Functions f and g define extended persistence modules EM_1 and EM_2, extended persistence diagrams EDg_1 and EDg_2, as well as extended landscapes $\lambda(\mathcal{G}_1)$ and $\lambda(\mathcal{G}_2)$, respectively. Stability of EPD (Cohen-Steiner, Edelsbrunner, and Harer 2009) holds in the following sense:

$$d_B(\mathrm{EDg}_1, \mathrm{EDg}_2) \leq \|f - g\|_\infty, \quad (2)$$

where d_B is the bottleneck distance. In particular, after extending the notion of ϵ-interleaving to the case of extended persistence, it is also true that

$$\Lambda_\infty(\mathrm{EM}_1, \mathrm{EM}_2) \leq d_I(\mathrm{EM}_1, \mathrm{EM}_2), \quad (3)$$

where d_I is the interleaving distance (Bubenik 2015). Armed with these results, we state the stability guarantees for EPL.

Proposition 0.3 (Stability of EPL)**.** *Let EDg_1 and EDg_2 be EPDs for the piecewise linear functions $f, g : \mathscr{K} \to \mathbb{R}$ respectively, then their corresponding ∞-landscape distance satisfies*

$$\Lambda_\infty(EDg_1, EDg_2) \leq d_B(EDg_1, EDg_2) \leq \|f - g\|_\infty.$$

Proof of Proposition 4.3 is in Appendix B.

Extended Persistent Image (EPI). Similarly, the extracted EP information can be encoded in a form of the *Extended Persistence Image* (EPI), which is an analogue of the ordinary persistence image proposed by (Adams et al.

2017). EPI has been used in graph learning before, for example, in conjunction with link prediction (Yan et al. 2021), but has not been formally defined.

EPL is as a finite-dimensional vector representation derived by the weighted kernel density function and can be formulated via the following two steps: *Step 1:* mapping the EDg to an integrable function $\rho_{\text{EDg}} : \mathbb{R} \to \mathbb{R}^2$, which is called a persistence surface. The persistence surface ρ_{EDg} is given by sums of weighted Gaussian functions that are centered at each point in EDg and *Step 2:* integrating the persistence surface ρ_{EDg} over each grid box to obtain EPI. Specifically, the value of each pixel z within the EPI is formed as:

$$\text{EPI}(z) = \iint_z \sum_{\mu \in I} \frac{f(\mu)}{2\pi\sigma_x\sigma_y} e^{-\left(\frac{(x-\mu_x)^2}{2\sigma_x^2} + \frac{(y-\mu_y)^2}{2\sigma_y^2}\right)} dy dx,$$

where $f(\mu)$ is a weighting function (where mean $\mu = (\mu_x, \mu_y) \in \mathbb{R}^2$), and σ_x and σ_y are the standard deviations in x and y directions.

Remark 0.4 (On theoretical properties of EPI and relationships to EPL). There are differences in theoretical properties between the two extended persistence summaries presented in the present work, namely, EPL and EPI. As shown by Proposition 4.3, EPLs are stable under the ∞-landscape distance and, furthermore, the distance between two EPIs is bounded by the bottleneck distance among the respective EPDs. On the contrary, similar stability result does not hold for EPIs. Indeed, as shown by (Adams et al. 2017), PIs for ordinary persistence are stable only with respect to the 1-Wasserstein distance. However, at this point there exists a result on stability of EPDs with respect to the bottleneck distance only (Cohen-Steiner, Edelsbrunner, and Harer 2009) and, hence, nothing can be said on stability of EPIs. Nevertheless, stability and the associated universal distances for extended persistence is an active research area in algebraic topology (Bauer, Botnan, and Fluhr 2022). We then leave this fundamental result as a future fundamental research direction.

Multiple Filtrations. To gain a better understanding of the complex representations of graph data, instead of a single filtration, we hypothesize that learning topological representations via multiple filtrations can benefit neural network framework in gaining both generalization and robustness. Hence, here we consider $\mathcal{Q}$ sublevel filtration functions $\mathfrak{F} = \{\mathfrak{F}_1, \ldots, \mathfrak{F}_q, \ldots, \mathfrak{F}_{\mathcal{Q}}\}$ defined on nodes of $\mathcal{G}$ (where $q \in [1, \mathcal{Q}]$ and $\mathcal{Q} \geq 1$). For each filtration $\mathfrak{F}_q$, we obtain an extended persistence diagram denoted by EDg_q. Then, by using $\mathcal{Q}$ different filtration functions, we can generate a set of persistence diagrams, i.e., $\textbf{EDg}_{\mathcal{Q}} = \{\text{EDg}_1, \ldots, \text{EDg}_{\mathcal{Q}}\}$. Through the extended persistence vectorization methods above, we can have $\textbf{EPI}_{\mathcal{Q}} = \{\text{EPI}_1, \ldots, \text{EPI}_{\mathcal{Q}}\}$ and $\textbf{EPL}_{\mathcal{Q}} = \{\text{EPL}_1, \ldots, \text{EPL}_{\mathcal{Q}}\}$. For the sake of simplicity, we denote $\textbf{EPI}_{\mathcal{Q}}/\textbf{EPL}_{\mathcal{Q}}$ as Ξ in the following discussion.

Extended Topological Layer. To learn critical information on the extended topological features, we propose the extended topological layer (ETL) denoted by Ψ. ETL is a extended topological representation learning layer, illustrated in Figure 1 (see the orange box). Given extended topological features $\tilde{\Xi}$, the ETL operator will output the latent extended topological representation $\tilde{Z}$ with shape d_c as follows (d_c is the number of channels in output):

$$\tilde{Z} = \Psi(\tilde{\Xi}) = \begin{cases} \phi_{\text{MAX}}\left(f_{\text{CNN}}^{(\ell)}(\tilde{\Xi})\right), & \tilde{\Xi} = \tilde{\textbf{EPI}} \\ \text{MLPs}(\tilde{\Xi}), & \tilde{\Xi} = \tilde{\textbf{EPL}} \end{cases}, \quad (4)$$

where $f_{\text{CNN}}^{(\ell)}$ is the convolutional neural network (CNN) in the ℓ-th layer, ϕ_{MAX} denotes global max-pooling layer, and MLPs denotes multi-layer perceptrons. Specifically, the ETL provides two types of topological signature embedding functions, i.e., (i) if the input is in the form of an extended persistence image, we use a CNN-based model (e.g., residual networks) to learn the corresponding topological features and employ global max-pooling layer to obtain the image-level feature, and (ii) if the input is the form of extended persistence landscape, we can use MLPs to generate latent topological representation.

Contrastive Learning on Topological Representations. Using the above Equation 4, we generate latent extended topological representations $\tilde{Z}_i = \Psi(\tilde{\Xi}_i)$ and $\tilde{Z}_i' = \Psi(\tilde{\Xi}_i')$ from two perturbed graphs $\tilde{\mathcal{G}}_i$ and $\tilde{\mathcal{G}}_i'$ (i.e., two augmented views from the same graph $\mathcal{G}$), respectively. Following the previous results of (You et al. 2020), for every latent extended topological representation $\tilde{Z}_i$ being the anchor instance, its positive sample is the latent extended topological representation $\tilde{Z}_i'$ of the another augmented view. Naturally, negative pairs are latent extended topological representations (e.g., $\tilde{Z}_\gamma$) generated from other augmented graphs (e.g., $\tilde{\mathcal{G}}_\gamma \in \tilde{\mathcal{G}} \setminus \{\tilde{\mathcal{G}}_i, \tilde{\mathcal{G}}_i'\}$, where $\tilde{\mathcal{G}}_\gamma$ is an augmented graph of $\mathcal{G}_\gamma$ and $\tilde{\mathcal{G}}$ is a set of 2Υ augmented graphs). Then the loss of function is defined to enforce maximizing the consistency between positive pairs $(\tilde{Z}_i, \tilde{Z}_i')$ compared with negative pairs, which is formulated as:

$$\ell_{i,\text{T}}(\tilde{\mathcal{G}}_i, \tilde{\mathcal{G}}_i') = -\log \frac{\exp\left(\text{sim}(\tilde{Z}_i, \tilde{Z}_i')/\zeta\right)}{\sum_{\gamma, \gamma \neq i}^{2\Upsilon} \exp\left(\text{sim}(\tilde{Z}_i, \tilde{Z}_\gamma)/\zeta\right)}, \quad (5)$$

where $\text{sim}(\tilde{Z}_i, \tilde{Z}_i') = \tilde{Z}_i^\top \tilde{Z}_i'/||\tilde{Z}_i||||\tilde{Z}_i'||$ and $\tilde{Z}_\gamma$ is the latent extended topological representation of $\tilde{\mathcal{G}}_\gamma$ learnt by our proposed ETL. Intuitively, the final training objective function ℓ combines Equations 1 and 5, i.e., $\ell = \alpha \times \sum_{i=1}^{\Upsilon} \ell_{i,\text{G}} + \beta \times \sum_{i=1}^{\Upsilon} \ell_{i,\text{T}}$, where α and β are hyperparameters which balance the contribution of two contrastive losses.

Experiments

Datasets and Baselines. We validate TopoGCL on unsupervised representation learning tasks using the following 12 real-world graph datasets: (i) 5 chemical compound datasets: NCI1, MUTAG, DHFR, BZR, and COX2, (ii) 4 molecular compound datasets: DD, PROTEINS, PTC_MR, and PTC_FM, (iii) 2 internet movie databases: IMDB-BINARY (IMDB-B) and IMDB-MULTI (IMDB-M), and (iv) 1 Reddit discussion threads dataset: REDDIT-BINARY (REDDIT-B). Each dataset includes multiple graphs of each class, and we aim to classify graph classes. The statistics
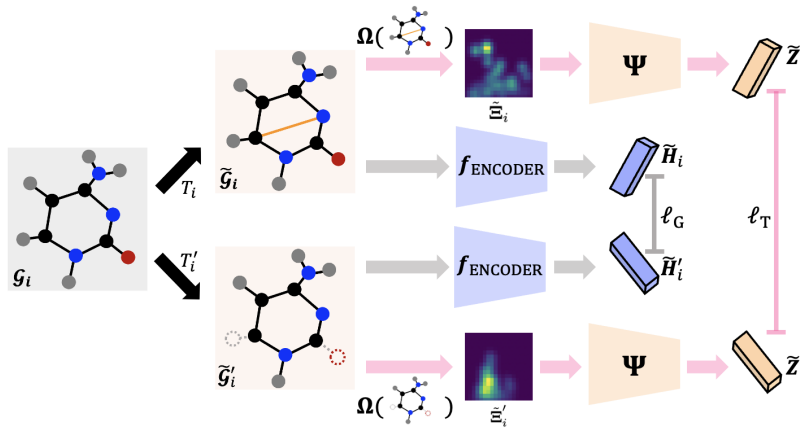
Figure 1: The overall architecture of TopoGCL (see more details in Appendix A.2).

Model	NCI1	PROTEINS	DD	MUTAG	DHFR	BZR	COX2	PTC_MR	PTC_FM
GL	N/A	N/A	N/A	81.66±2.11	N/A	N/A	N/A	57.30±1.40	N/A
WL	80.01±0.50	72.92±0.56	74.00±2.20	80.72±3.00	N/A	N/A	N/A	58.00±0.50	N/A
DGK	80.31±0.46	73.30±0.82	N/A	87.44±2.72	N/A	N/A	N/A	60.10±2.60	N/A
node2vec	54.89±1.61	57.49±3.57	N/A	72.63±10.20	N/A	N/A	N/A	N/A	N/A
sub2vec	52.84±1.47	53.03±5.55	N/A	61.05±15.80	N/A	N/A	N/A	N/A	N/A
graph2vec	73.22±1.81	73.30±2.05	N/A	83.15±9.25	N/A	N/A	N/A	N/A	N/A
InfoGraph	76.20±1.06	74.44±0.31	72.85±1.78	89.01±1.13	80.48±1.34	84.84±0.86	80.55±0.51	61.70±1.40	61.55±0.92
GraphCL	77.87±0.41	74.39±0.45	78.62±0.40	86.80±1.34	68.81±4.15	84.20±0.86	81.10±0.82	61.30±2.10	65.26±0.59
AD-GCL	73.91±0.77	73.28±0.46	75.79±0.87	88.74±1.85	75.66±0.62	85.97±0.63	78.68±0.56	63.20±2.40	64.99±0.77
AutoGCL	**82.00±0.29**	75.80±0.36	77.57±0.60	88.64±1.08	77.33±0.76	86.27±0.71	79.31±0.70	63.10±2.30	63.62±0.55
RGCL	78.14±1.08	75.03±0.43	78.86±0.48	87.66±1.01	76.37±1.35	84.54±1.67	79.31±0.68	61.43±2.50	64.29±0.32
GCL-TAGS	71.43±0.49	75.78±0.41	75.78±0.52	89.12±0.76	N/A	N/A	N/A	N/A	N/A
TopoGCL	81.30±0.27***	**77.30±0.89***	**79.15±0.35*****	**90.09±0.93*****	**82.12±0.69*****	**87.17±0.83****	**81.45±0.55**	63.43±1.13***	**67.11±1.08**

Table 1: Performance on molecular and chemical graphs.

Model	IMDB-B	IMDB-M	REDDIT-B
GL	65.87±0.98	46.50±0.30	77.34±0.18
WL	72.30±3.44	47.00±0.50	68.82±0.41
DGK	66.96±0.56	44.60±0.50	78.04±0.39
node2vec	56.40±2.80	36.00±0.70	69.70±4.10
sub2vec	55.26±1.54	36.70±0.80	71.48±0.41
graph2vec	71.10±0.54	50.40±0.90	75.78±1.03
InfoGraph	73.03±0.87	49.70±0.50	82.50±1.42
GraphCL	71.14±0.44	49.20±0.60	89.53±0.84
AD-GCL	70.21±0.68	50.60±0.70	90.07±0.85
AutoGCL	72.32±0.93	50.60±0.80	88.58±1.49
RGCL	71.85±0.84	49.31±0.42	90.34±0.58
GCL-TAGS	73.65±0.69	52.16±0.72	83.62±0.64
TopoGCL	*****74.67±0.32**	**52.81±0.31**	**90.40±0.53**

Table 2: Performance on social graphs.

of the 12 graph datasets are shown in Appendix D, Table 2. For all graphs, following the experimental settings of GraphCL (You et al. 2020), we use 10-fold cross validation accuracy as the classification performance (based on a non-linear SVM model, i.e., LIB-SVM (Chang and Lin 2011)) and repeat the experiments 5 times to report the mean and standard deviation. The best results are given in **bold** while the best performances achieved by the runner-ups are underlined. We also conduct a one-sided two-sample t-test between the best result and the best performance achieved by the runner-up, where *, **, *** are p-value $< 0.1, 0.05, 0.01$, i.e., significant, statistically significant, highly statistically significant results, respectively. We evaluate the performances of our TopoGCL on 12 graph datasets versus 12 state-of-the-art baselines including: (i) Graphlet Kernel (GL) (Shervashidze et al. 2009), (ii) Weisfeiler-Lehman Sub-tree Kernel (WL) (Shervashidze et al. 2011), (iii) Deep Graph Kernels (DGK) (Yanardag and Vishwanathan 2015), (iv) node2vec (Grover and Leskovec 2016), (v) sub2vec (Adhikari et al. 2018), (vi) graph2vec (Narayanan et al. 2017), (vii) InfoGraph (Sun et al. 2019), (viii) GraphCL (You et al. 2020), (ix) AD-GCL (Suresh et al. 2021), (x) AutoGCL (Yin et al. 2022), (xi) RGCL (Li et al. 2022), and (xii) GCL-TAGS (Lin, Chen, and Wang 2022).

Experiment Settings. We conduct our experiments on two NVIDIA RTX A5000 GPU cards with 24GB memory. TopoGCL is trained end-to-end by using Adam op-

timizer. The tuning of TopoGCL on each dataset is done via grid hyperparameter configuration search over a fixed set of choices and the same cross-validation setup is used to tune baselines. Table 3 in Appendix D shows the average running time of extended persistence image (EPI) and extended persistence landscape (EPL) generation (in seconds) on all 12 graph datasets. See Appendix D for more details.The source code of TopoGCL is publicly available at https://github.com/topogclaaai24/TopoGCL.git. See Appendix D for more details.

Experiment Results

The evaluation results on 12 graph datasets are summarized in Tables 1 and 2. We also conduct ablation studies and robustness analysis to assess the contributions of the extended persistence and the robustness of TopoGCL against noisy scenarios.

Molecular, Chemical, and Social Graphs. Table 1 shows the performance comparison among 11 baselines on NCI1, PROTEINS, DD, MUTAG, DHFR, BZR, COX2, and two PTC datasets with different carcinogenicities on rodents (i.e., PTC_MR, and PTC_FM) for graph classification. Our TopoGCL is comparable with the state-of-the-art on the NCI1 and PTC_MM datasets, and consistently outperforms baseline models on other 9 datasets. In particular, the average relative gain of TopoGCL over the runner-ups is 0.78%. The results demonstrate the effectiveness of TopoGCL. In terms of unsupervised graph-level representation learning baselines, e.g., node2vec only focuses on the graph structure learning and generates node embeddings through random walks. In turn, InfoGraph works by learning graph representation via GNNs, and taking graph representation and patch representation as pairs for unsupervised representation learning, hence, resulting in improvement over random walk-based graph embedding methods. Comparing with InfoGraph, graph contrastive learning methods such as GraphCL and RGCL explore the view augmentations approaches and learn representations of augmented graph structures for graph contrastive learning. A common limitation of these approaches is that they do not simultaneously capture both topological properties of the graph and information from the graph structure. Hence, it is not surprising that performance of our proposed TopoGCL which systematically integrates all types of the above information on the observed graphs is substantially higher than that of the benchmark models. Besides, we have conducted a visual experiment and the corresponding validity evaluation of the extracted topological and geometric information and its role in GCL (see Appendix D for a discussion). Table 2 shows the performance comparison on 3 social graph datasets. Similarly, Table 2 indicates that our TopoGCL model is always better than baselines for all social graph datasets. Moreover, we find that TopoGCL also has the smallest standard deviation across 3 social graph datasets, revealing that local topological representation learning module can enhance the model stability.

Ablation Study of the Topological Signatures. We perform ablation studies on MUTAG, PTC_MR, and IMDB-B to justify the following opinions: (i) the benefit of the

	AD-GCN	RGCL	TopoGCL (ours)
MUTAG	88.20±1.24	87.05±1.16	***89.47±1.04**
PTC_MR	60.88±0.92	61.10±1.47	**61.63±1.09**
IMDB-B	70.01±0.86	70.59±0.34	***72.18±0.15**

Table 3: Robustness study against noise.

extended topological signatures in topological representation learning; and (ii) measuring the similarity of positive samples and the diversity between negative samples via both global-global CL (i.e., GCL) and topo-topo CL can achieve better performance than only considering Global-Global CL. For opinion (i), we compare TopoGCL (i.e., TopoGCL + EPI/EPL) with its variant (i.e, TopoGCL + PI/PL). Note that, for PTC_MR, we use EPL/PL due to TopoGCL based on persistence landscapes achieves better performance than persistence images. Results are shown in Table 4, Appendix E. We observe that TopoGCL based on extended topological signatures performs much better than the variant, demonstrating that our extended topological signatures can capture higher-order structural and topological information efficiently. Note that we found TopoGCL can achieve more competitive results on PTC_MR by using EPL instead of EPI; thus we specially compare TopoGCL+EPL with TopoGCL+PL in this ablation study. Besides, we validate TopoGCL by comparing TopoGCL's performance with the performance of TopoGCL W/o Topo and the experimental results in Table 5 in Appendix E show that topo-topo CL can bring performance gain since the enhancement of structural and topological information learning.

Robustness Study. To evaluate the robustness of our proposed TopoGCL under noisy conditions, in this section, we consider adding Gaussian noise into node features of 20% data. Note that the added noise follows i.i.d Gaussian density, i.e., $\mathcal{N}(1, 1)$ (where $\mu = \sigma = 1$). The performances of TopoGCL and other baselines (i.e., AD-GCL and RGCL) on MUTAG, PTC_MR, and IMDB-B are shown in Table 3. We observe that our TopoGCL always outperforms two state-of-the-art baselines on all three datasets. The strong performance verifies the superiority of TopoGCL and demonstrates that TopoGCL can efficiently exploit the underlying topological features of the input graph structure.

Conclusion

We have proposed a novel contrastive learning model, i.e., Topological Graph Contrastive Learning (TopoGCL). TopoGCL adopts a new contrasting mode (topo-topo CL) which can capture both local and global latent topological information. Our extensive experimental results have demonstrated that TopoGCL achieves impressive improvements over the state-of-the-art GCL models in terms of accuracy and enhances the robustness against noisy scenarios. In the future, we will advance TopoGCL and the ideas of extended persistence to self-supervised learning of time-evolving graphs, with a particular focus on streaming scenarios.

Acknowledgements

References

Abu-El-Haija, S.; Perozzi, B.; Kapoor, A.; Alipourfard, N.; Lerman, K.; Harutyunyan, H.; Steeg, G. V.; and Galstyan, A. 2019. Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing. In *ICML*.

Adams, H.; Emerson, T.; Kirby, M.; Neville, R.; Peterson, C.; Shipman, P.; Chepushtanova, S.; Hanson, E.; Motta, F.; and Ziegelmeier, L. 2017. Persistence images: A stable vector representation of persistent homology. *JMLR*, 18.

Adams, H.; and Moy, M. 2021. Topology applied to machine learning: From global to local. *Frontiers in Artificial Intelligence*, 4: 668302.

Adhikari, B.; Zhang, Y.; Ramakrishnan, N.; and Prakash, B. A. 2018. Sub2vec: Feature learning for subgraphs. In *PAKDD*, 170–182. Springer.

Asano, Y.; Rupprecht, C.; and Vedaldi, A. 2020. Self-labelling via simultaneous clustering and representation learning. In *ICLR*.

Bauer, U.; Botnan, M. B.; and Fluhr, B. 2022. Universal Distances for Extended Persistence. *arXiv preprint arXiv:2007.01834*.

Benson, A. R.; Gleich, D. F.; and Leskovec, J. 2016. Higher-order organization of complex networks. *Science*, 353(6295): 163–166.

Bruna, J.; Zaremba, W.; Szlam, A.; and LeCun, Y. 2014. Spectral networks and locally connected networks on graphs. In *ICLR*.

Bubenik, P. 2015. Statistical Topological Data Analysis using Persistence Landscapes. *JMLR*, 16(1): 77–102.

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

Carlsson, G.; and Vejdemo-Johansson, M. 2021. *Topological data analysis with applications*. Cambridge University Press.

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

Chang, C.-C.; and Lin, C.-J. 2011. LIBSVM: a library for support vector machines. *ACM TIST*, 2(3): 1–27.

Chen, Y.; Coskunuzer, B.; and Gel, Y. 2021. Topological relational learning on graphs. *NeurIPS*, 34: 27029–27042.

Chen, Y.; Gel, Y. R.; and Poor, H. V. 2022. BScNets: block simplicial complex neural networks. In *AAAI*, volume 36, 6333–6341.

Cohen-Steiner, D.; Edelsbrunner, H.; and Harer, J. 2009. Extending persistence using Poincaré and Lefschetz duality. *Foundations of Computational Mathematics*, 9(1): 79–103.

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

Du, J.; Wang, S.; Miao, H.; and Zhang, J. 2021. Multi-Channel Pooling Graph Neural Networks. In *IJCAI*, 1442–1448.

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

Fang, Y.; Zhang, Q.; Yang, H.; Zhuang, X.; Deng, S.; Zhang, W.; Qin, M.; Chen, Z.; Fan, X.; and Chen, H. 2022. Molecular contrastive learning with chemical element knowledge graph. In *AAAI*, volume 36, 3968–3976.

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

Gasteiger, J.; Weißenberger, S.; and Günnemann, S. 2019. Diffusion improves graph learning. *NeurIPS*, 32.

Grover, A.; and Leskovec, J. 2016. node2vec: Scalable feature learning for networks. In *SIGKDD*, 855–864.

Hassani, K.; and Khasahmadi, A. H. 2020. Contrastive multi-view representation learning on graphs. In *ICML*, 4116–4126.

Hofer, C. D.; Kwitt, R.; and Niethammer, M. 2019. Learning Representations of Persistence Barcodes. *JMLR*, 20(126): 1–45.

Horn, M.; De Brouwer, E.; Moor, M.; Moreau, Y.; Rieck, B.; and Borgwardt, K. 2022. Topological Graph Neural Networks. In *ICLR*.

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

Klicpera, J.; Bojchevski, A.; and Günnemann, S. 2019. Predict then propagate: Graph neural networks meet personalized pagerank. In *ICLR*.

Koker, T.; Quigley, K.; Spaeth, W.; Frey, N. C.; and Li, L. 2022. Graph Contrastive Learning for Materials.

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

Li, S.; Wang, X.; Zhang, A.; Wu, Y.; He, X.; and Chua, T.-S. 2022. Let invariant rationale discovery inspire graph contrastive learning. In *ICML*, 13052–13065.

Lin, L.; Chen, J.; and Wang, H. 2022. Spectrum Guided Topology Augmentation for Graph Contrastive Learning. In *NeurIPS 2022 Workshop: New Frontiers in Graph Learning*.

Linsker, R. 1988. Self-organization in a perceptual network. *Computer*, 21(3): 105–117.

Narayanan, A.; Chandramohan, M.; Venkatesan, R.; Chen, L.; Liu, Y.; and Jaiswal, S. 2017. graph2vec: Learning distributed representations of graphs. *arXiv preprint arXiv:1707.05005*.

Pun, C. S.; Lee, S. X.; and Xia, K. 2022. Persistent-homology-based machine learning: a survey and a comparative study. *Artificial Intelligence Review*, 55(7): 5169–5213.

Qiu, J.; Chen, Q.; Dong, Y.; Zhang, J.; Yang, H.; Ding, M.; Wang, K.; and Tang, J. 2020. GCC: Graph contrastive coding for graph neural network pre-training. In *SIGKDD*, 1150–1160.

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

Shervashidze, N.; Vishwanathan, S.; Petri, T.; Mehlhorn, K.; and Borgwardt, K. 2009. Efficient graphlet kernels for large graph comparison. In *AISTATS*, 488–495.

Stärk, H.; Beaini, D.; Corso, G.; Tossou, P.; Dallago, C.; Günnemann, S.; and Liò, P. 2022. 3D Infomax improves GNNs for molecular property prediction. In *ICML*, 20479–20502.

Sun, F.-Y.; Hoffman, J.; Verma, V.; and Tang, J. 2019. InfoGraph: Unsupervised and Semi-supervised Graph-Level Representation Learning via Mutual Information Maximization. In *ICLR*.

Suresh, S.; Li, P.; Hao, C.; and Neville, J. 2021. Adversarial graph augmentation to improve graph contrastive learning. *NeurIPS*, 34: 15920–15933.

Thakoor, S.; Tallec, C.; Azar, M. G.; Munos, R.; Veličković, P.; and Valko, M. 2021. Bootstrapped representation learning on graphs. In *ICLR 2021 Workshop on Geometrical and Topological Representation Learning*.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *ICLR*.

Velickovic, P.; Fedus, W.; Hamilton, W. L.; Liò, P.; Bengio, Y.; and Hjelm, R. D. 2019. Deep Graph Infomax. *ICLR*, 2(3): 4.

Xu, M.; Wang, H.; Ni, B.; Guo, H.; and Tang, J. 2021. Self-supervised graph-level representation learning with local and global structure. In *International Conference on Machine Learning*, 11548–11558.

Yan, Z.; Ma, T.; Gao, L.; Tang, Z.; and Chen, C. 2021. Link prediction with persistent homology: An interactive view. In *ICML*, 11659–11669.

Yan, Z.; Ma, T.; Gao, L.; Tang, Z.; Wang, Y.; and Chen, C. 2022. Neural Approximation of Graph Topological Features. In *NeurIPS*.

Yanardag, P.; and Vishwanathan, S. 2015. Deep graph kernels. In *SIGKDD*, 1365–1374.

Yang, Y.; Wang, X.; Song, M.; Yuan, J.; and Tao, D. 2019. SPAGAN: shortest path graph attention network. In *IJCAI*, 4099–4105.

Yin, Y.; Wang, Q.; Huang, S.; Xiong, H.; and Zhang, X. 2022. Autogcl: Automated graph contrastive learning via learnable view generators. In *AAAI*, volume 36, 8892–8900.

You, Y.; Chen, T.; Sui, Y.; Chen, T.; Wang, Z.; and Shen, Y. 2020. Graph contrastive learning with augmentations. *NeurIPS*, 33: 5812–5823.

Zeng, J.; and Xie, P. 2021. Contrastive self-supervised learning for graph classification. In *AAAI*, volume 35, 10824–10832.

Zhao, J.; Dong, Y.; Ding, M.; Kharlamov, E.; and Tang, J. 2021. Adaptive diffusion in graph neural networks. *NeurIPS*, 34: 23321–23333.

Zhao, Q.; Ye, Z.; Chen, C.; and Wang, Y. 2020. Persistence enhanced graph neural network. In *AISTATS*, 2896–2906.

Zhu, Y.; Xu, Y.; Yu, F.; Liu, Q.; Wu, S.; and Wang, L. 2021. Graph contrastive learning with adaptive augmentation. In *WWW*, 2069–2080.

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